Lei Wang
Michael Segal
Jenhui Chen
Tie Qiu (Eds.)

LNCS 13472

Wireless Algorithms, Systems, and Applications

17th International Conference, WASA 2022
Dalian, China, November 24–26, 2022
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Computer Science 13472

More information about this series at https://link.springer.com/bookseries/558

Lei Wang · Michael Segal · Jenhui Chen ·
Tie Qiu (Eds.)

Wireless Algorithms, Systems, and Applications

17th International Conference, WASA 2022
Dalian, China, November 24–26, 2022
Proceedings, Part II

Springer

Editors

Lei Wang
Dalian University of Technology
Dalian, China

Michael Segal
Ben-Gurion University of the Negev
Beer-Sheva, Israel

Jenhui Chen
Chang Gung University
Taiwan, China

Tie Qiu
Tianjin University
Tianjin, China

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-19213-5 ISBN 978-3-031-19214-2 (eBook)
https://doi.org/10.1007/978-3-031-19214-2

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The 17th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2022) was held in Dalian during November 24–26, 2022. The conference focused on new ideas and recent advances in computer systems, wireless networks, distributed applications, and advanced algorithms that are pushing forward the new technologies for better information sharing, computer communication, and universal connected devices in various environments, especially in wireless networks. WASA has become a broad forum for computer theoreticians, system and application developers, and other professionals in networking-related areas to present their ideas, solutions, and knowledge of emerging technologies and challenges in computer systems, wireless networks, and advanced applications.

The technical program of WASA 2022 consisted of 94 regular papers and 68 short papers, selected by the Program Committee from 265 full submissions in response to the call for papers. All submissions were reviewed by at least 115 Program Committee members in a 115 double blind process. The submissions cover numerous cutting edge topics: cognitive radio networks; software-defined radio and reconfigurable radio networks; cyber-physical systems (CPSs) including intelligent transportation systems and smart healthcare systems; theoretical frameworks and analysis of fundamental cross-layer protocol and network design and performance issues; distributed and localized algorithm design and analysis; information and coding theory for wireless networks; localization; mobility models and mobile social networking; mobile cloud; topology control and coverage; security and privacy; underwater and underground networks; vehicular networks; radar and sonar networks; PHY/MAC/routing protocols; information processing and data management; programmable service interfaces; energy-efficient algorithms; systems and protocol design; operating system and middleware support; algorithms, systems, and applications of the Internet of Things (IoT); and algorithms, systems, and applications of edge computing, etc. In the first place, we would like to express our grateful appreciation for all Program Committee members for their hard work in reviewing all submissions. Furthermore, we would like to give our special thanks to the WASA Steering Committee for their consistent leadership and guidance; also, we would like to extend our gratitude to the the local chairs (Jingang Yu, Zumin Wang, and Jie Wang), the publication chairs (Chi Lin, Lei Shu, Guangjie Han, and Pengfei Wang), the publicity chairs (Zichuan Xu, Haipeng Dai, Zhibo Wang, and Chenren Xu), organizing chairs (Dongsheng Zhou and Zhenquan Qin), and the Web chair (Bingxian Lu) for their remarkable contributions to WASA 2022, ensuring that it was a successful conference. In particular, we wish to express our deepest respect and thankfulness to all the authors for submitting and presenting their outstanding ideas and solutions at the conference.

November 2022

Lei Wang
Michael Segal
Jenhui Chen
Tie Qiu

Organization

Steering Committee Members

Xiuzhen Susan Cheng	George Washington University, USA
Zhipeng Cai	Georgia State University, USA
Jiannong Cao	Hong Kong Polytechnic University, Hong Kong, China
Ness Shroff	The Ohio State University, USA
Wei Zhao	University of Macau, Macau, China
Peng-Jun Wan	Illinois Institute of Technology, USA
Ty Znati	University of Pittsburgh, USA
Xinbing Wang	Shanghai Jiao Tong University, China

General Co-chairs

Zhongxuan Luo	Dalian University of Technology, China
Peng-Jun Wan	Illinois Institute of Technology, USA
Xingwei Wang	Northeastern University, China

Program Co-chairs

Lei Wang	Dalian University of Technology, China
Michael Segal	Ben-Gurion University, Israel
Jen-Hui Chen	Chang Gung University, Taiwan, China
Tie Qiu	Tianjin University, China

Publicity Co-chairs

Zichuan Xu	Dalian University of Technology, China
Haipeng Dai	Nanjing University, China
Zhibo Wang	Zhejiang University, China
Chenren Xu	Peking University, China

Publication Co-chairs

Chi Lin	Dalian University of Technology, China
Lei Shu	Nanjing Agricultural University, China
Guangjie Han	Hohai University, China
Pengfei Wang	Dalian University of Technology, China

Local Co-chairs

Jingang Yu	University of Chinese Academy of Sciences, China
Zumin Wang	Dalian University, China
Jie Wang	Dalian Maritime University, China

Web Chair

Bingxian Lu	Dalian University of Technology, China

Organizing Co-chairs

Dongsheng Zhou	Dalian University, China
Zhenquan Qin	Dalian University of Technology, China

Program Committee

Ran Bi	Dalian University of Technology, China
Edoardo Biagioni	University of Hawaii at Manoa, USA
Salim Bitam	University of Biskra, Algeria
Azzedine Boukerche	University of Ottawa, Canada
Zhipeng Cai	Georgia State University, USA
Srinivas Chakravarthi Thandu	Amazon, USA
Sriram Chellappan	University of South Florida, USA
Quan Chen	Guangdong University of Technology, China
Xianfu Chen	VTT Technical Research Centre of Finland, Finland
Xu Chen	Sun Yat-sen University, China
Wei Wang	Sun Yat-sen University, China
Songqing Chen	George Mason University, USA
Soufiene Djahel	Manchester Metropolitan University, UK
Yingfei Dong	University of Hawaii, USA
Zhuojun Duan	James Madison University, USA
Luca Foschini	University of Bologna, Italy
Jing Gao	Dalian University of Technology, China
Xiaofeng Gao	Shanghai Jiao Tong University, China
Jidong Ge	Nanjing University, China
Chunpeng Ge	Nanjing University of Aeronautics and Astronautics, China
Daniel Graham	University of Virginia, USA
Ding Wang	Nankai University, China
Ning Gu	Fudan University, China

Deke Guo	National University of Defense Technology, China
Bin Guo	Northwestern Polytechnical University, China
Meng Han	Kennesaw State University, USA
Suining He	University of Connecticut, USA
Zaobo He	Miami University, USA
Pengfei Hu	Shandong University, China
Peng Sun	The Chinese University of Hong Kong, China
Yan Huang	Kennesaw State University, USA
Yan Huo	Beijing Jiaotong University, China
Holger Karl	University of Paderborn, Germany
Donghyun Kim	Kennesaw State University, USA
Hwangnam Kim	Korea University, South Korea
Bharath Kumar Samanthula	Montclair State University, USA
Abderrahmane Lakas	United Arab Emirates University, UAE
Sanghwan Lee	Kookmin University, South Korea
Feng Li	Shandong University, China
Feng Li	Indiana University-Purdue University Indianapolis, USA
Ruinian Li	Bowling Green State University, USA
Wei Li	Georgia State University, USA
Zhenhua Li	Tsinghua University, China
Zhetao Li	Xiangtan University, China
Peng Li	University of Aizu, Japan
Qi Li	Tsinghua University, China
Yaguang Lin	Shaanxi Normal University, China
Zhen Ling	Southeast University, China
Weimo Liu	George Washington University, USA
Jia Liu	Nanjing University, China
Fangming Liu	Huazhong University of Science and Technology, China
Liang Liu	Beijing University of Posts and Telecommunications, China
Hongbin Luo	Beihang University, China
Jun Luo	Nanyang Technological University, Singapore
Liran Ma	Texas Christian University, USA
Jian Mao	Beihang University, China
Bo Mei	Texas Christian University, USA
Hung Nguyen	Carnegie Mellon University, USA
Pasquale Pace	University of Calabria, Italy
Claudio Palazzi	University of Padua, Italy
Chuan Lin	Northeastern University, China

Junjie Pang	Qingdao University, China
Javier Parra-Arnau	University of Ottawa, Canada
Tie Qiu	Tianjin University, China
Ruben Rios	University of Malaga, Spain
Kazuya Sakai	Tokyo Metropolitan University, Japan
Omar Sami Oubbati	University of Laghouat, Algeria
Kewei Sha	University of Houston - Clear Lake, USA
Hao Sheng	Beihang University, China
Bo Sheng	University of Massachusetts Boston, USA
Tuo Shi	Harbin Institute of Technology, China
Tong Liu	Shanghai University, China
Sukhpal Singh Gill	Queen Mary University of London, UK
Junggab Son	Kennesaw State University, USA
Riccardo Spolaor	Shandong University, China
Chunhua Su	University of Aizu, Japan
Violet Syrotiuk	Arizona State University, USA
Guoming Tang	National University of Defense Technology, China
Bin Tang	Hohai University, China
Xiaohua Tian	Shanghai Jiao Tong University, China
Luis Urquiza	Universitat Politècnica de Catalunya, Spain
Tian Wang	Huaqiao University, China
Yawei Wang	George Washington University, USA
Yingjie Wang	Yantai University, China
Zhibo Wang	Zhejiang University, China
Leye Wang	Peking University, China
Wei Wei	Xi’an University of Technology, China
Alexander Wijesinha	Towson University, USA
Mike Wittie	Montana State University, USA
Kaishun Wu	Shenzhen University, China
Xiaobing Wu	University of Canterbury, New Zealand
Wei Xi	Xi’an Jiaotong University, China
Yang Xiao	University of Alabama, USA
Kun Xie	Hunan University, China
Xuan Liu	Hunan University, China
Kaiqi Xiong	University of South Florida, USA
Kuai Xu	Arizona State University, USA
Wen Xu	Texas Woman’s University, USA
Lei Yang	The Hong Kong Polytechnic University, China
Panlong Yang	University of Science and Technology of China, China

Changyan Yi	Nanjing University of Aeronautics and Astronautics, China
Wei Yu	Towson University, USA
Dongxiao Yu	Shandong University, China
Sherali Zeadally	University of Kentucky, USA
Deze Zeng	China University of Geosciences, China
Bowu Zhang	Marist College, USA
Yong Zhang	Shenzhen Institutes of Advanced Technology, China
Yang Zhang	Wuhan University of Technology, China
Cheng Zhang	George Washington University, USA
Xu Zheng	University of Science and Technology of China, China
Yanwei Zheng	Shandong University, China
Lu Zhou	Nanjing University of Aeronautics and Astronautics, China
Jindan Zhu	Amazon Web Services, USA
Tongxin Zhu	Southeast University, China
Yifei Zou	Shandong University, China

Contents – Part II

Algorithms, Systems, and Applications of Internet of Things

Information Processing and Data Management

Radar and Sonar Networks

Algorithms, Systems, and Applications of Internet of Things

Unsupervised Deep Learning-Based Hybrid Beamforming in Massive MISO Systems

Teng Zhang[1], Anming Dong[1,2](✉), Chuanting Zhang[3], Jiguo Yu[2], Jing Qiu[4], Sufang Li[1], Li Zhang[1,2], and You Zhou[5]

[1] School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China
tengzhang_qlu@163.com, anmingdong@qlu.edu.cn

[2] Big Data Institute and School of Mathematics and Statistics, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China

[3] Department of Electrical and Electronic Engineering, University of Bristol, Bristol BS8 1UB, UK

[4] School of Mathematical Science, Qufu Normal University, Qufu 273100, China

[5] Shandong HiCon New Media Institute Co., Ltd., Jinan, China

Abstract. Hybrid beamforming (HBF) is a promising approach for balancing the hardware cost, training overhead and system performance in massive MIMO systems. Optimizing the HBF through deep learning (DL) has gained considerable attention in recent years due to its potential in dealing with the nonconvex problems. However, existing DL-based HBF methods require wider or deeper neural networks to guarantee training performance, which not only leads to higher complexity in training and deploying, but also increases the risk of over-fitting. In this paper, we propose a low-complexity HBF method based on convolutional neural network (CNN) to solve the spectral efficiency (SE) maximization problem with constant modulus constraint for the analog phase shifters over the transmit power budget in a multiple-input single-output (MISO) system. An unsupervised learning strategy is derived for the constructed CNN to learn to generate feasible beamforming solutions adaptively and thus avoiding any label data when training them. Simulations show its advantages in both SE and complexity over other related algorithms.

This work was supported in part by the National Key R&D Program of China under Grant 2019YFB2102600, the National Natural Science Foundation of China (NSFC) under Grants 61701269, 61832012 and 61771289, and the Shandong Provincial Natural Science Foundation under Grant ZR2021MF026; the Fundamental Research Enhancement Program for Computer Science and Technology, the Talent Cultivation Promotion Program for Computer Science and Technology, and the Piloting Fundamental Research Program for the Integration of Scientific Research, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grants 2021JC02014, 2021PY05001 and 2022XD001, respectively.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 3–15, 2022.
https://doi.org/10.1007/978-3-031-19214-2_1

Keywords: Massive multiple-input multiple-output (MIMO) · Hybrid beamforming · Spectral efficiency · Deep learning · Convolutional neural network

1 Introduction

Massive multiple-input multiple-output (MIMO) has been proposed as a promising solution to meet the requirements of high rate and low latency [1–3], which can compensate for the severe path loss of millimeter wave (mmWave) signals by utilizing a large number of antennas at the transceivers. However, new challenges are posed to massive MIMO since the traditional fully digital beamforming is prohibitively expensive, where dedicated radio frequency (RF) chain is demanded for each antenna [4]. To overcome such issue, hybrid beamforming (HBF) is proposed by combining baseband digital beamformer and analog beamformer in RF domain [5,6]. The HBF architecture significantly reduces the number of RF chains by connecting much fewer RF chains to the antennas via analog phase shifters thereby lowering hardware costs. It gains the benefits of traditional beamforming while providing high beamforming gain. Implementing HBF is non-trivial, since the phase shifters introduce nonconvex constant modulus constraint. Many efforts have been dedicated to address the HBF optimization. Paper [7] proposed an orthogonal matching pursuit-based spatially sparse (SOMP) algorithm that transforms the HBF matrix design into reconstructing the sparse matrix of the signal. An orthogonal codebook vector model is designed in [8] to avoid matrix inverse operations in the optimization process, thus reducing the computational complexity. Paper [9] proposed an manifold optimization-based alternating minimization (MO-AltMin) HBF algorithm. An element-based iterative algorithm was proposed in [10] to further improve the performance. In [11], an exhaustive search method is used for beam selection based on the maximum signal-to-noise ratio (SNR). These works focus mainly on iterative algorithms, which require considerable time for iterative operations and high computational complexity.

Deep learning (DL) is a powerful tool to deal with complex nonconvex optimization problems for its excellent learning and feature extraction capabilities [12–14]. A boom in applying DL to HBF design has emerged in recent years [15–20]. In [15], the authors considered a coordinated beamforming system that uses a DL model to learn how to predict the beamforming vector directly by using the signal received at the distributed base station (BS). [16] used deep neural networks to establish mapping relationships to enhance HBF performance. [17] solved three beamforming optimization problems using DL to design near-optimal beamformer. All these papers use supervised learning to train the network. Supervised learning based on local optimal cannot achieve good performance, since it is hard to obtain global optimal solutions for nonconvex optimization problems. Moreover, the performance of supervised learning relies heavily on a large amount of label data, but the label data is not easily available in wireless communication. Besides, [18,19] using multiple fully connected layers to construct network models may increase the computational complexity.

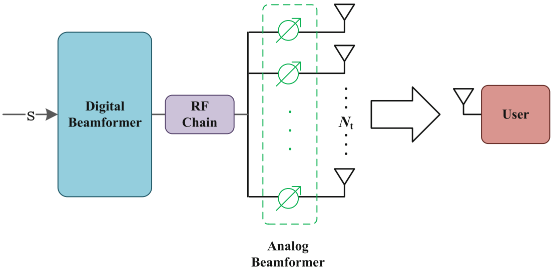


Fig. 1. SU-MISO system architecture with hybrid (analog and baseband) beamforming.

To overcome these challenges, we propose a low-complexity HBF scheme based on convolutional neural network (CNN) trained by an unsupervised learning mechanism. Specifically, we formulate a HBF optimization problem for a multiple-input single-output (MISO) system to maximize the throughput subject to constant modulus constraint of the phase shifters and power constraint at the transmitter. To solve such a nonconvex problem, we construct a novel CNN architecture, which takes the analog beamformer as the optimization target. It employs multiple convolutional blocks to extract more channel features. Besides, a self-defined network layer is designed to make the output satisfy the constant modulus constraint. CNN automatically extracts all important features during the weight updating back-propagation process, which offers a greater advantage over the inefficient manual feature extraction [21]. Compared with fully connected neural network (FCNN)-based algorithm [18], the number of parameters and floating-point operations (FLOPs) of our proposed CNN are reduced significantly due to the feature sharing of convolutional operations, which will result in lower computational complexity. Considering that it is nontrivial to obtain high quality label data, we attempt to train the CNN through an unsupervised mechanism. To this end, we construct a loss function that is the negative of the objective function of the formulated noncovex problem. Given the channel state information (CSI) data, the CNN is then trained by minimizing such a loss function, which equivalently maximizes the achievable rate, without needing any optimal beamformers as label data. Simulations shown that the proposed CNN-based unsupervised learning HBF scheme is capable of optimizing the beamformers effectively and performs superior to the referenced FCNN-based scheme with much lower complexity.

2 System Model and Problem Formulation

2.1 System Model

We consider a downlink MISO communication system shown in Fig. 1, which transmits data to the user by a HBF transmitter. In this scenario, a BS equipped with a single RF chain and N_t antennas transmits a data stream to a single antenna user in an ideal channel environment. We assume that the BS at the transmitter is equipped with the uniform linear array (ULA) consisting of N_t

antenna units. Generally, the antenna spacing r is half of the transmission wavelength λ, i.e., $r = 0.5\lambda$. The input signal s at BS obeys a complex Gaussian distribution with mean 0 and variance 1, i.e., it satisfies $s \sim \mathcal{CN}(0, 1)$.

In the HBF system, the input signal s first passes through the digital beamformer v_{D}, which is actually a scalar since there is only one RF chain at the transmitter side. The signal s is then converted to analog phase shifters through a RF chain, and then the transmit signal $\mathbf{x} = \mathbf{v}_{\mathrm{A}} v_{\mathrm{D}} s \in \mathbb{C}^{N_t \times 1}$ is constructed by passing through the analog beamforming vector $\mathbf{v}_{\mathrm{A}} \in \mathbb{C}^{N_t \times 1}$. The whole downlink HBF vector can be expressed as $\mathbf{v} = \mathbf{v}_{\mathrm{A}} v_{\mathrm{D}}$, where $\mathbf{v}$ is an $N_t \times 1$-dimensional complex vector. After that the transmit signal $\mathbf{x}$ passes through a channel $\mathbf{h}$ to get the received signal y at the receiver side. The received signal at the user is given as

$$y = \mathbf{h}^H \mathbf{v}_{\mathrm{A}} v_{\mathrm{D}} s + n, \tag{1}$$

where $\mathbf{h} \in \mathbb{C}^{N_t \times 1}$ denotes the downlink channel gain complex vector. n stands for the additive Gaussian white noise obeying a complex Gaussian distribution with zero mean and variance σ^2, i.e., n satisfies $n \sim \mathcal{CN}(0, \sigma^2)$. Besides, σ^2 represents the noise power. The achievable spectral efficiency (SE) of the HBF system is then calculated as $R = \log_2\left(1 + \frac{|\mathbf{h}^H \mathbf{v}_{\mathrm{A}} v_{\mathrm{D}}|^2}{\sigma^2}\right)$.

2.2 Problem Formulation

We assume that the analog beamformer is implemented by simple phase shifters with adjustable phase and nonadjustable amplitude. Under this assumption, the elements of the analog beamforming vector $\mathbf{v}_{\mathrm{A}}$ are constrained by constant modulus, i.e., $|[\mathbf{v}_{\mathrm{A}}]_i|^2 = 1, \forall i = 1, 2, \ldots, N_t$. The goal is to find the feasible beamformer by maximizing the SE of the MISO system subject to the constant modulus constraint and transmit power constraint, which is formulated as

$$\max_{\mathbf{v}} \quad \log_2\left(1 + \frac{|\mathbf{h}^H \mathbf{v}_{\mathrm{A}} v_{\mathrm{D}}|^2}{\sigma^2}\right) \tag{2a}$$

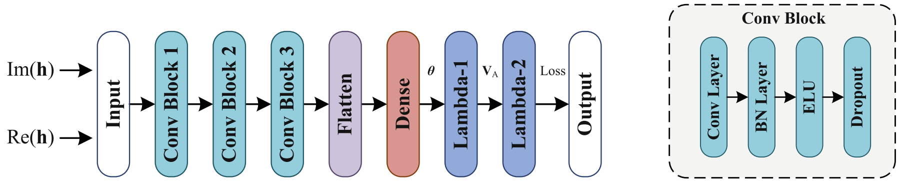
$$s.t. \quad |\mathbf{v}_{\mathrm{A}} v_{\mathrm{D}}|^2 \le P_{max}, \tag{2b}$$

$$|[\mathbf{v}_{\mathrm{A}}]_i|^2 = 1, \forall i = 1, 2, \ldots, N_t. \tag{2c}$$

Since $\|\mathbf{v}_{\mathrm{A}}\|_F^2 = N_t$, the constraint term (2b) is equivalent to $|v_{\mathrm{D}}|^2 \le P_{max}/N_t$. Moreover, the rate function is monotone increasing on $|v_{\mathrm{D}}|^2$, which means the equality of (2b) must be satisfied, otherwise the rate can be further improved by increasing the transmit power. The optimal digital precoding parameter is then given by $v_{\mathrm{D}}^* = \sqrt{\frac{P_{max}}{N_t}}$. As a result, HBF optimization problem (2) is regenerated to find the optimal analog beamforming vector, which is written as

$$\max_{\mathbf{v}_{\mathrm{A}}} \quad \log_2\left(1 + \frac{P_{max}|\mathbf{h}^H \mathbf{v}_{\mathrm{A}}|^2}{N_t \sigma^2}\right) \tag{3a}$$

$$s.t. \quad |[\mathbf{v}_{\mathrm{A}}]_i|^2 = 1, \forall i = 1, 2, ..., N_t. \tag{3b}$$



(a) Network Structure (b) Conv Block

Fig. 2. The proposed neural network architecture for hybrid beamforming design.

Problem (3) is still nonconvex due to the constant modulus constraint thus hard to solve. Recently, a FCNN-based deep learning method is proposed to solve it in [18]. Although the FCNN-based method is verified to be effective in finding a solution, it is not known whether better solutions can be achieved by other deep learning methods. This motivates our work in this paper to develop a different neural network architecture on top of CNN to solve the HBF optimization problem (3).

3 Proposed CNN-Based Hybrid Beamforming Optimization

In this section, we propose a CNN-based framework to solve the HBF optimization problem. CNN is chosen since it not only has better feature extraction capability, but also can reduce the number of learning parameters by sharing weights and biases through convolution kernels, which is potential in improving performance with low computational complexity. We also derive an unsupervised scheme to train the CNN.

3.1 CNN Structure

Our designed CNN structure is shown in Fig. 2, which consists of an input layer, multiple convolutional (Conv) blocks, a fully connected layer, two self-defined layers, and an output layer. We adopts three Conv blocks for feature extraction. Each Conv block includes a Conv layer, a batch normalization (BN) layer, an activation layer and a dropout layer inside. The hyperparameter settings of each layer are shown in Table 1. A brief description of these network layers is given below.

1. Input Layer: A three-dimensional (3-D) matrix with real numbers of size $1 \times N_t \times 2$, as the input of the first Conv layer. To simplify complex operations, we convert the complex CSI vector $\mathbf{h}$ into its corresponding real part and imaginary part, which is fed to the neural network. In this work, we split the real and imaginary part of each complex channel vector and rearrange them into a 3-D real matrix with size $1 \times N_t \times 2$ in an element-wise manner.

Table 1. Parameters of the proposed DL-based HBF model

Layer	$N_o \times C_o$	Activation func.	Number of params. (when $N_t = 64$)
Input	$N_t \times 2$	–	0
Conv Block 1	$(N_t - 2) \times 16$	ELU	176
Conv Block 2	$(N_t - 4) \times 8$	ELU	424
Conv Block 3	$(N_t - 6) \times 4$	ELU	116
Flatten	$4\,(N_t - 6)$	–	0
Dense	$N_t \times 1$	Sigmoid	14912
Lambda-1	$N_t \times 1$	–	0

2. Conv Blocks: The Conv layer takes the input signal and convolves it by convolution kernels to produce the output signal. Specifically, The Conv layer employs C_o kernels of size 1×3 with stride 1 to perform feature extraction for the real and imaginary parts of the input channel matrix, respectively. The BN layer normalizes the output of the Conv layer. BN is a regularization technique that prevents overfitting and achieves faster learning, thus accelerating convergence. The Exponential Linear Units (ELU) activation function performs activation on the output of the BN layer. It can alleviate the gradient disappearance problem by positive value identification. And it has better robustness to negative value input. After that, Dropout layer is added, which is a technique to force the output of some neurons to zero with random probability. The random probability is set to 0.05 to avoid the over-regularization problem.
3. Flatten Layer: After extracting the features from the CNN blocks, the Flatten layer converts these multi-dimensional features into a one-dimensional vector. In order to improve convergence, we add a BN layer after the Flatten layer, which is omitted in Fig. 2 for simplicity.
4. Dense Layer: The Dense layer consists of N_t neurons, which is connected to the outputs of the Flatten layer. The output of the Dense layer corresponds to the phase vector $\boldsymbol{\theta}$ of the analog beamformer, which can be used to construct the analog beamformer through the relationship of $\mathbf{v}_{\mathrm{A}} = e^{j2\pi\boldsymbol{\theta}}$. The sigmoid activation function is used to map the output of the neurons of the Dense layer to the range of (0, 1). The activated output vector of this layer is denoted as $\mathbf{c}_o = \mathrm{Sig}\,(x)\,(\mathbf{W}_o\mathbf{c}_i + \mathbf{b}_o)$, where $\mathrm{Sig}\,(x) \triangleq \frac{1}{1+e^{-x}}$ denotes the sigmoid activation function, $\mathbf{c}_o \in \mathbb{R}^{N_t \times 1}$, $\mathbf{W}_o \in \mathbb{R}^{N_t \times 4(N_t-6)}$, $\mathbf{c}_i \in \mathbb{R}^{4(N_t-6)\times 1}$ and $\mathbf{b}_o \in \mathbb{R}^{N_t \times 1}$ represent output vector, weight matrix, input vector and bias vector of this layer, respectively.
5. Lambda Layers: Since we expect to obtain the analog beamformer through the relationship $\mathbf{v}_{\mathrm{A}} = e^{j2\pi\boldsymbol{\theta}}$, we devise a Lambda layer for such a transform, which is named as Lambda-1 in Fig. 2 and the output of which is $\mathbf{v}_{\mathrm{A}}$. Through the Lambda-1 layer, we map the real values of $\boldsymbol{\theta}$ into complex values of $\mathbf{v}_{\mathrm{A}}$. Moreover, we further devise the Lambda-2 layer to convert the analog

beamformer $\mathbf{v}_{\mathrm{A}}$ into a real value through a function $F_{\mathrm{Loss}}\left(v_{\mathrm{D}}^{*}, \mathbf{v}_{\mathrm{A}}\right) \triangleq -R$, which denotes the loss function and is defined as the negative of the rate function. We note that the output layer is used also as the loss function, this is a key point to design an unsupervised training scheme, which will be described in the following.

3.2 Training Strategy

The goal of the training is to find the feasible analog beamformer by maximizing the SE. The channel samples are fed into the proposed CNN-based model in batches for offline training. Note that the training weights are saved during the training process. We train the proposed CNN-based model with 1000 epochs and there are 16 batches per epoch. We use the Adam optimizer to update the network parameters such as weights and biases with the initial learning rate of 0.01. A learning rate dynamic decay strategy is also used. Specifically, if no improvement in model performance was seen in each 20 epochs, the learning rate was reduced by the factor of 0.2. We train the network using the unsupervised learning mechanism, which is achieved through the Lambda-2 layer. Recall that the Lambda-2 is designed to be the loss function, the output of which is the negative of the rate. By defining such a Lambda function, we can train CNN without using the label data, i.e., the optimal analog beamformers for the input CSI samples, and thus achieve unsupervised learning for the constructed network. The parameters of the CNN network are then optimized though batch optimization. For the given training batch, the parameters are updated by minimizing the loss

$$F_{\mathrm{Loss}} = -\frac{1}{N}\sum_{n=1}^{N}\log_2\left(1+\frac{\gamma_n|\mathbf{h}_n^H\mathbf{v}_{\mathrm{A}}^{(n)}|^2}{N_t}\right) \tag{4}$$

where N denotes the total number of training samples in a batch. $\gamma_n = \frac{P_{max}}{\sigma^2}$, $\mathbf{h}_n$, $\mathbf{v}_{\mathrm{A}}^{(n)}$ represent the SNR value, channel vector and analog beamforming vector of the n-th sample in the training batch.

3.3 Complexity Analysis

Considering only the online stage, we compare the complexity of the proposed CNN-based HBF scheme, the FCNN-based scheme [18] and traditional HBF scheme [9,10] in terms of the number of parameters and FLOPs. Assume that the number of input neurons in each layer is N_i, the number of output neurons is N_o, the number of input channels is C_i, and the number of output channels is C_o. Each Conv layer consists of C_o kernels of size $1 \times z$, where we set $z = 3$, and we also learn that $N_o = N_i - z + 1$ for each Conv layer. When calculating FLOPs, we consider bias, so the number of FLOPs in the Conv layer is $2 \times z \times C_i \times C_o \times N_o$, and the number of FLOPs in the Dense layer is $2 \times N_i \times N_o$. According to the parameters shown in Table 1, it can be calculated that the total number of FLOPs for the proposed CNN-based algorithm is about 0.09 million, the number of FLOPs for the FCNN-based algorithm [18] is around 0.15 million while $N_t = 64$. However, the traditional HBF schemes such as [9,10] have higher complexity due to a large number of complex iterative operations, and the number of FLOPs is approximately 0.26 million.

4 Simulation Results

We consider a downlink MISO system model where a BS equipped with N_t transmit antennas and one RF chain serves a single-antenna user for HBF design. This section compares the performance of the proposed CNN-based HBF algorithm with the full digital beamforming algorithm and two traditional HBF schemes [9,10], using simulation experiments. Furthermore, we refer to the network architecture in [18] and comparison with the FCNN-based HBF algorithm. To ensure the generality of the network, we have given different realizations of $\mathbf{h}$ to construct two datasets, each consisting of 100 channel samples. 90% of the first dataset is selected as the train set for training the network model, and the remaining 10% is used as the validation set. The validation set is used to adjust the hyperparameters of the neural network during the training process to maximize the generalization ability of the model to achieve accurate prediction of new data. The second dataset is used as the test set to evaluate the final performance of the model. The simulation experiment environment is deployed on a computer with Windows 10 OS as well as NVIDIA GeForce GTX 1650 GPU and Intel(R) Core(TM) i7-10750 CPU, and the model training is based on Python 3.7 and Tensorflow 2.0.0. All simulation results are obtained by taking the average of all channel realizations.

Specifically, we use two typical channel models (i.e., Rayleigh fading channel and geometric mmWave channel) as the channel $\mathbf{h}$ between BS and user for correlation simulations. We assume the elements of Rayleigh fading channel are independent and identically distributed (i.i.d.) zero-mean circularly symmetric complex Gaussian random variables. Besides, we adopt a geometric mmWave channel model with limited clusters, which can be expressed as

$$\mathbf{h}^H = \sqrt{\frac{N_t}{L}} \sum_{l=1}^{L} \alpha_l \mathbf{a}_t^H \left(\theta_l\right), \tag{5}$$

where $L = 3$ denotes the number of clusters between the BS and the user. $\alpha_l \sim \mathcal{CN}(0,1)$ stands for the complex gain of the l-th cluster. $\mathbf{a}_t\left(\theta_l\right)$ indicates the transmitting antenna array response vector at the BS, and furthermore θ_l is the azimuth angles of departure(AoD) of the l-th cluster.

The learning rate setting is crucial when training the model, since it controls the magnitude of parameters updated per time. Figure 3 shows the SE performance versus SNR of the proposed scheme with various learning rates in large geometric mmWave channel with $N_t = 64$. Obviously, the learning rate setting of 0.01 has the highest SE value, while setting it to 0.5 or 0.0001 will not give excellent SE performance. This is because too high the learning rate will cause larger update amplitude and the parameters to be optimized fluctuate around the minimum value and do not converge, while too low will cause converge slowly.

Figure 4 illustrates the convergence performance of the proposed CNN structure with 1000 epochs and learning rate of 0.01 in large geometric mmWave channel with $N_t = 64$. At the beginning of the training stage, the training weight parameters are not optimal, thus the loss value for the first few epochs are quite large. As the training proceeds, the parameters tend to be optimal and the loss decreases abruptly. After that, the system loss function tends to be stable with very small fluctuations and low loss value.

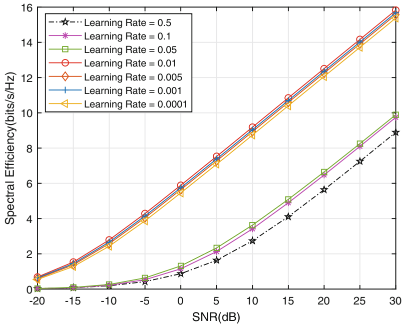


Fig. 3. SE performance versus SNR of the proposed scheme under various learning rates in geometric mmWave channel with $N_t = 64$.

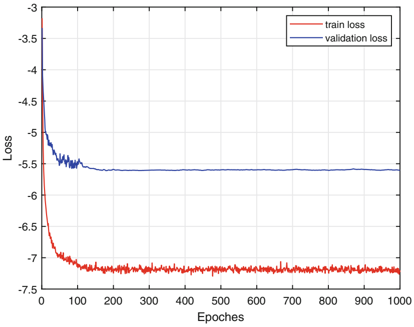


Fig. 4. Convergence performance of the proposed scheme in geometric mmWave channel with $N_t = 64$.

Figure 5 gives the comparison of SE performance under different beamforming schemes in large Rayleigh fading channel with $N_t = 64$. The full digital beamforming method provides higher SE compared to HBF schemes. It can be seen that under the same channel samples, the proposed CNN-based HBF scheme achieved better performance than traditional HBF iterative algorithms, and obtained higher SE than the FCNN-based scheme. Furthermore, except for Rayleigh fading channels, our proposed HBF design scheme is also applicable to mmWave channel with limited clusters. Figure 6 compares the performance of the proposed algorithm with other beamforming algorithms in large geometric mmWave channel when $N_t = 64$. With increasing SNR, the performance of our proposed algorithm is followed only by the fully digital beamforming algorithm and has much higher SE than other HBF algorithms. As mentioned above, in both classical channel scenarios, our proposed CNN-based scheme has higher SE performance compared to the traditional HBF algorithms and FCNN.

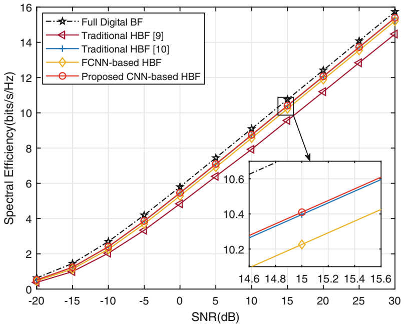


Fig. 5. Comparison of spectral efficiency performance under different schemes in Rayleigh fading channel with $N_t = 64$.

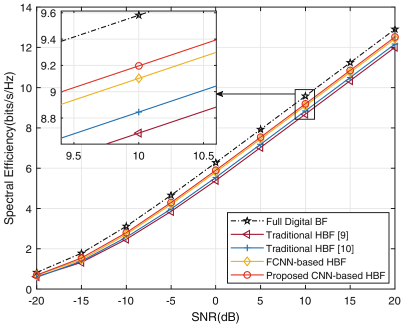


Fig. 6. Comparison of spectral efficiency performance under different schemes in geometric mmWave channel with $N_t = 64$.

Finally, the detailed complexity comparison when $N_t = 64$ is shown in Table 2. The analysis of the number of parameters and FLOPs shows the great superiority of the proposed CNN-based scheme over other scheme in terms of complexity. Moreover, the significant reduction in complexity leads to the increase in execution speed. We also compare the average execution time of the proposed CNN-based HBF scheme, the FCNN-based scheme as well as the two traditional schemes, as shown in Table 3, where we set $N_t = 64$. It can be noticed that the traditional scheme [9] has the highest execution time, followed by scheme [10]. The execution times of the two HBF traditional schemes are much higher than the two schemes using DL. Since FCNN has the ability of global perceptive, FCNN has a serious issue, i.e., there are too many parameters. While CNN can achieve local perception, the weights of different neurons in the Conv layer are shared, which greatly reduces the parameters and improves the training performance of the whole network, and can extract features more effectively. Meanwhile, CNN can handle the coupling between different elements more efficiently than FCNN [20]. It is shown that the proposed CNN model obtained superior performance compared to FCNN.

Table 2. Complexity comparison

HBF scheme	Number of params.	Number of FLOPs
Proposed CNN-based	16556	0.09 million
FCNN-based [18]	75720	0.15 million
Traditional	–	0.26 million

Table 3. Execution time comparison

HBF scheme	Execution time
Proposed CNN-based	0.3223 s
FCNN-based [18]	0.3338 s
Traditional [9]	11.9553 s
Traditional [10]	9.5333 s

5 Conclusion

In this paper, we proposed a low-complexity HBF optimization scheme for the downlink MISO system, which employed a CNN-based network framework and used an unsupervised learning mechanism for training. The simulation results demonstrated the effectiveness of the scheme. The proposed scheme was compared with other existing works in terms of complexity and SE performance. Our proposed CNN-based HBF algorithm achieved higher SE performance with lower complexity compared to traditional HBF algorithms and FCNN. The work we have done provides a novel solution and offers an effective fresh idea for the HBF optimization.

References

1. Niu, Y., Li, Y., Jin, D., Su, L., Vasilakos, A.V.: A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges. Wireless Netw. **21**(8), 2657–2676 (2015). https://doi.org/10.1007/s11276-015-0942-z
2. Kuo, C.H., Chang, H.Y., Chang, R.Y., Chung, W.H.: Unsupervised learning based hybrid beamforming with low-resolution phase shifters for MU-MIMO systems. arXiv preprint arXiv:2202.01946 (2022)
3. Hong, S.H., Park, J., Kim, S.J., Choi, J.: Hybrid beamforming for intelligent reflecting surface aided millimeter wave MIMO systems. IEEE Trans. Wirel. Commun. (2022)
4. Molisch, A.F., et al.: Hybrid beamforming for massive MIMO: a survey. IEEE Commun. Mag. **55**(9), 134–141 (2017)
5. Zhang, X., Molisch, A.F., Kung, S.Y.: Variable-phase-shift-based RF-baseband codesign for MIMO antenna selection. IEEE Trans. Signal Process. **53**(11), 4091–4103 (2005)
6. Mo, J., Alkhateeb, A., Abu-Surra, S., Heath, R.W.: Hybrid architectures with few-bit ADC receivers: achievable rates and energy-rate tradeoffs. IEEE Trans. Wireless Commun. **16**(4), 2274–2287 (2017)
7. El Ayach, O., Rajagopal, S., Abu-Surra, S., Pi, Z., Heath, R.W.: Spatially sparse precoding in millimeter wave MIMO systems. IEEE Trans. Wireless Commun. **13**(3), 1499–1513 (2014)
8. Hung, W.L., Chen, C.H., Liao, C.C., Tsai, C.R., Wu, A.Y.A.: Low-complexity hybrid precoding algorithm based on orthogonal beamforming codebook. In: 2015 IEEE Workshop on Signal Processing Systems (SiPS), pp. 1–5. IEEE (2015)
9. Yu, X., Shen, J.C., Zhang, J., Letaief, K.B.: Alternating minimization algorithms for hybrid precoding in millimeter wave MIMO systems. IEEE J. Sel. Top. Signal Process. **10**(3), 485–500 (2016)
10. Sohrabi, F., Yu, W.: Hybrid digital and analog beamforming design for large-scale antenna arrays. IEEE J. Sel. Top. Signal Process. **10**(3), 501–513 (2016)
11. Ren, Y., Wang, Y., Qi, C., Liu, Y.: Multiple-beam selection with limited feedback for hybrid beamforming in massive MIMO systems. IEEE Access **5**, 13327–13335 (2017)
12. Xiong, Z., Cai, Z., Takabi, D., Li, W.: Privacy threat and defense for federated learning with non-iid data in AIoT. IEEE Trans. Industr. Inf. **18**(2), 1310–1321 (2022)
13. Cai, Z., Xiong, Z., Xu, H., Wang, P., Li, W., Pan, Y.: Generative adversarial networks: a survey toward private and secure applications. ACM Comput. Surv. **54**(6), 1–38 (2021)
14. Xu, H., Cai, Z., Li, R., Li, W.: Efficient CityCam-to-Edge cooperative learning for vehicle counting in ITS. IEEE Trans. Intell. Transp. Syst. (2022)
15. Alkhateeb, A., Alex, S., Varkey, P., Li, Y., Qu, Q., Tujkovic, D.: Deep learning coordinated beamforming for highly-mobile millimeter wave systems. IEEE Access **6**, 37328–37348 (2018)
16. Huang, H., Song, Y., Yang, J., Gui, G., Adachi, F.: Deep-learning-based millimeter-wave massive MIMO for hybrid precoding. IEEE Trans. Veh. Technol. **68**(3), 3027–3032 (2019)
17. Xia, W., Zheng, G., Zhu, Y., Zhang, J., Wang, J., Petropulu, A.P.: A deep learning framework for optimization of MISO downlink beamforming. IEEE Trans. Commun. **68**(3), 1866–1880 (2020)

18. Lin, T., Zhu, Y.: Beamforming design for large-scale antenna arrays using deep learning. IEEE Wireless Commun. Lett. **9**(1), 103–107 (2020)
19. Attiah, K.M., Sohrabi, F., Yu, W.: Deep learning approach to channel sensing and hybrid precoding for TDD massive MIMO systems. In: 2020 IEEE Globecom Workshops (GC Wkshps), pp. 1–6. IEEE (2020)
20. Song, H., Zhang, M., Gao, J., Zhong, C.: Unsupervised learning-based joint active and passive beamforming design for reconfigurable intelligent surfaces aided wireless networks. IEEE Commun. Lett. **25**(3), 892–896 (2021)
21. Liang, Y., Cai, Z., Yu, J., Han, Q., Li, Y.: Deep learning based inference of private information using embedded sensors in smart devices. IEEE Netw. **32**(4), 8–14 (2018)

An Adaptive BSCO Algorithm of Solid Color Optimization for 3D Reconstruction System with PIFuHD

Chao-Hsien Hsieh(✉), Yubo Song, Zhen Wang, and Changfeng Li

Qufu Normal University, Jining 273165, Shandong, China
george_hsieh@qq.com

Abstract. PIFuHD can generate high-resolution model in the process of human 3D reconstruction. However, PIFuHD will produce debris outside the human body when the image background is more complex. In order to solve this problem, this paper develops an adaptive BSCO algorithm for the background of human body and image of human body in 3D reconstruction system. The BSCO algorithm is divided into four steps in processing. First, BSCO algorithm uses Go-selfies to separate the background. Second, BSCO algorithm converges the RGB of all pixels of the character into a set. Third, BSCO algorithm finds the greatest difference from the set through HSV conversion. Fourth, BSCO algorithm weighs the set and then calculates the RGB score. The highest score of RGB is used as the RGB of the background after solid color optimization. The experimental results show that the proposed method improves the reconstruction effect of PIFuHD.

Keywords: PIFuHD · 3D reconstruction · Image processing · Background adaptive

1 Introduction

1.1 A Subsection Sample

Nowadays, the intelligent devices have been popularized to all walks of life, such as clothing industry, medicine, etc. This makes 3D human modeling of images not only a reality, but also a trend of universal application. The relevant applications are as follows. Combining 3D reconstruction and 3D topology can handle 3D fingerprint recognition [1]. The volumetric network predicts the animated skeleton of the 3D joint model to help the doctor's treatment [2]. 3D reconstruction could identify hidden 3D living space for search and rescue management [3]. These applications have higher requirements for the completeness and accuracy of the reconstruction of the human body model.

Until this moment, the best 3D reconstruction algorithm for the image of human body is PIFuHD proposed by Shunsuke Saito et al. [4]. It is a multi-level pixel alignment implicit function for high-resolution 3D human body digitization. It solves the limitations of deep neural networks in the field of 3D reconstruction by developing an end-to-end trainable multi-level architecture.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 16–27, 2022.
https://doi.org/10.1007/978-3-031-19214-2_2

However, all 3D reconstruction methods including PIFuHD have a problem that the reconstruction effect is greatly affected by background factors. The background mentioned in this article shows the person/object in the photo such as tennis player will be the foreground of the photo. Also, everything related to the person including their items, clothes, etc. will not be regarded as the background. Due to the complexity of the real world, the 3D reconstruction effect of the picture is poor in some cases. For example, the reconstruction of athletes wears green sportswear on the green field. Affected by the background color, the current algorithm is hard to distinguish which part is the green field and which part is the athlete.

Therefore, background processing plays a very important role in the reconstruction process. To solve the above problems, this paper proposes the Background Solid Color Optimization algorithm(BSCO). This method reduces the influence of background factors on the reconstruction effect by optimizing the background.

The contributions of this paper are summarized as the following as two parts.

1. This paper firstly proposes BSCO algorithm. It reduces the influence of background factors on 3D reconstruction by optimizing the original image. BSCO algorithm can be used for 3D reconstruction of human body in any scene.
2. This paper improves PIFuHD algorithm. The BSCO algorithm combines PIFuHD to make the output of the final 3D reconstruction method more accurate.

The structure of this paper is as follows. The second part is related work, which introduces the 3D modeling technology and lists the relevant symbols of the formula. The third section introduces the method of this paper. The fourth part is the experimental results and analysis. Finally, the fifth part presents conclusions and future research directions.

2 Related Work

2.1 Research Status of 3D Reconstruction Technology

Early 3D reconstruction, such as [5, 6], uses a set of simple geometric elements for reconstruction. Also, some models can show human facial expressions [7]. Later, some papers mention that deep neural network is used for reconstruction [8, 9]. These methods are only limited for the bare human body, not for reconstructed clothes or accessories. In order to solve this problem, [10] proposes a contour based on reconstruction method of dressed human body. Up to now, many methods have solved the problems of memory efficiency and resolution through implicitly defined continuous neural representation. PIFuHD is based on the Pixel-Aligned Implicit Function (PIFu) framework. In order to get higher resolution outputs, PIFuHD superimposes an additional pixel alignment prediction module on this framework. In the prediction module, the fine module inputs an image with a resolution of 1024×1024 and then encodes it into an image feature with a resolution of 512×512 [4].

2.2 Symbol Table

This section summarizes the related symbols and meanings as shown in Table 1.

Table 1. Symbol table.

Symbol	Meaning
y_P	Predictive value
y_t	The label of the corresponding pixel
I_L	Low-resolution image
F_L	Frontal normal graph
B_L	Negative normal graph
g^L	Multi layer perceptron for processing low-pixel images
Z	Depth under the camera's visual angle
I_H	High-resolution image
F_H	High-resolution frontal normal map
B_H	High-resolution negative normal map
g^H	Multi layer perceptron for processing high-pixel images
$\Omega(X)$	Global characteristics of the previous stage
x_{new}	Performance evaluation results
x_{av}	Average value of single experiment
x_{min}	Minimum value during a single experiment
x_{max}	Maximum value during a single experiment
BG_{RGB}	Final background color

3 Method

The algorithm proposed in this paper is an improvement of PIFuHD algorithm, which combines BSCO and PIFuHD. Part A introduces the overall framework of the algorithm. Part B describes the BSCO algorithm proposed in this paper. And then, part C introduces PIFuHD algorithm.

3.1 PIFuHD Algorithm Framework for Solid Color Optimization

The algorithm proposed in this paper consists of two parts. First, the background is adaptive conversion based on BSCO algorithm. The BSCO algorithm is divided into four steps, background removal, RGB conversion, weight processing, and weight statistics. Second, the function of PIFuHD algorithm is reconstruction. Also, the image processed by BSCO is used as the input of PIFuHD. Finally, PIFuHD outputs the reconstructed model.

As shown in Fig. 1, compared with the original PIFuHD algorithm, the algorithm proposed in this paper has four steps to process the original image. There are Go-selfies, RGB conversion function $\Gamma(\text{RGB})$, weight processing function $\Psi(\Pi)$, and statistical function of RGB score $\delta(\Lambda)$. After the above four steps, the algorithm can generate 1024 * 1024 high-resolution image and 512 * 512 low-resolution image. As shown in

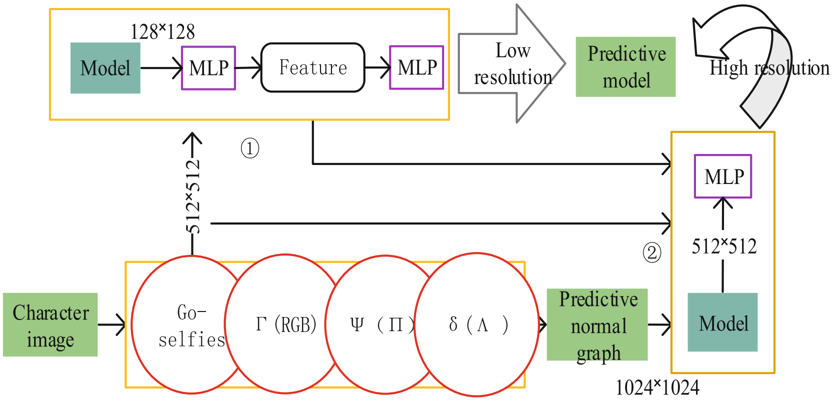


Fig. 1. PIFuHD algorithm framework for solid color optimization

① and ②, the algorithm respectively generates the image features of both 512 * 512 and 128 * 128 after coding. As long as passing through the multi-layer perceptron, the 3D embedding generated by 128 * 128 image features is combined with the 3D embedding generated by 512 * 512 image features. The high-resolution result is generated through the multi-layer perceptron. Also, the 3D embedding generated by 128 * 128 image features produces low resolution results through multi-layer perceptron.

3.2 BSCO

This section introduce BSCO algorithm which flow chart is shown as Fig. 2. The process is divided into four steps as below.

Step (1) Background Removal. This step removes the background of the image and reduces the influence of background factors on 3D reconstruction algorithm. The Go-selfies are used by BSCO for background removal. It uses the model architecture of encoder and decoder. The encoder is based on VGG16 and ResNet34 classification models. The decoder is composed of up-sampling feature map and corresponding feature map of decode. Go-selfies find the best learning rate through the circular learning method and fine tune the model [13]. It establishes eight models for experiments according to different resolutions and encoders. The best model is determined by comparing dice coefficient, binary cross entropy loss, and confusion matrix [14]. The specific evaluation indicators are as follows.

Dice coefficient. It needs predicted data and real data. The closer the coefficient is to 1, the better the performance of the model. Both |X| and |Y| represents the number of pixels in the foreground. The definition is as following as formula (1).

$$\mathrm{Dice} = \frac{2|\mathrm{X} \cap \mathrm{Y}|}{|\mathrm{X}| + |\mathrm{Y}|} \tag{1}$$

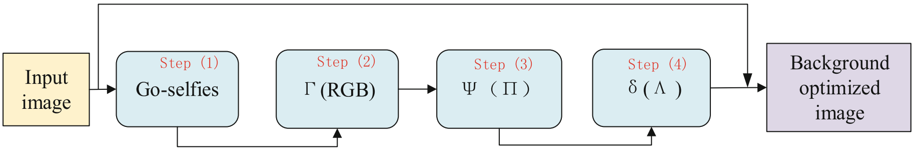


Fig. 2. The flow chart of BSCO algorithm

Binary cross entropy loss. The y_p is the predicted value and y_t is the label of the pixel. The loss function is defined as formula (2).

$$\text{loss} = (-y_t \log(y_p) - (1 - y_t)\log(1 - y_p)) \tag{2}$$

Confusion matrix. The N is the number of images as defined in formula (3).

$$\text{Accuracy} = \frac{\sum N \frac{\text{Correct predicted pixels}}{\text{All pixels in each image}}}{N} \tag{3}$$

Step (2) RGB Conversion. This step converts RGB values of pixels to HSV values and applies to supplement the preprocessing of the character background. Function Γ(RGB) can return an RGB set Π. The element of the set has the largest difference from the RGB of the pixel corresponding to the character. First, it converts the RGB of each pixel of the character into HSV. Then, it set $R_i^{'} = \frac{R_i}{255}, G_i^{'} = \frac{G_i}{255}, P_i^{'} = \frac{P_i}{255}$, $\text{Cmax} = \max(R^{'}, G^{'}, B^{'})$, and $\text{Cmin} = \min(R^{'}, G^{'}, B^{'})$. RGB conversion is shown in formulas (4), (5), and (6).

$$H = \begin{cases} 0, \text{ if Cmax} = \text{Cmin} \\ 60 \times (\frac{G^{'}-B^{'}}{\text{Cmax}-\text{Cmin}} + 0), \text{ if Cmax} = R^{'} \\ 60 \times (\frac{B^{'}-R^{'}}{\text{Cmax}-\text{Cmin}} + 2), \text{ if Cmax} = G^{'} \\ 60 \times (\frac{R^{'}-G^{'}}{\text{Cmax}-\text{Cmin}} + 4), \text{ if Cmax} = B^{'} \end{cases} \tag{4}$$

$$S = \begin{cases} 0, \text{ if Cmax } = 0 \\ \frac{\text{Cmax}-\text{Cmin}}{\text{Cmax}}, \text{Otherwise} \end{cases} \tag{5}$$

$$V = \text{Cmax} \tag{6}$$

The HSV is then processed to set. $H_i = \lfloor \frac{H}{60} \rfloor \text{mod} 6$, $P = V \times (1 - S)$, $q = V \times (1 - (\frac{H}{60} - H_i) \times S)$, $t = V \times (1 - (1 - \frac{H}{60} + H_i) \times S)$. And then, the formula (7) converts HSV to RGB.

$$(R, G, B) = \begin{cases} (V, t, p), \text{ if H } = 0 \\ (q, V, p), \text{ if H } = 1 \\ (p, V, t), \text{ if H } = 2 \\ (p, q, V), \text{ if H } = 3 \\ (t, p, V), \text{ if H } = 4 \\ (V, p, q), \text{ if H } = 5 \end{cases} \tag{7}$$

Step (3) Weight Processing. This step assigns weights for each element of RGB set after Step (2). The RGB weights of each part of the human body are different. The function of character and background is very important for 3D modeling. Therefore, BSCO algorithm does weight processing for different parts of character.

As shown in Fig. 3, the weights of different parts of character are 0.6, 0.3, and 0.1 from outside to inside. The outer layer of character has the greatest influence on the modeling results. So it occupies the largest weight. The middle layer and inner layer still have an impact on the modeling results. For example, the color of the clothes is the same as the background color. At this time, the middle layers and inner layers will affect the algorithm's judgment on the outline and details of the character. However, this effect is mainly reflected in the reconstruction of character details. The middle layers and inner layers have little effect on the algorithm's judgment of the character contour, so the total impact is 0.4. The calculation is shown in formula (8). In the formula (8), Π_{i} is the RGB set of the layer i.

$$\Lambda = \sum\nolimits_{i=1}^{3} \Pi_{i} \times W_{i} \tag{8}$$

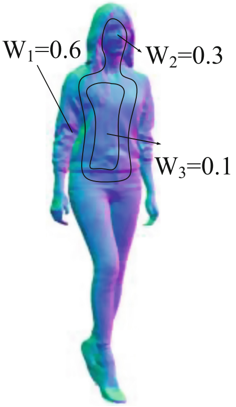


Fig. 3. Weight distribution graph

Step (4) Weight Statistics. Based on the result of Step (3), this step finds a maximum value of weight for the final image background. The function of $\delta(\Lambda)$ is to count the scores of all RGBs in set Λ. The highest score is the solid color of the character image to optimize the background color. The pseudo code is as follows.

Algorithm Statistics of background adaptation results

```
Input: The result set obtained by weighting $\Lambda$
Output: Final background color $BG_{RGB}$
Define $\kappa$ as the set of RGB that exists in $\Lambda$
for $rgb_1$ in $\Lambda$
  for $rgb_2$ in $\kappa$
    if $rgb_1 = rgb_2$
    The corresponding RGB in $\kappa$ plus the $rgb_1$ score
    end if
  end for
end for
Define $BG_{RGB}$ and set the initial value to 0
for rgb in $\kappa$
  if rgb > $BG_{RGB}$
    $BG_{RGB}$
  end if
end for
return $BG_{RGB}$
```

The pseudocode is described as follows. First, the algorithm defines κ to include all RGB in Λ. All elements in κ can not be repeated. Next, the algorithm traverses all RGBs in set Λ and calculates the score of each RGB into Λ to generate a complete κ. Finally, the algorithm returns the RGB with the highest score in κ. This score is the RGB of the dynamically optimized background color.

3.3 PIFuHD

PIFuHD realizes 3D modeling through rough estimation and fine reconstruction. First, in the rough estimation operation, the resolution of the input image is 512 × 512. After processing, it obtains the embedded features of 128 × 128. In the fine reconstruction operation, the resolution of the input image is 1024 × 1024. After processing, it obtains the embedded features of 512 × 512. Then, the extracted high-resolution embedded features and the roughly estimated low-resolution embedded features are used to predict the position of the three-dimensional point [4]. PIFuHD improves the traditional modeling method and proposes a multi-layer method. It inputs high-resolution images to establish a high-precision model. This algorithm is divided into two stages.

1) The first stage. The original input image is sampled to generate a 512 × 512 low-resolution image. A 128 × 128 low-resolution image feature is achieved through the model. At the same time, the input includes both front and back normal map as following by the formula(9). The $X_L(x_L \in R^2)$ is the position of the 3D point X image space projection.

$$f^L(X) = g^L(\Phi^L(x_L, I_L, F_L, B_L,), Z) \tag{9}$$

2) The second stage. The resolution of the input image is 1024 × 1024. After processing, it obtains the image feature model of 512 × 512. This is different from the first stage resolution 128 × 128 image feature. The 512 × 512 resolution image feature obtains the three-dimensional model fine details. Both the original 1024 × 1024 resolution image and 512 × 512 resolution image features need to be input together, as well as high-resolution frontal and backside normal maps. Because the second stage is based on the features of the first stage construction, so the result is better than the output of the first stage. The fine level is denoted as belows.

$$f^H(X) = g^H(\Phi^H(x_H, I_H, F_H, B_H,), \Omega(X)) \tag{10}$$

In the formula (10), $X_H(x_H \in R^2)$ is the high-resolution 2D projection position. Function Φ^H extracts image features from high-resolution input and its structure is similar to low-resolution feature extractor Φ^L.

4 Experiment

This experiment compares the modeling results preprocessed by BSCO algorithm with the modeling results without preprocess. At the same time, this paper analyzes the model reconstruction effect, face details, time consumption, and CPU utilization.

4.1 Experimental Environment

The hardware configuration of the experiment is Intel i5-7200u processor, NVIDIA GTX 1650 4 GB video memory. The test set is 1000 human body images randomly selected from the COCO [15] data set.

4.2 Experimental Results

The results of modeling by PIFuHD are compared with the results of modeling after BSCO algorithm preprocessing, as shown in Fig. 4(a), (b), (c), and (d).

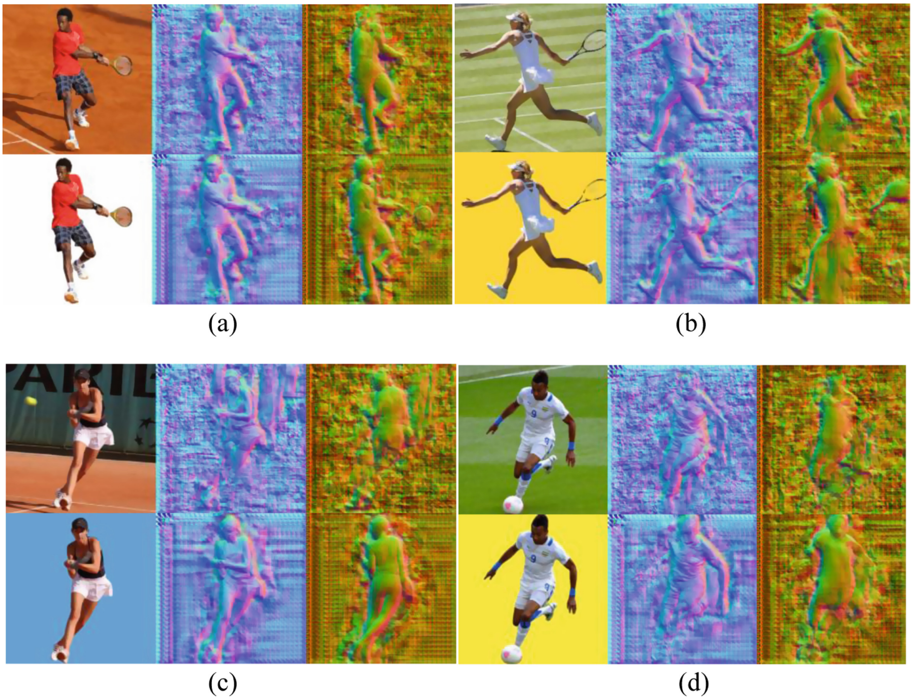


Fig. 4. Comparison of modeling results

4.3 Experimental Analysis

1) Data analysis. The comparison of reconstruction time, reconstruction effect, and quantitative results is analyzed as follows.

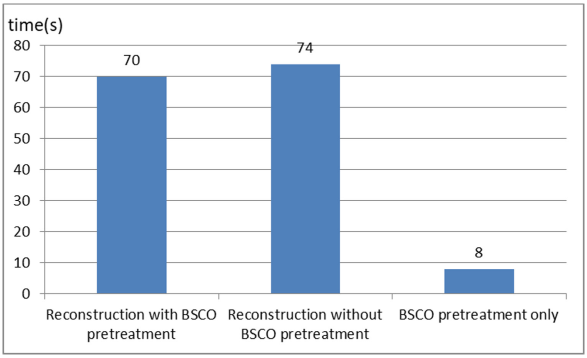


Fig. 5. Time consuming experimental results

a) Reconstruction time

As shown in Fig. 5, the reconstruction with BSCO pretreatment took 70 s. The reconstruction without BSCO pretreatment took 74 s. The BSCO pretreatment only took 8 s.

b) Reconstruction effect

It can be seen from Fig. 4 that the 3D reconstruction effect of human body has been improved to a certain extent after background optimization. For example, the prediction of the back of the human body in Fig. 4(a) is more precise. The reconstruction of the tennis racket can be shown in Fig. 4 (b). And, both the front and back of the human body in Fig. 4(c) have a good reconstruction effect, which reflects the better results of this paper.

Table 2. Quantitative results.

Methods	Norm Cosine ↓	Norm L2 ↓	CD (cm)↓	Occ L1 ↓
PIFuHD [4]	0.181	0.544	2.008	5.837
PIFu [11]	0.103	0.376	0.592	2.079
PaMIR [12]	0.097	0.361	0.554	1.977
Our	0.180	0.550	2.002	5.832

c) Comparative analysis of quantitative results

Norm Cosine, Norm L2, CD, and Occ L1 were calculated as shown in Table 2. This experiment consideres the surface normal of the reconstructed mesh and the ground active mesh. It calculates the L2 and cosine distances between Norm Cosine and Norm L2. Also, it calculates the chamfer distance (CD) between the ground live mesh and the reconstructed mesh. The results can be used to measure the overall quality of the reconstruction. Finally, this experiment calculates the average between the predicted and actual ground occupancy (Occ L1), which can evaluate the algorithm's original prediction of the character. Because BSCO algorithm is used for preprocessing before PIFuHD algorithm to reconstruct the picture, the quantitative comparison of the four methods is similar to PIFuHD. The difference for output image is more accurate than PIFuHD.

2) Performance evaluation. This experiment records the average CPU utilization, average RAM usage, and time consumption of each experiment in the process. The min-max formula (11) is used to standardize the three parameters. The evaluation index SUM is used to evaluate the performance of each experiment as shown in formula (12).

$$X_{new} = \frac{x_{av} - x_{min}}{x_{max} - x_{min}} \tag{11}$$

$$SUM = (CO + MFP) \times 0.5 + CPU_{time} \times 0.5 \tag{12}$$

Among them, CO represents the average CPU utilization of a single experiment. MFP represents the average RAM usage of a single experiment. The CPU_{time} shows the completion time.

Table 3. Equipment occupancy.

BSCO algorithm used	CPU	$\mathrm{CPU}_{\mathrm{time}}$	RAM	SUM
N	4.313	7.3	4.60	8.10
Y	7.022	6.7	4.71	9.22

As shown in Table 3, the SUM with BSCO algorithm is 9.22. Also, the SUM without BSCO algorithm is 8.10. From the Table 3, the image using BSCO algorithm for 3D reconstruction has the better effect than image without BSCO algorithm.

5 Conclusion and Future Work

This paper uses the BSCO algorithm to improve the stability of the PIFuHD algorithm for background adaptation. The experimental result represents to BSCO algorithm that can get great 3D reconstruction effect. This paper proves the effectiveness of the 3D reconstruction method. In the future, the 3D reconstruction of human faces should be one of important research topics.

Acknowledgment. This work was supported by the Shandong Provincial Natural Science Foundation of China under Grant (ZR2020MF048). The corresponding author is Chao-Hsien Hsieh. Both Chao-Hsien Hsieh and Yubo Song have contributed equally to be co-first authors.

References

1. Yin, X., Zhu, Y., Hu, J.: 3D Fingerprint recognition based on ridge-valley-guided 3D reconstruction and 3D topology polymer feature extraction. In: IEEE Trans. Pattern Anal. Mach. Intell. **43**, 1085–1091 (2021)
2. Xu, Z., Zhou, Y., Kalogerakis, E., Karan, S.: Predicting animation skeletons for 3D articulated models via volumetric nets. In: International Conference on 3D Vision 2019. 3DV, pp. 298–307 (2019)
3. Panagiotis, M., Nikolaos, D., Anastasios, D., Vassilis, S., Aggelos, A.: Pervasive 3D reconstruction to identify hidden 3D survival spaces for search and rescue management. In: IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, pp. 808–813 (2018)
4. Shunsuke, S., Tomas, S., Jason, S., Hanbyul, J.: PIFuHD: multi-level pixel-aligned implicit function for high-resolution 3D human digitization. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 81–90
5. Plänkers, R., Fua, P.: Tracking and modeling people in video sequences. Comput. Vis. Image Underst. **81**(3), 285–302 (2001)
6. Sminchisescu, C., Triggs, B.: Estimating articulated human motion with covariance scaled sampling. Int. J. Robot. Res. **22**(6), 371–392 (2003)
7. Pavlakos, G., et al.: Expressive body capture: 3d hands, face, and body from a single image. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10975–10985 (2019)

8. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7122–7131 (2018)
9. Xu, Y., Zhu, S.-C., Tung, T.: Denserac: joint 3d pose and shape estimation by dense render-and-compare. In: IEEE/CVF International Conference on Computer Vision, pp. 7759–7769 (2019)
10. Natsume, R., et al.: Siclope: Silhouette-based clothed people. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4480– 4490 (2019)
11. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: IEEE/CVF International Conference on Computer Vision, pp. 2304–2314 (2019)
12. Zheng, Z., Tao, Yu., Liu, Y., Dai, Q.: Pamir: parametric model-conditioned implicit representation for image-based human reconstruction. IEEE Trans. Pattern Anal. Mach. Intell. (to appear, 2021)
13. Smith Leslie, N.: Cyclical learning rates for training neural networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 464–472 (2017)
14. Wu, Y.: Go-selfies: a fast selfies background removal method using ResU-Net deep learning. In: 2020. 28th European Signal Processing Conference (EUSIPCO), pp. 615–619 (2020)
15. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48

RF-Line: RFID-Based Line Crossing Detection

Xuan Wang[1], Kai Xun[1], Xingyu Chen[1], Xia Wang[1,2], Jia Liu[1](✉), and Zhihong Zhao[1,3](✉)

[1] State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing, China
{jialiu,zhaozhih}@nju.edu.cn
[2] School of Computer Engineering, Jinling Institute of Technology, Nanjing, China
[3] Principal of Suzhou City College, Suzhou, China

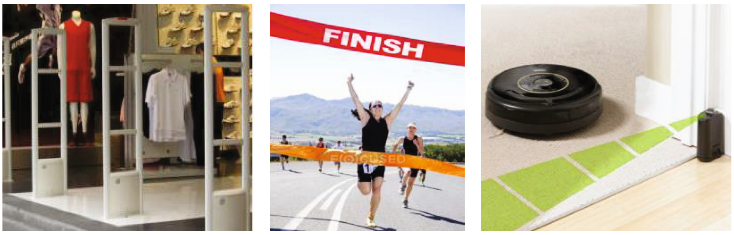
Abstract. Line crossing detection is to check whether people or objects go across a given barrier line, which is quite common and important in our daily life, such as the EAS checkpoint in a retail store or the finish line in track and field. Although existing solutions to line crossing detection have achieved great advancement, they do not function well when multiple objects or people cross the line at the same time. In this paper, we propose a new RFID-based solution called RF-Line to the problem of line crossing detection, especially for multi-object scenarios. The biggest challenge is that the RFID reader's coverage zone is invisible and irregular; we cannot roughly take the time when a tag is seen by the reader for the first time as the time when line crossing occurs. In RF-Line, we deploy two antennas opposite each other and collect the RF phase profiles of two antennas at the same time. By a series of geometric transformations and mathematical derivations, we find that summing up the two phase profiles will get a new phase curve, in which the inflection point of the curve is the time of line crossing. We implement RF-Line with commodity RFID systems. Extensive experiments show that RF-Line can achieve accurate line crossing detection with a small error of 6.1 cm, with no need for any system calibration or complicated deployment.

Keywords: RFID · Line crossing detection · Mobile localization

1 Introduction

Line crossing detection is to check whether and when people or objects go across a given barrier line, which is quite common and important in our daily life. For example, in a retail store, line crossing detection is deployed on the electronic article surveillance system (EAS) for detecting pass in and out of customers [1]. If an attacker passes the EAS door with improperly bought items, the system sets off alarms and alerts staff to an attempted theft in progress. In track and field, high-resolution cameras are used to capture images and compute the time when athletes hit the finish line, which determines the winner and ranking of

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 28–39, 2022.
https://doi.org/10.1007/978-3-031-19214-2_3



(a) EAS System (b) Finish Line (c) Virtual Wall

Fig. 1. Applications of line crossing detection.

the game. In robotics, the technology of line crossing detection can be applied to create a virtual wall, which is an invisible barrier that robots won't cross. That makes it easy to confine a robot to a particular area or room, and prevents it from approaching anything dangerous [2,3], which is shown in Fig. 1.

Existing solutions to line crossing detection generally fall into three categories: infrared sensor [4,5], camera [6,7], and virtual wall [2,3]. Infrared sensor detects line crossing by measuring infrared light radiating from objects in its field of view, which is widely used in security alarms. It tracks general movements but does not tell who or what moved. Camera uses image processing algorithm to recognize the movement of people or objects over a given virtual line. It can be used to detect people crossing over the fence or entering some restricted area. In robotics, a robot constructs the virtual wall, which is an invisible barrier that the robot cannot cross in automatic path planning. Although existing solutions to line crossing detection have achieved great advancement, they do not function well in the case of detecting concurrent line crossing, i.e., multiple objects or people cross the line at the same time.

In this paper, we propose a new solution called RF-Line to the problem of line crossing detection by using Radio Frequency Identification (RFID). As a non-contact passive sensing technology, RFID has attracted increasing attention in recent years and has been widely used in various fields, such as object tracking [8–13], warehouse inventory [14–17], library management [18]. Each RFID tag has a unique ID that exclusively indicates every tagged object and brings them item-level intelligence. By tracking tags in real-time, we can figure out when the tagged people or objects go across a given barrier line, especially for multi-target scenarios. However, this is not easy. The biggest challenge of RF-Line is that the interrogation zone of an RFID reader is irregular and unpredictable since the reader's signals are susceptible to various factors, e.g., the reader planning, multi-path effects, material of tagged objects, surroundings. We cannot roughly take the time when a tag is seen by the reader for the first time as the time when line crossing occurs. Mobile RFID localization is a feasible solution, but it needs complicated system deployment, accurate calibration, or high computation overhead. For example, PinIt and BackPos [10,19] have to perform a set of calibration experiments to eliminate diversity. Tagoram [8] needs to know the antenna's position in advance and is also compute-intensive.

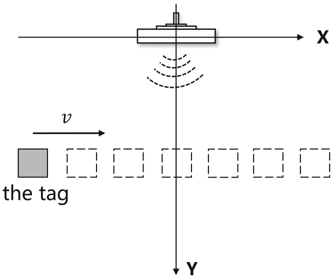


Fig. 2. The tag passes the y-axis.

Unlike localization, line crossing detection concerns whether and when people or objects go across a given barrier line, instead of the tag's coordinates at any time. RF-Line breaks down this problem into two parts. First, when the tag moves along the line parallel to the plane of the antenna (e.g., tagged luggage on conveyor belt), we keep collecting RF phases of the tag and form a phase profile with timestamps. By removing the periodicity of the phase profile and using curve-fitting of hyperbola, we are able to get an inflection point that indicates the time when line crossing happens. Second, in a more generalized case where the tag's trajectory is uncertain, we deploy two antennas opposite to each other and collect the RF phase profiles of two antennas at the same time. By a series of geometric transformations and mathematical derivations, we find that summing up the two phase profiles will get a new phase curve, in which the inflection point is the time of line crossing. We implement RF-Line with commercial off-the-shelf (COTS) RFID reader (Impinj R420 [20]) equipped with two antennas (Laird S9028PCR [21]). Extensive experiments show that RF-Line can achieve accurate line crossing detection with a small error of 6.1 cm, with no need for any system calibration nor complicated deployment.

2 Problem Definition

An RFID system typically consists of some tags and one or more readers (antennas). Each tag is attached to an object to exclusively indicate the associated object. By communicating with a tag, the antenna can obtain the attributes of the tagged object or the information of physical-layer signals emitted by the tag. As shown in Fig. 2, let the center of an antenna be the origin O. If one antenna is used, the y-axis is on the line perpendicular to plane of the antenna. If two antennas are deployed, the y-axis is the line that goes through the two centers of the antennas. The x-axis is perpendicular to the y-axis. In RF-Line, the problem of line crossing detection is to check whether and when people or objects go across the y-axis. This is quite common and important in our daily life, such as the EAS checkpoint in a retail store or the finish line in track and field.

The difficulty is that the interrogation zone of an RFID antenna is not a line. Instead, its shape is irregular and unpredictable since the antenna's signals are susceptible to various factors, e.g., the antenna planning, multi-path effects, and surroundings. We cannot roughly take the time when a tag is seen by the antenna for the first time as the time when line crossing occurs. For ease of presentation, we just use a one-tag case to show how RF-Line works in what follows. If multiple tagged targets go across the line at the same time, we can classify the collected data based on the tag ID and deal with each tag's data individually. Hence, the multi-tag case can be easily reduced to one-tag case.

As the tagged target moves, the distance between the antenna and the tag keeps changing, which leads to the variance of RF phase that is our vehicle for line crossing detection. The RF phase reflects the offset degree between the received signal and the sent signal of electromagnetic wave, ranging from 0 to 2π (360°), which is a common parameter supported by COTS readers, e.g., Impinj R420 [20]. Suppose the distance between the reader antenna and the tag is d. According to the round-trip backscatter communication, the signal travels a total distance of $2d$. In addition, the tag's reflection coefficient, the reader's transmission circuit, and the reader's receiver circuits will also cause extra phase rotations, which are denoted as θ_{TAG}, θ_{TX} and θ_{RX} respectively. The phase output θ can be expressed as follows:

$$\begin{cases} \theta = (2\pi \times \frac{2d}{\lambda} + \mu) \ mod \ 2\pi \\ \mu = \theta_{TX} + \theta_{RX} + \theta_{TAG}, \end{cases} \tag{1}$$

where λ is the wavelength. The term μ is called diversity term, which is determined by hardware characteristics. As shown in Fig. 2, when a tagged object moves, the antenna keeps querying tags and collecting a sequence of RF phase values together with the corresponding timestamps, which is a phase profile of the tag, denoted by $\{(\hat{\theta_1}, t_1), ..., (\hat{\theta_n}, t_n)\}$. The objective of RF-Line is to use this profile to estimate whether and when the tag crosses the y-axis.

3 RF-Line

3.1 Basic Idea

We first consider a simple case that the tag moves along the x-axis at a constant speed, which is demonstrated in Fig. 2. This can be used in some applications such as conveyor belt in the airport for baggage check or delivery of cargo from storage. In this case, we use one antenna and keep collecting signals from the tag and label each of the corresponding phase value with the timestamps. The labeled RF phase is denoted by $\{(\hat{\theta_1}, t_1), ..., (\hat{\theta_n}, t_n)\}$, which forms a phase profile. In this profile, the x-coordinate is the timestamp and the y-coordinate is the phase value. We find that the phase value looks symmetrical due to the tag's movement. As the tag moves, the displacement between the antenna and the tag first decreases and then increases after reaching a minimum when crossing the barrier line, resulting in a symmetrical phase pattern in the phase profile. We draw a typical pattern in Fig. 3. As we can see, the phase value repeatedly reduces from 2π to 0 until the tag reaches the nearest place to the antenna. After

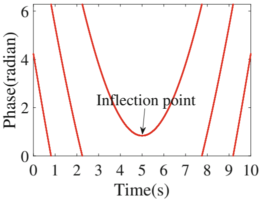


Fig. 3. Theoretical phase profile of the tag.

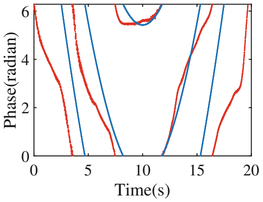


Fig. 4. Without DTW.

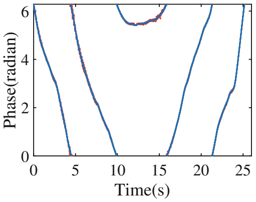


Fig. 5. With DTW.

that, the phase value starts to increase from 0 to 2π periodically and results in an inflection point in this pattern. The inflection point happens when the tag passes the antenna. Through its timestamp, we can easily know when the tag crosses the line (i.e., the y-axis of the coordinate). To find the zero-crossing of the derivative, we use a threshold test. The phase value whose first derivative is equal to zero will be chosen as the inflection point and its y-coordinate indicates when the tag crosses the line.

3.2 Line Crossing Detection with Time Warping

In this subsection, we consider a more generalized case that the object moves at a non-uniform speed. In this case, the curve of RF phase will be compressed or stretched as the speed of the tag changes. To solve this problem, we use the Dynamic Time Warping(DTW) algorithm. DTW is one of the algorithms for measuring similarity between two temporal sequences, which might be with different lengths. Assume two phase profiles are X and Y with the lengths of M and N respectively. DTW defines a warping path w in the form of $w = w_1, w_2, ..., w_K$, where $Max(M, N) \leq K \leq M + N$. The form of w_k is (i, j), where i represents the i^{th} coordinate in M and j represents the j^{th} coordinate in N. The warping path W must begin with $w_1 = (1, 1)$ and end with $w_K = (M, N)$ for ensuring that every coordinate in M and N appears in W. In addition, i and j of $w(i, j)$ in W must be monotonically increased, which means:

$$w_k = (i, j), w_{k+1} = (i', j'), i \leq i' \leq i + 1, j \leq j' \leq j + 1. \tag{2}$$

The result warping path is the one with the shortest distance D as follows:

$$D(i, j) = Dist(i, j) + min\{D(i-1, j), D(i, j-1), D(i-1, j-1)\}, \tag{3}$$

where $Dist(i, j) = ||x_i - y_j||$, x_i and y_j are the i^{th} and j^{th} elements of phase X and phase Y. The aim of the algorithm is to find out the final warping path w that minimizes $D(M, N)$ by using dynamic programming.

Figure 4 and Fig. 5 show the results of DTW algorithm. The curve of the measured phase profile and the theoretical phase profile matches well. DTW

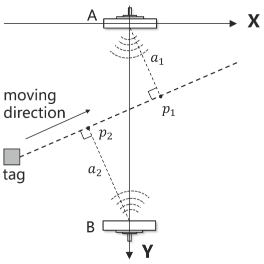


Fig. 6. Two antennas.

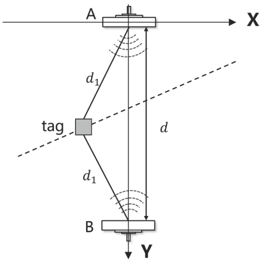


Fig. 7. The sum of phases of two antennas.

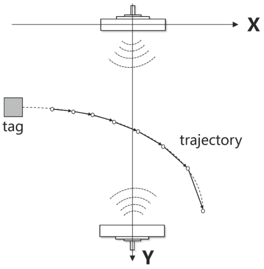


Fig. 8. Non-linear trajectory.

algorithm uses global matching to get the best matching point. The point in the measured phase curve which corresponds to the inflection point of the theoretical curve with DTW is considered as the time when the tag passes y-axis.

3.3 Generalized Cases

In this subsection, we discuss three more generalized situations. First, we assume that the tag moves along a straight line which is not parallel to x-axis, at a constant speed. The point closest to the antenna is not on the line perpendicular to plane of the antenna, which means the x-coordinate of the inflection point of the phase profile does not correspond to the time when the tag passes the y-axis. To handle this problem, we deploy one more antenna and try to find the inflection point by jointly considering both of their phase values. As shown in Fig. 6, two antennas are on the y-axis, opposite to each other. The inflection points happen when the tag is located at p_1 and p_2. Let t_1, t_2, and t be the time when the tag passes p_1, p_2, and the y-axis, respectively. We have:

$$\frac{t_1 - t}{a_1} = \frac{t - t_2}{a_2}, \tag{4}$$

where a_1 and a_2 means the distance from p_1 to A and that of p_2 to B. According to Sect. 3.1, t_1 and t_2 can be calculated. If the value of a_1 and a_2 can be obtained, then we can get the time t as desired. To do so, we can deal with a_1 and a_2 individually. Hence, the problem is reduced to one-antenna case. As shown in Fig. 2, the antenna is at the origin and the coordinate of the tag is (a, b). When the tag moves at a constant speed of v along the x-axis, the location of the tag is $(a - vt, b)$ at the time t. Hence, the distance d at the time t is:

$$d = \sqrt{b^2 + (a - vt)^2}. \tag{5}$$

Substituting Eq. (5) in Eq. (1) and remove the periodical pattern of the phase profile, we get the phase value:

$$\theta = \frac{4\pi}{\lambda}\sqrt{b^2 + (a - vt)^2} + \mu. \tag{6}$$

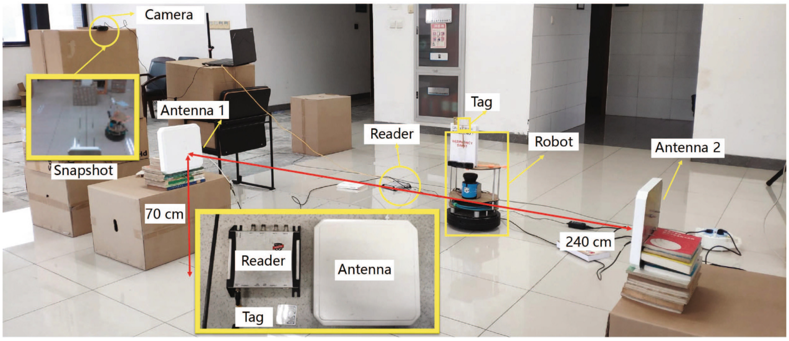


Fig. 9. System deployment.

Deforming the formula, we have:

$$\frac{(\theta-\mu)^2}{(\frac{4\pi}{\lambda})^2 b^2} - \frac{(t-\frac{a}{v})^2}{\frac{b^2}{v^2}} = 1, \tag{7}$$

where a, b and v are considered as constants, θ and t are variables. This formula shows that the curve of the phase profile is actually half of a hyperbola. Hence, we can get the estimates of b by using one of the existing curve-fitting algorithms of hyperbola. Then we can use this method to calculate a_1 and a_2 in Eq. (4), and finally calculate the desired time t.

However, when the tag does not move at a constant speed, the above method does not work. As a result, we propose a new method by jointly considering the phase measurements from the two antennas. As shown in Fig. 7, when the object moves, we assume the distances from the tag to the two antennas are d_1 and d_2 and the distance between two antennas is d. Obviously, $d_1 + d_2 \geq d$ because the sum of the two sides of a triangle is greater than the third. The value of $d_1 + d_2$ reaches the minimum when the object passes the y-axis. Assume that the phase values of the object read by the two antennas are θ_1 and θ_2 respectively, then we can get the formula according to Eq. (1):

$$\theta_1 + \theta_2 = (\frac{4\pi(d_1+d_2)}{\lambda} + \mu)\ mod\ 2\pi. \tag{8}$$

As aforementioned, the sum of the two phases also first repeatedly reduces from 2π to 0, then suddenly changes to 2π. When the object passes the y-axis, the sum reaches a local minimum and then starts to increase, from 0 to 2π. So the figure of $\theta_1 + \theta_2$ also has an inflection point with x-coordinate corresponding to the time when the object passes the y-axis, which is similar to the method we introduced in Sect. 3.1. We also determine this by calculating the first-order derivative and the point whose first-order derivative is equal to 0 is just the inflection point and its x-coordinate is just the line crossing time.

Note that the method we introduced above also works when the tag does not move linearly as shown in Fig. 8. In this case, the minimum of the sum of the two phases also holds if and only if the tag crosses the y-axis. Therefore, the same method can be used to infer the time when the tag's trajectory is a curve.

4 Evaluation

4.1 System Setup

As shown in Fig. 9, we build a prototype of RF-Line. The system mainly consists of two parts: an RFID module and a robot module. The RFID module consists of an RFID reader, two antennas, and several RFID tags. The reader model is Impinj Speedway R420 [20] with working frequency ranging from 920 to 924 MHz. The antenna model is Laird S9028PCL [21] with 8 dBm signal gain. The two RFID antennas are deployed in opposite directions and both of them are connected to the same reader through a cable line. The tag model is Impinj H47 [20]. The valid data acquired by the reader will be forwarded to a backend server through WIFI. We use the Robot Operating System (ROS) to synchronize the clock of the reader and the backend server. The robot module consists of a robot and several RFID tags. We attach tags to the robot, which is used to perform line crossing. The robot model is TurtleBot2 [22] with programable moving trajectory. The ground truth is recorded through a camera.

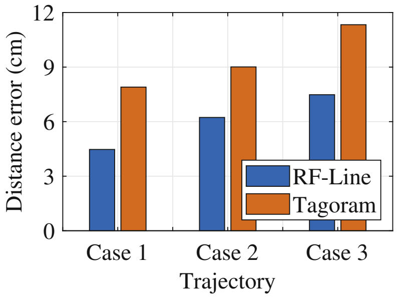


Fig. 10. Accuracy of different trajectories.

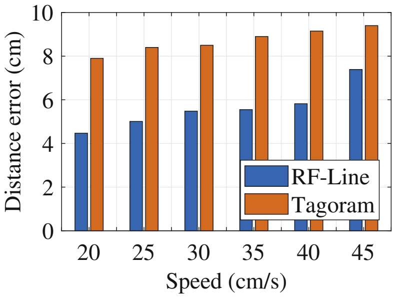


Fig. 11. Impact of moving speed.

4.2 Accuracy

In this subsection, we set up the real scenarios and compare the accuracy of RF-Line with a state-of-the-art localization method called Tagoram [8]. We use the exact passing line time obtained by the camera as ground truth. Tagoram is a localization work that can also be used to detect passing line time obviously. We study the impact of different parameter settings on distance error and compare the performance of our method with Tagoram under different scenarios.

We keep collecting phase values from the tag while the tag on the robot is moving in three different trajectories: straight line, slanted line and curved line, respectively denoted as case 1, case 2 and case 3. Then we observe the distance errors of RF-Line, Baseline solution and Tagoram. As shown in Fig. 10, the distance errors of Tagoram and RF-Line are 7.90 cm and 4.47 cm respectively in the straight line scenario. The distance errors of Tagoram and RF-Line are 9.01 cm and 6.23 cm respectively in the slanted line scenario. The distance errors of Tagoram and RF-Line are 11.33 cm and 7.48 cm respectively in the curved line scenario. Clearly, we can see that RF-Line is superior to Tagoram under the three cases. Figure Next, we discuss the impact of different factors on the accuracy, i.e., the moving speed, the number of tags, and the angle of antenna rotation.

Impact of Moving Speed. In the experiment, we vary the speed v of the robot from 20 cm/s to 45 cm/s, which covers most of the situations in practice. As shown in Fig. 11, the results show that RF-Line outperforms Tagoram at all testing speeds. The distance errors are 4.47 cm, 5.01 cm, 5.48 cm, 5.55 cm, 5.82 cm, 7.39 cm when the speed is 20 cm/s, 25 cm/s, 30 cm/s, 35 cm/s, 40 cm/s, 45 cm/s. Besides, it is worth noting that the detection error increases as the speed of robot increases. This is because a higher speed will reduce the sampled data from the tags, which has same impact on the estimate accuracy.

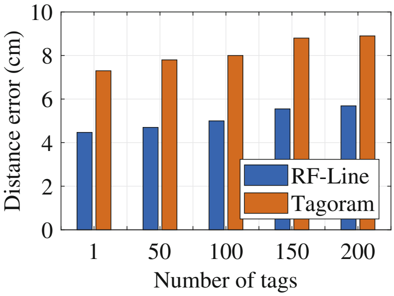


Fig. 12. Impact of number of tags.

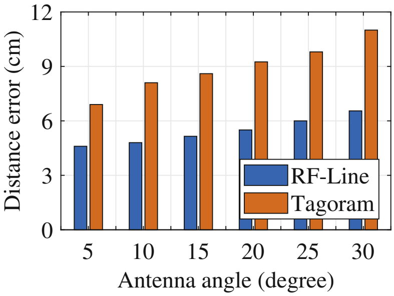


Fig. 13. Impact of antenna placement.

Impact of Number of Tags. In a multi-tag case, the system needs to detect multiple line crossing events at the same time. More tags will decrease the sampling rate of the reader for each tag, which shall have a negative effect on the detection accuracy. To study this impact, we vary the number of tags used in the system and test the corresponding accuracy. In all experiments, the robot moves along a straight line at a constant speed of 20 cm/s. As shown in Fig. 12, the distance error is 5.69 cm when the number of tags is 200, which is 1.24 cm bigger than non-interference condition. Clearly, the error is not large as well. The results illustrate that the number of tags has little impact on our method.

Impact of Antenna Placement. Next, we collect the data when the two antennas use different rotation angles. We rotate the angle of two antennas from 5° to 30°, and observe the detection accuracy. The results are shown in Fig. 13. As we can see, the distance error increases as the rotation angle increases. This is because the rotation of the antenna will result in some errors in its phase measurements. A larger angle causes a lower detection accuracy.

5 Related Work

Existing solutions to line crossing detection generally fall into three categories: infrared sensor based systems [4,5], camera based systems [6,7,23] and virtual wall [2,3,24]. Infrared sensor detects line crossing by measuring infrared light radiating from objects. It can detect a line crossing when the object is closed to sensors, which is widely used in an access control system. Camera based system recognizes object moving through image processing. By observing the pixels of the object, the line crossing event can be detected as well. In robotics, a robot constructs the virtual wall, which is an invisible barrier that the robot can not cross. When the robot is close to the virtual wall, its navigation system will give immediately change its moving trajectory. These line crossing systems have already been widely used but they still do not run well in many practical cases, i.e., multiple objects or people cross the line at the same time.

RFID, as a non-contact passive sensing technology, can solve this problem. There is no direct solution to line crossing detection, but similar RFID localization work. RFID localization work can be divided into two types, RSSI based methods [10,25] and RF phase based methods [8,18]. RSSI based methods usually appear in some early works. These methods commonly need to deploy a large number of reference tags whose positions are known. Due to the low-resolution of RSSI and the restrictions of deployment, it is not a very good choice for localization. In recent years, the research starts to shift to using RF phase for RFID localization. The competitive advantage of RF phase is that it provides us with a high-resolution measurement of the signals when the tag-reader communication distance varies. Many advanced phase based methods have been proposed for dealing with different scenarios, especially for mobile localization. The most related work is localization of RFID tags moving on a conveyor belt. The work [25] uses the reader antenna's radiation pattern together with the RSSI threshold to determine the order of tagged tires on the conveyor belt. It however cannot figure out when the tagged tires go across a given line and also suffers from environmental affects. The work [26] jointly uses synthetic-array radar principles, knowledge-based processing, and the reader-tag communication signal to track tagged items moving along a conveyor belt. The above solutions however are tailored to the applications of conveyor belt, which requires to know the trajectory and speed of the object.

6 Conclusion

Line crossing detection is quite common in our daily life. In this paper we propose a lightweight method called RF-Line that deploys two antennas opposite to each other to track the objects crossing a given barrier line, without any complicated system deployment or calibration. By concurrently collecting the RF phase and jointly estimating an inflection point by two antennas, RF-Line is able to know whether and when a tagged object crosses the line connecting the center of the two antennas. We implement a prototype of RF-Line with a commodity RFID system. Extensive experiments show that RF-Line can achieve a high tracking accuracy with a small error of 6.1 cm.

Acknowledgment. This research is financially supported by the National Natural Science Foundation of China under Grant 62072231, the Open Project of State Key Laboratory for Novel Software Technology under Grant KFKT2021B15, and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

1. Thomas, E.P.: Product tags, systems, and methods for crowdsourcing and electronic article surveillance in retail inventory management, March 12 2019. US Patent 10,229,386
2. Abbott, J.J., Okamura, A.M.: Effects of position quantization and sampling rate on virtual-wall passivity. IEEE Trans. Rob. **21**(5), 952–964 (2005)
3. Ordonez, C., Collins, E.G., Jr., Selekwa, M.F., Dunlap, D.D.: The virtual wall approach to limit cycle avoidance for unmanned ground vehicles. Robot. Auton. Syst. **56**(8), 645–657 (2008)
4. Norman, T.L.: Integrated security systems design: concepts, specifications, and implementation. Elsevier (2011)
5. Tan, S.: Intelligent entrance guard lock system based on remote control, 2016. US Patent CN106355724A (2016)
6. Curry, G.: Camera-based tracking and position determination for sporting events using event information and intelligence data extracted in real-time from position information, November 10 2015. US Patent 9,185,361
7. Newcombe, R., et al.: Real-time camera tracking using depth maps, March 19 2013. US Patent 8,401,242 (2013)
8. Yang, L., Chen, Y., Li, X.-Y., Xiao, C., Li, M., Liu, Y.: Tagoram: real-time Tracking of mobile RFID tags to high precision using COTS devices. In: Proceedings of ACM MobiCom, pp. 237–248 (2014)
9. Wang, C., Xie, L., Wang, W., Chen, Y., Xue, T., Sanglu, L.: Probing into the physical layer: moving tag detection for large-scale RFID systems. IEEE Trans. Mob. Comput. **19**(5), 1200–1215 (2020)
10. Wang, J., Katabi, D.: Dude, where's my card? RFID positioning that works with multipath and non-Line of sight. In: Proceedings of ACM SIGCOMM, pp. 51–62 (2013)
11. Shangguan, L., Yang, Z., Liu, A.X., Zhou, Z., Liu, Y.: Relative localization of RFID tags using spatial-temporal phase profiling. In: Proceedings of USENIX NSDI, pp. 251–263 (2015)

12. Wang, G., et al.: HMRL: relative localization of RFID tags with static devices. In: Proceedings of IEEE SECON, pp. 1–9 (2017)
13. Liu, J., Chen, M., Chen, S., Pan, Q., Che, L.: Tag-compass: determining the spatial direction of an object with small dimensions. In: Proceedings of IEEE INFOCOM, pp. 1–9 (2017)
14. Chen, X., Liu, J., Wang, X., Liu, H., Jiang, D., Chen, L.: Eingerprint: robust energy-related fingerprinting for passive RFID tags. In: Proceedings of USENIX NSDI, pp. 1101–1113 (2020)
15. Liu, J., Chen, X., Chen, S., Wang, W., Jiang, D., Chen, L.: Retwork: exploring reader network with COTS RFID systems. In: Proceedings of USENIX ATC, pp. 889–896 (2020)
16. Qian, C., Liu, Y., Ngan, R.H., Ni, L.M.: ASAP: scalable collision arbitration for large RFID systems. IEEE Trans. Parallel Distrib. Syst. **24**(7), 1277–1288 (2013)
17. Qi, S., Zheng, Y., Li, M., Liu, Y., Qiu, J.: Scalable industry data access control in RFID-enabled supply chain. IEEE/ACM Trans. Networking **24**(6), 3551–3564 (2016)
18. Liu, J., Zhu, F., Wang, Y., Wang, X., Pan, Q., Chen, L.: Rf-scanner: shelf scanning with robot-assisted RFID systems. In: Proceedings of IEEE INFOCOM, pp. 1–9 (2017)
19. Liu, T., Liu, Y., Yang, L., Guo, Y., Wang, C.: Backpos: high accuracy backscatter positioning system. IEEE Trans. Mob. Comput. **15**(3), 586–598 (2016)
20. Inc Impinj. http://www.impinj.com/
21. Inc Laird. https://www.lairdconnect.com/rf-antennas/rfid-antennas/s902-series-rfid-antenna
22. Inc TurtleBot. https://www.turtlebot.com/turtlebot2/
23. Pupilli, M., Calway, A.: Real-time camera tracking using a particle filter. In: BMVC (2005)
24. Chiu, T.-Y.: Virtual wall system, November 16 2006. US Patent App. 11/176,244 (2006)
25. Caccami, M.C., Amendola, S., Occhiuzzi, C.: Method and system for reading RFID tags embedded into tires on conveyors. In: Proceedings of IEEE RFID-TA, pp. 141–144 (2019)
26. Buffi, A., Nepa, P., Lombardini, F.: A phase-based technique for localization of UHF-RFID tags moving on a conveyor belt: performance analysis and test-case measurements. IEEE Sens. J. **15**(1), 387–396 (2015)

Joint Beamforming and Deployment Optimization for UAV-Assisted Maritime Monitoring Networks

Lin Liu[1], Bin Lin[1,2](✉), Ran Zhang[3], Yudi Che[1], and Chaoyue Zhang[1]

[1] School of Information Science and Technology, Dalian Maritime University,
Dalian 116026, China
[2] Peng Cheng Laboratory, Shenzhen 518052, China
binlin@dlmu.edu.cn
[3] Department of Electrical and Computer Engineering, Miami University,
Oxford, OH 45056, USA

Abstract. With the wide application of Internet of Things (IoT) systems in the smart ocean, many unmanned surface vehicles (USVs) have been deployed jointly with unmanned aerial vehicles (UAVs) to monitor the maritime environment. However, conventional means of maritime communications fail to provide high-rate services due to the complex maritime channel conditions and large transmission distance, which will affect the environmental monitoring performance. In this paper, we propose a USV-UAV collaborative patrol scheme for maritime environment monitoring networks. Considering the characteristic of energy concentration in beamforming, we investigate the joint beamforming and location deployment optimization problem (BLDO) for UAV relay. Specifically, we decompose the BLDO problem into two subproblems. In the first sub-problem, the location deployment of UAV and beam gain allocation is optimized via an iterative algorithm based on the approximated beam patterns. The algorithm can effectively reduce the computational complexity of the grid-search method. In the second sub-problem, beamforming optimization is conducted with a high-dimensional constant-modulus (CM) constraint. A micro-particle swarm optimization-based algorithm with boundary relaxation ($BR{-}\mu PSO$) is proposed to obtain an optimal solution. Finally, the simulation results demonstrate that the proposed algorithms can improve the performance in terms of the achievable sum rate and the beam gain.

Keywords: UAV · USV · Maritime environment monitoring · Deployment · Beamforming

1 Introduction

With the rapid development of the maritime economy, oily wastewater, toxicant-containing wastewater, and domestic solid wastes, etc., pose a serious threat to ecological environment protection, which is becoming an urgent issue [1].

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 40–51, 2022.
https://doi.org/10.1007/978-3-031-19214-2_4

Yan *et al.* [2] deployed a wireless sensor network (WSN) to locate the source of pollution in the urban water supply network. However, WSN is inflexible and has limited monitoring range and unsatisfactory adaptability to the complex maritime environment. The existing maritime communication systems typically rely on satellite communications and very-high-frequency (VHF) communications [3]. However, the high cost of satellite communication and the limited bandwidth of VHF cannot support the access of multiple acquisition terminals. Therefore, it is imperative to design efficient data uploading schemes to improve the communication capacity for maritime environmental monitoring networks.

To increase the communication capacity, multiple antenna technique has been introduced for maritime communication systems in [4]. Particularly, beamforming (BF) in multiple-input multiple-output (MIMO) system has been considered as one of the major candidate technologies [6–9]. Beamforming technology provides the benefits of increased diversity for the BS and user equipment. Smart antennas enable increase of capacity in wireless communication systems by successfully reducing channel interference. Zhu *et al.* in [6] employed the analog beamforming to achieve the directional beamforming, which can effectively suppress the interference from other users. Su *et al.* in [7] demonstrated that beamforming technique can offer considerable beam gain to overcome the high propagation loss. To further improve the transmission rate, Zhu *et al.* in [6] and Xiao *et al.* in [8] explored the joint power allocation and beamforming for a two-user downlink and uplink mm-Wave NOMA scenario, respectively. At present, most of the studies are based on terrestrial communication systems.

Unmanned Aerial Vehicles (UAVs) have been widely employed in emergency and environmental monitoring tasks in the past few years. However, for the existing methods on UAV deployment monitoring [9,10,12], beamforming has not been taken into consideration yet. They may suffer from the interference from the maritime climate and neighboring infrastructures [10]. Considering the flexibility of UAVs and the advantages of beamforming technology such as anti-interference and energy concentration, the combination of the two is very promising [11,12]. It can not only improve the communication quality of UAV, but also save communication energy consumption. However, the joint beamforming and UAV location optimization problem will be more complicated since it is highly non-convex and involves high-dimensional, highly coupled variable vectors. For example, Mozaffari *et al.* in [12] presented a grid-search method to calculate the maximum achievable rate of each grid intersection point to determine the approximate optimal location of the UAV. Whereas the algorithm complexity increases exponentially making it difficult to determine the optimal grid accuracy.

The aforementioned beamforming schemes are suitable for terrestrial systems, yet few contributions have been devoted to the problem of maritime beamforming systems. When designing the UAV-assisted USV patrol scheme, the following differences between ocean and land have to be investigated:

- Channel distinction: Maritime propagation environment has unique characteristics such as sparsity, instability and the ducting effect over the sea surface. Therefore, we need to establish a multipath channel model suitable for the characteristics of the maritime environment.

- Energy limitation: The offshore relays are usually powered by solar energy due to the lack of infrastructure. Therefore, we should reduce communication energy consumption as much as possible.

In order to meet the aforementioned challenges, we investigate the beamforming and location deployment optimization (BLDO) problem for UAV-assisted maritime environment monitoring networks. Specifically, the energy is concentrated in the target USV direction through beamforming technology. Since the variables are coupled with each other and have high dimensions, the BLDO problem is decomposed into two sub-problems by introducing the ideal beamforming. In the first stage, an iterative algorithm based on water injection is proposed to find the UAV's optimal position. In the second stage, considering the difficulty of the constant modulus (CM) constraint and the "curse of dimensionality" of the high-dimensional problems, a micro-particle swarm algorithm ($BR - \mu PSO$) is proposed based on boundary relaxation to obtain the beamforming vector. Our main contributions are summarized as follows:

- Beamforming technology is combined with UAV assisted communication to maximize the achievable sum rate of data uploading from the patrol USVs. The beam gain of the target USV direction is significantly enhanced, thus solving the problem of limited maritime communication bandwidth without increasing hardware cost.
- An iterative algorithm and a particle swarm optimization algorithm based on boundary relaxation ($BR - \mu PSO$) are proposed to solve the UAV location deployment and beamforming optimization problems, respectively. The results show that, the energy is concentrated in the direction of target USVs, and the proposed algorithms can efficiently improve the achievable sum rate and the beam gain.

The rest of this paper is organized as follows. The system and channel model of the maritime MIMO system is introduced in Sect. 1. Sections 2 and 3 describe the deployment and analog beamforming optimization of the hovering UAV, respectively. The simulation results are presented and discussed in Sect. 4. Section 5 concludes this paper.

Notation: In this paper, $\mathbf{I}_n$ stands for an $n \times n$ identity matrix, $()^H$, $||$, $\|\|$ denote Hermitian transpose, the absolute value of a complex number, the Euclidean norm respectively.

2 System Model and Problem Formulation

2.1 System Model

We consider a UAV-assisted USV patrol scheme for maritime monitoring network as depicted in Fig. 1, where one UAV is responsible for air patrol and the USVs are responsible for information collection. The network is expected to realize high reliability and low delay in information transmission while increasing communication capacity. The UAV is equipped with an M-element uniform linear array (ULA), serving K USVs with a single antenna. To enable multistream communications, each antenna branch has a phase shifter and a power amplifier (PA) to drive the antenna.

For the sake of convenience, we establish a 3-D rectangular coordinate system to represent UAV and USVs' location relationship, where USVs are distributed on the horizontal plane located at $(x_k, y_k, 0)$ and the coordinate of the UAV is (x_u, y_u, h_u). Note that we use orthogonal frequency division multiplexing (OFDM) technology, where each USV occupies an independent frequency to transmit the information $s_k \sim \mathcal{CN}(0,1)$ to the UAV relay. The kth USV transmits signal s_i to the UAV with the corresponding transmit power p_k, where $\mathbb{E}(|s_i|^2) = 1$. Then the received signal $\mathbf{y}_{UAV} \in \mathbb{C}^{M\times 1}$ at the UAV can be expressed as

$$\mathbf{y}_{UAV} = \sum_{k=1}^{K} \mathbf{H}_k^H \mathbf{w} \sqrt{p_k} s_k + \mathbf{n_1} \tag{1}$$

where $\mathbf{H}_k$ is channel response vector between the k**th** USV and the UAV, the elements of vector $\mathbf{n_1}$ represent additive white Gaussian noise (AWGN) with variance σ_1^2, and $\mathbf{w}$ denotes an $M \times 1$ beamforming (BF) vector with CM constraint for ULA structure, i.e., $|[\mathbf{w}_k]| = \frac{1}{\sqrt{M}}$ for $k = 1, 2, \ldots, M$.

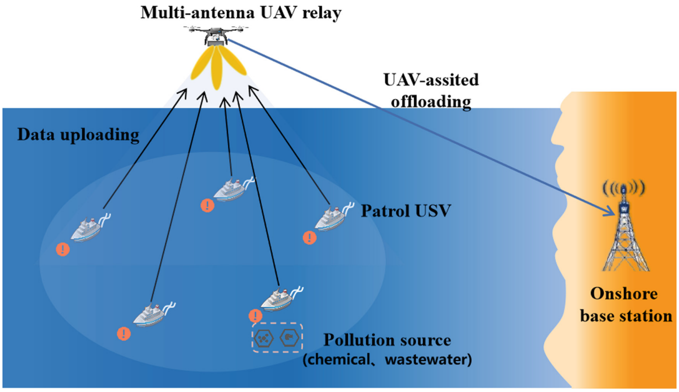


Fig. 1. Illustration of a maritime patrol scenario including one UAV, and multiple USVs.

Due to the lack of scatters in the vast sea area, the line-of-sight (LoS) path will dominate most of the air-to-sea channels. The Rayleigh fading, generally analyzed in the terrestrial communication systems, may no longer be suitable for the maritime environment. Instead, the finite scattering channel [13] could be more appropriate for the maritime model. Furthermore, the reflection path from the sea surface may exist in some conditions, resulting in severe multipath effects. Therefore, a sparse multipath channel based on multipath fading is conceived to describe the

USV-UAV channel in our model. The uplink channel (UL) between USV and UAV is denoted by $\mathbf{h}_k$. Different multipath components (MPCs) have different physical receive steering angles, i.e., angles of arrival (AoAs). With half-spaced ULAs adopted at the receiver, the channel matrix can be expressed as

$$\mathbf{a}_r\left(\phi_l\right) = \sqrt{\frac{1}{M}}\left[1, e^{j\pi\phi}, \cdots, e^{j\pi(M-1)\phi}\right]^T \tag{2}$$

$$\phi_l = \frac{x_u - x_i}{\sqrt{\left(x_u - x_k\right)^2 + \left(y_u - y_k\right)^2 + h_u^2}} \tag{3}$$

$\mathbf{a}_r\left(\phi_l\right)$ is the antenna array response vector at the UAV, where ϕ_l denotes the real AoA of the lth MPC for the kth USV i.e. $\phi_l = cos\left(AoA\right)$, and ϕ_l is within the range of $(-1, 1)$. We only consider the azimuth and neglect elevation to implement horizontal 2-D beamforming. The extension to 3-D beamforming by adopting an uniform planar array (UPA) configuration may also be possible.

2.2 Problem Formulation

In this subsection, we aim to maximize the achievable sum uploading rate of all USVs when the channel is known prior. For each USV, under the constraints of minimal rate for USVs and antenna structure, the achievable rate R_k is denoted by

$$R_k = \log_2\left(1 + \frac{p_k\left|\mathbf{h}_k^H\mathbf{w}\right|^2}{\sigma^2}\right) \tag{4}$$

where p_k is the transmission power at each USV, and σ^2 is the power of Gaussian white noise at ith USV. $\left|\mathbf{h}_k^H\mathbf{w}\right|^2$ denotes the effective channel gain between the kth USV and UAV. In this problem, the UAV deployment intertwines with the beamforming design, accordingly, the achievable sum rate maximization problem can be formulated as

$$\begin{aligned} P_0 : \max_{\mathbf{w}, x_u, y_u} & \sum_{k=1}^{K} \log_2\left(1 + \frac{p_k\left|\mathbf{h}_k^H\mathbf{w}\right|^2}{\sigma^2}\right) \\ \text{s.t. } & \text{C1 :} R_k \geq \zeta_k \qquad k = 1, 2 \ldots, K \\ & \text{C2 :} \left|[\mathbf{w}]_i\right| = \frac{1}{\sqrt{M}} \qquad i = 1, 2 \ldots, N \\ & \text{C3 :} \left(x_u, y_u\right) \in \mathbb{D} \end{aligned} \tag{5}$$

where ζ_k denotes the minimum rate requirement for kth USV, and thus, C1 denotes the QoS requirement for each USV. Meanwhile, $|[\mathbf{w}]| = \frac{1}{\sqrt{M}}$ is the CM constraint due to using the phase shifters in each antenna branch at the UAV. The optimization variables are the projected coordinates of UAV (x_u, y_u) and the beamforming vector $\mathbf{w}$.

3 Problem Solution

Directly solving the BLDO problem (5) by using the existing optimization tools is infeasible, because the problem is non-convex, and the UAV position variables intertwine with the beamforming vector. Since the location of the UAV crucially affects the channel matrix, we can resort to the approximate beam pattern and decompose the BLDO problem into two sub-problems that are relatively easy to solve one by one.

3.1 UAV Deployment and Beam Gain Allocation Sub-problem

We first resort to approximate beam patterns and try to decompose the deployment and beamforming variables. Then, we have the following lemma.

Lemma 1: With the ideal beamforming, the beam gains satisfy

$$\frac{\delta_1}{\left|\bar{\lambda}_1\right|^2}+\frac{\delta_2}{\left|\bar{\lambda}_2\right|^2}+\cdots+\frac{\delta_k}{\left|\bar{\lambda}_k\right|^2}=M \tag{6}$$

Note that in the case of ideal beamforming, the beam gains along the USV directions are fixed with a beam width of $\frac{K}{M}$, while those along nonuser directions are all zeros, i.e., there are no side lobes. Then, we have $\sum_{k=1}^{K}\frac{\left|\mathbf{h}_k^H\mathbf{w}\right|^2}{\left|\bar{\lambda}_k\right|^2}=M$, where $\delta_k{=}\left|\mathbf{a}_k^H\mathbf{w}\right|^2$ denotes the antenna beam gain of the kth USV, and $\left|\bar{\lambda}_k\right|=\max\left|\lambda_{m,l}^k\right|$, denotes the index of the strongest MPC for USVs. For the kth USV, the UAV maximizes the effective channel gain by fixed beam direction. It can be approximated as

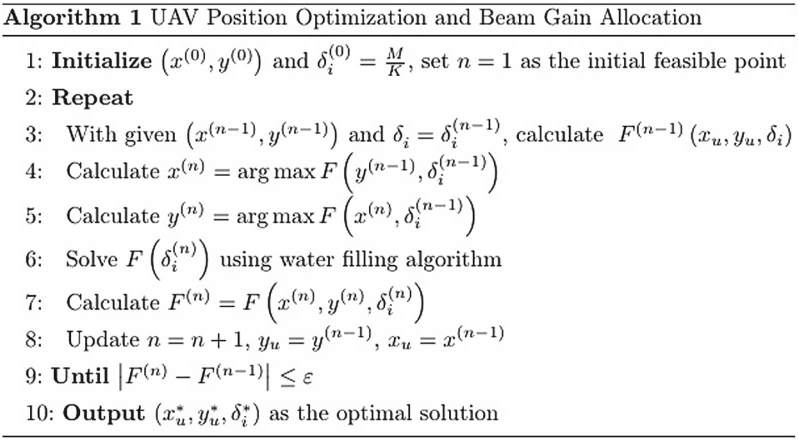
$$\left|\mathbf{h}_k^H\mathbf{w}\right|^2\approx\left|\bar{\lambda}_k\right|^2\left|\mathbf{a}_k^H\mathbf{w}\right|^2 \tag{7}$$

Therefore, based on Lemma 1, we can rewrite the original achievable sum rate maximization problem with the beamforming gains, and simplify it to the problems of UAV deployment and beam gain assignment.

$$\begin{aligned}
P_1: &\max_{(x_u,y_u),\delta_k}\sum_{k=1}^{K}\log_2\left(1+\frac{\sum\limits_{m=1}^{M}p_m^k\left|\bar{\lambda}_k\right|^2\delta_k}{\sigma^2}\right)\\
\text{s.t. } &\text{C1}:\log_2\left(1+\frac{\sum\limits_{m=1}^{M}p_m^k\left|\bar{\lambda}_k\right|^2\delta_k}{\sigma^2}\right)\geq\zeta_{m,k}\qquad k=1,2\ldots,K\\
&\text{C2}:\sum_{k=1}^{K}\frac{\delta_k}{\left|\bar{\lambda}_k\right|^2}=M\\
&\text{C3}:(x_u,y_u)\in\mathbb{D}\qquad r_i\in\mathcal{R}
\end{aligned} \tag{8}$$

We impose a threshold ζ_m^k on the $SINR_m^k$ for reliable decoding (i.e., $SINR_m^k \geq \zeta_m^k$). C2 is the constraint on ideal beamforming. At the same time, the CM constraint can be ignored in the first sub-problem.

The details of the proposed algorithm are presented in Algorithm 1.



Algorithm 1 UAV Position Optimization and Beam Gain Allocation

```
Initialize $\left(x^{(0)}, y^{(0)}\right)$ and $\delta_i^{(0)} = \frac{M}{K}$, set $n = 1$ as the initial feasible point
Repeat
    With given $\left(x^{(n-1)}, y^{(n-1)}\right)$ and $\delta_i = \delta_i^{(n-1)}$, calculate $F^{(n-1)}\left(x_u, y_u, \delta_i\right)$
    Calculate $x^{(n)} = \arg\max F\left(y^{(n-1)}, \delta_i^{(n-1)}\right)$
    Calculate $y^{(n)} = \arg\max F\left(x^{(n)}, \delta_i^{(n-1)}\right)$
    Solve $F\left(\delta_i^{(n)}\right)$ using water filling algorithm
    Calculate $F^{(n)} = F\left(x^{(n)}, y^{(n)}, \delta_i^{(n)}\right)$
    Update $n = n + 1$, $y_u = y^{(n-1)}$, $x_u = x^{(n-1)}$
Until $\left|F^{(n)} - F^{(n-1)}\right| \leq \varepsilon$
Output $(x_u^*, y_u^*, \delta_i^*)$ as the optimal solution
```

We have hereto solved the first subproblem, and obtain an optimal location of the UAV under the assumption of approximate beamforming.

3.2 Beamforming Optimization Sub-problem

Substituting the obtained optimal location of UAV to the BLDO problem, we obtain the beamforming sub-problem. Since the analog beamforming should support all of the patrol USVs, the principle of beamforming design is to maximize the array gains for all USVs. However, the CM constraint is not accounted for in P_1, and we consider it in the following beamforming sub-problem P_2:

$$\begin{aligned} P_2 :&\max_{\mathbf{w}} \sum_{k=1}^{K} \left|\mathbf{h}_k^H \mathbf{w}\right|^2 \\ \text{s.t. } &\text{C1} : \log_2\left(1 + c_k \cdot \left|\mathbf{a}_k^H \mathbf{w}\right|^2\right) \geq \xi_k \qquad k = 1, 2 \ldots, K \\ &\text{C2} : \sum_{k=1}^{K} \frac{\delta_k}{\left|\bar{\lambda}_k\right|^2} = M \qquad i = 1, 2 \ldots, M \\ &\text{C3} : \left|[\mathbf{w}]_m\right| = \frac{1}{\sqrt{M}} \end{aligned} \tag{9}$$

where $c_k = \left(\frac{P \cdot \left|\bar{\lambda}_k\right|^2}{\delta^2}\right)$ is the channel gain coefficient along the strongest MPC. Problem P_2 is clearly non-convex. In order to ensure that the modulus value of each element in $\mathbf{w}$ is $1/\sqrt{M}$, we transform it into angle domain, and then optimize its phase. It has been confirmed that the phase rotation of the BF does not affect the optimality of this problem. Let $\mathbf{w} = \left(1/\sqrt{M}\right) \cdot e^{j\boldsymbol{\varphi}}$, then we have $\left|\mathbf{h}_k^H \mathbf{w}\right|^2 = |\lambda_k|^2 \cdot \frac{1}{M}\left|\mathbf{a}_k^H e^{j\boldsymbol{\varphi}}\right|^2$.

It has been confirmed that the phase rotation of the BF does not affect the optimality of this problem. Therefore, the elimination norm operation can be performed, and $\mathbf{a}_k^H e^{j\boldsymbol{\varphi}}$ is real and non-negative. We proposed a suboptimal solution, meanwhile, we will provide the optimal solution by relaxing P_2 into the following convex problem:

$$\begin{aligned}
P_3 :&\max_{\boldsymbol{\varphi}} \sum_{k=1}^{K} \mathbf{a}_k^H e^{j\boldsymbol{\varphi}} \\
\text{s.t. C1 :}&\log_2\left(1 + c_k \cdot \left|\mathbf{a}_k^H e^{j\boldsymbol{\varphi}}\right|^2\right) \geq \xi_k \qquad k = 1, 2 \ldots, K \\
\text{C2 :}&\sum_{k=1}^{K} \frac{\delta_k}{\left|\bar{\lambda}_k\right|^2} = M \qquad i = 1, 2 \ldots, M \\
\text{C3 :}&\text{Im}(\mathbf{a}_k^H e^{j\boldsymbol{\varphi}}) = 0
\end{aligned} \tag{10}$$

To solve this problem, some swarm-based algorithms can be considered here, e.g., particle swarm optimization (PSO) algorithm. However, the performance of PSO algorithm begins to decline for high-dimensional problems. In this paper, a micro-particle swarm algorithm with boundary relaxation ($BR - \mu PSO$) is proposed. We transform P_3 into an unconstrained one by means of the penalty function, so we redescribe the constraint of C1 as

$$g_i\left(\boldsymbol{\varphi}\right) = \log_2\left(1 + c_k \cdot \frac{1}{M}\left|\mathbf{a}_k^H e^{j\boldsymbol{\varphi}}\right|^2\right) - \xi_k \geq 0 \tag{11}$$

The objective function can be rewritten as:

$$\begin{aligned}
P_4 :&\text{Minimize}_{\boldsymbol{\varphi}} - \sum_{k=1}^{K} \mathbf{a}_k^H e^{j\boldsymbol{\varphi}} + \mu \sum_{i=1}^{K} \left[\max\left\{0, -g_i\left(\boldsymbol{\varphi}\right)\right\}\right]^2 \\
\text{s.t. C1 :}&\sum_{k=1}^{K} \frac{\delta_k}{\left|\bar{\lambda}_k\right|^2} = M \qquad i = 1, 2 \ldots, M \\
\text{C2 :}&\text{Im}(\mathbf{a}_k^H e^{j\boldsymbol{\varphi}}) = 0
\end{aligned} \tag{12}$$

where the penalty function is expressed as

$$\max\left\{0, -g_i\left(\boldsymbol{\varphi}\right)\right\} = \begin{cases} 0 \\ -g_i\left(\varphi\right) \end{cases} \tag{13}$$

If $\boldsymbol{\varphi}$ is a feasible solution, the value is 0. If not, the value is $-g_i\left(\boldsymbol{\varphi}\right)$. Each particle has a memory for its best found position $\mathbf{P}_{best}$ and the globally best position $\mathbf{G}_{best}$. The rate update formula of $\mathbf{G}_{best}$:

$$[\mathbf{V}]_{g,n}^{t+1} = \omega\, [\mathbf{V}]_{g,n}^{t} - [\mathbf{X}]_{g,n}^{t} + [\mathbf{G}_{best}]_n + [\mathbf{rep}]_{g,n}^{t} \tag{14}$$

For each iteration, the velocity and position of each particle are updated based on:

$$\begin{aligned}[\mathbf{V}]_{j,n}^{t+1} &= \omega\, [\mathbf{V}]_{j,n}^{t} + \text{rand}() * ([\mathbf{P}_{best}]_{j,n} - [\mathbf{X}]_{j,n}^{t}\,) \\ &+ \text{rand}() * ([\mathbf{G}_{best}]_n - [\mathbf{X}]_{j,n}^{t}\,) + [\mathbf{rep}]_{j,n}^{t}\end{aligned} \tag{15}$$

$$[\mathbf{X}]_{j,n}^{t+1} = \begin{cases} [\mathbf{X}]_{j,n}^{t} + [\mathbf{V}]_{g,n}^{t+1}\;, [\mathbf{X}]_{j,n}^{t} = [\mathbf{X}]_{g,n}^{t} \\ [\mathbf{X}]_{j,n}^{t} + [\mathbf{V}]_{j,n}^{t+1}\;, \text{else} \end{cases} \tag{16}$$

The parameter ω is the inertia weight of velocity. $[\mathbf{rep}]_{i,n}^{t}$ is the repulsion experienced from K blacklisted solutions. $\mathbf{d}_{ki} = \mathbf{x}_i - \hat{\mathbf{x}}_k$ is a vector pointing from the blacklisted solution l to the ith particle. The details of the proposed $BR-\mu PSO$ algorithm are presented in Algorithm 2.

Due to the equality constraint, the search space for $\mathbf{X}$ is high-dimensional. We relax the search space to a convex set and adjust the particles on the boundaries of each iteration. The outer and inter boundary is defined as

$$\left\{\mathbf{X} \middle| \left|[\mathbf{X}]_{i,j}\right| = d_{beyond}\right\} \quad d_{beyond} = \frac{1}{\sqrt{M}} \tag{17}$$

$$\left\{\mathbf{X} \middle| \left|[\mathbf{X}]_{i,j}\right| = d_{in}\right\} \quad d_{in} = \frac{t}{T_{\max}} \frac{1}{\sqrt{M}} \tag{18}$$

For each iteration, the particles out of the boundary are adjusted onto the boundary, and eventually converge.

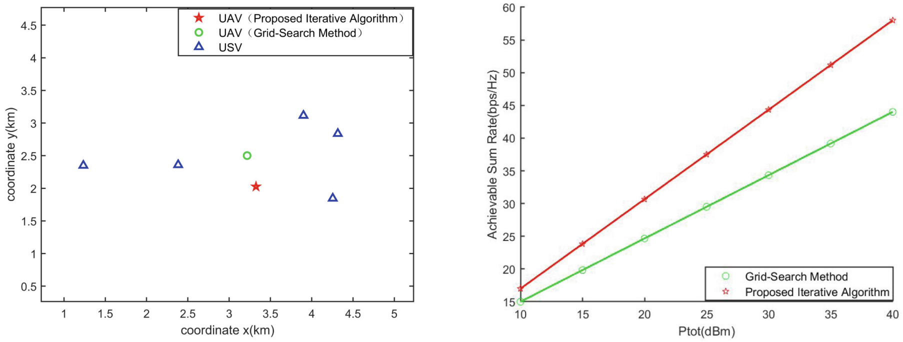
4 Simulation Results

In this section, simulation results are presented to demonstrate the performance of our proposed iterative algorithm for UAV deployment and the $BR-\mu PSO$ algorithm for beamforming optimization. We consider a scenario that one UAV serves multiple patrol USVs. In the simulation experiment, the positions of USVs are randomly generated. Then we set $p_k = 35\,\text{dBm}$, $\sigma^2 = -100\,\text{dBm}$ and $h_u = 200\,\text{m}$, which are some typical parameters of offshore area [8].

First, we evaluate the performance of the proposed UAV deployment approach. Figure 2 compares the random beam pattern with the designed beam pattern by solving problem P_1, where we assume the minimum rate constraints for each USVs are 1, 4, 4, 3 and 3 bps/Hz, respectively. It shows the uplink achievable sum rate and the optimal UAV position comparison between the proposed iterative algorithm and the grid-search method in the scenario of five USVs. Figure 2 (a) shows a 2D scatter plot of the USV-UAV deployment relationship, which is affected by USVs minimum rate constraints. It can be seen in Fig. 2 (b) that the proposed iterative algorithm has better performance in terms of the achievable sum rate than the grid-search method. Then, we evaluate the performance of the proposed beamforming algorithm. The beamforming vector

Algorithm 2 Implementation of $BR - \mu PSO$

```
Input:
  The number of antennas $M$
  The number of particle swarm $I$
  Maximum number of the iterations $T$
  The range of inertia weight $\omega_{\min}$ and $\omega_{\max}$
Output: $\varphi^{opt}$
Initialize the position $\mathbf{x}_i = \boldsymbol{\varphi}_i$ and velocity $\mathbf{v}_i$
Obtain the $\mathbf{rep}_i$, $d_{beyond}$ and $d_{in}$ according to (17)(18)(19)
while $t \leq T$ do
    Obtain the fitness function $F_t(\mathbf{X})$ according to (12)
    for $i = 1 : I$ do
        for $n = 1 : M$ do
            Update $[\mathbf{X}]_{j,n}^{t+1}$ and $[\mathbf{V}]_{j,n}^{t+1}$ according to (15) (16)
            if $\left|[\mathbf{X}]_{i,j}\right| > d_{beyond}$ then
                $\mathbf{X}_{i,j}{=}d_{beyond}\frac{[\mathbf{X}]_{i,j}}{\left|[\mathbf{X}]_{i,j}\right|}$
            end if
            if $\left|[\mathbf{X}]_{i,j}\right| < d_{in}$ then
                $\mathbf{X}_{i,j}{=}d_{in}\frac{[\mathbf{X}]_{i,j}}{\left|[\mathbf{X}]_{i,j}\right|}$
            end if
            if $\left|[\mathbf{p}_{best}]_{i,j}\right| < d_{in}$ then
                $[\mathbf{p}_{best}]_{i,j}{=}d_{in}\frac{[\mathbf{p}_{best}]_{i,j}}{\left|[\mathbf{p}_{best}]_{i,j}\right|}$
            end if
        end for
        Update $\mathbf{p}_{best}$
    end for
    Update $\mathbf{g}_{best}$
    if $(F_{t+1}(\mathbf{X}^*) < F_t(\mathbf{X}^*))$ then
        $\mathbf{T}[:, i] = [\mathbf{X}^*]^{t+1}$
        $\omega = \min(\omega_{\min}(1{+}\beta), \omega_{\max})$
    else
        Reset $\omega{=}\omega_{\min}$
    end if
    Reset $\mathbf{g}_{best}$ and $\mathbf{p}_{best}$ location and velocity matrices
    $t = t + 1$
end while
$\boldsymbol{\varphi}^{opt} = \mathbf{g}_{best}$
return $\boldsymbol{\varphi}^{opt}$
```

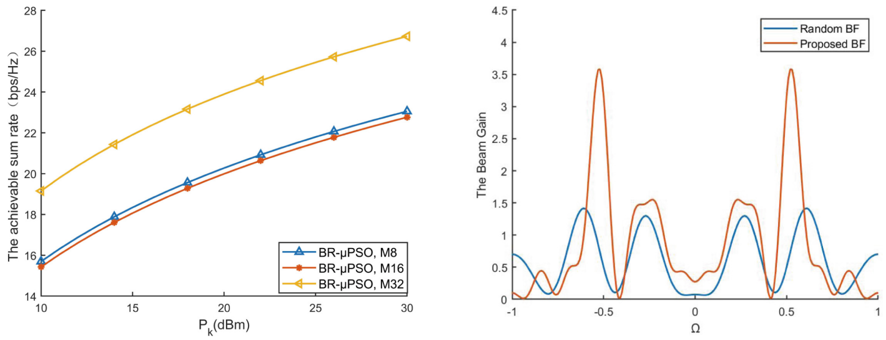


(a) UAV location deployment.

(b) The achievable sum rate via P_k.

Fig. 2. Location and performance of UAV deployments.

w is designed to approach the approximate beam gain of each USV. Figure 3 (a) shows the comparison between the achievable sum rate via the proposed beam pattern with different numbers of antennas against P_k, $M = 8, 16, 32$ and $K = 2$. Figure 3 (b) shows the beam gain comparison result between the random and proposed beamforming. We can observe that the proposed beamforming pattern is effective, and the beam gains are concentrated on the target USVs' directions.



(a) The achievable sum rate via P_k.

(b) The beam gain with different numbers of antennas.

Fig. 3. The achievable sum rate gain and beam gain of the proposed beam patterns.

5 Conclusion

This paper has investigated the joint beamforming and location deployment optimization problem (BLDO) for UAV relay, aiming to maximize the uplink achievable sum rate of the USV-UAV collaborative patrol scheme for maritime monitoring network. The original formulated BLDO problem has been decomposed into two sub-problems by the approximate beam pattern. The subproblem of deployment and beam gain allocation sub-problem has been first solved via the proposed alternating optimization. Then, the beamforming sub-problem has been tackled by the proposed $BR - \mu PSO$ algorithm. Simulation results have shown that the proposed scheme could effectively increase the performance of the achievable sum rate and beam gain in the USVs direction. For future work, we will investigate the effect of the unstable beam pointing problem.

Acknowledgements. The work was supported by the National Key Research and Development Program of China (No. 2019YFE0111600), National Natural Science Foundation of China (No. 61971083 and No. 51939001), LiaoNing Revitalization Talents Program (No. XLYC2002078) and the Major Key Project of PCL (PCL2021A03-1).

References

1. Jiao, F., Liu, J.: Study of inland ship water pollution control policy strategy based on Game theory. IOP Conf. Ser. Earth Environ. Sci. **191**(1), 012128 (2018)
2. Yan, X., Gong, J., Wu, Q.: Pollution source intelligent location algorithm in water quality sensor networks. Neural Comput. Appl. **33**(1), 209–222 (2020). https://doi.org/10.1007/s00521-020-05000-8
3. Kidston, D., Kunz, T.: Challenges and opportunities in managing maritime networks. IEEE Commun. Mag. **46**(10), 162–168 (2008)
4. Wang, H., Pan, P., Shen, L., Zhao, Z.: On the pair-wise error probability of a multi-cell MIMO uplink system with pilot contamination. IEEE Trans. Wireless Commun. **13**(10), 5797–5811 (2014)
5. Zhou, Z., Ge, N., Wang, Z.: Two-timescale beam selection and power allocation for maritime offshore communications. IEEE Commun. Lett. **25**(9), 3060–3064 (2021)
6. Zhu, L., Zhang, J., Xiao, Z., Cao, X., Wu, D.O., Xia, X.G.: Joint power control and beamforming for uplink non-orthogonal multiple access in 5g millimeter-wave communications. IEEE Trans. Wireless Commun. **17**(10), 6177–6189 (2018)
7. Su, X., Hui, B., Chang, K., Kim, S.: Improvement of the link reliability for ship ad-hoc network by employing multiple antennas. J. Korean Inst. Commun. Inf. Sci. **37**(12), 1065–1075 (2012)
8. Xiao, Z., Zhu, L., Choi, J., Xia, P., Xia, X.G.: Joint power allocation and beamforming for non-orthogonal multiple access (noma) in 5g millimeter wave communications. IEEE Trans. Wireless Commun. **17**(5), 2961–2974 (2018)
9. Duan, R., Wang, J., Zhang, H., Ren, Y., Hanzo, L.: Joint multicast beamforming and relay design for maritime communication systems. IEEE Trans. Green Commun. Networking **4**(1), 139–151 (2019)
10. Wang, J., et al.: Wireless channel models for maritime communications. IEEE Access **6**, 68070–68088 (2018)
11. Zhu, L., Zhang, J., Xiao, Z., Cao, X., Xia, X.G., Schober, R.: Millimeter-wave full-duplex UAV relay: joint positioning, beamforming, and power control. IEEE J. Sel. Areas Commun. **38**(9), 2057–2073 (2020)
12. Mozaffari, M., Saad, W., Bennis, M., Debbah, M.: Unmanned aerial vehicle with underlaid device-to-device communications: performance and tradeoffs. IEEE Trans. Wireless Commun. **15**(6), 3949–3963 (2016)
13. Bajwa, W.U., Sayeed, A., Nowak, R.: Sparse multipath channels: modeling and estimation. In: 2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, pp. 320–325. IEEE (2009)

A Monte Carlo Algorithm Based on Stochastic Geometry for Simulating Satellite Systems Interference

Zhaohua Qiu[1,2], Wen Wang[1](✉), and Dong Wei[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences,
Beijing 100093, China
qiuzhaohua@iie.ac.cn

[2] School of Cyberspace Security, University of Chinese Academy of Sciences,
Beijing 100049, China

Abstract. Providing truly ubiquitous Internet connectivity requires development of massive satellite constellations, whose interference scenario changes dynamically in time and space, leading to interference analysis becomes more complicated and challenging. Conventional interference analysis is restricted to few number of satellites with deterministic locations which is not capable of evaluating the performance of a massive satellite network consisting of thousands of satellites. In this paper, we propose a Monte Carlo algorithm based on stochastic geometry for simulating massive satellite systems interference. In our algorithm, we first utilize stochastic geometry to model the satellites' and user terminals' location as a randomly distributed points process on a sphere, imitating the high spatiotemporal dynamic interference characteristics of the scenario. Then, Monte Carlo is used to randomly sample the interference scenario in an infinite continuous timeline. According to Monte Carlo simulation, we finally calculate the cumulative probability distribution of interference indicator, which can be used as the statistical result of long-term interference of massive satellite constellations.

Keywords: Massive satellite constellations · Interference analysis · Stochastic geometry · Monte Carlo

1 Introduction

Since 2014, massive Non-Geostationary Orbit (NGSO) constellation provides global broadband access services has been a new trend. However, subject to limited spectrum resources, massive constellations will inevitably face shortage of frequency resources. Frequency band becomes more and more crowded, which causes serious interference and even makes the system unusable. On the other hand, if you want to launch new satellite system, you must provide interference analysis on the existing satellite systems to ensure that there will be no harmful interference. Therefore, it is necessary to investigate new evaluation methods to

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 52–63, 2022.
https://doi.org/10.1007/978-3-031-19214-2_5

analyze interference between massive NGSO; Moreover, it is realistic needs to design a set of practical interference evaluation methods in engineering.

This paper summarizes that satellite communication interference evaluation can be roughly divided into two categories: 1) computer simulation based on time frame (CS-BTF)); 2) Analysis based on mathematical model (A-BMM).

Because the trajectory of satellite changes continuously, CS-BTF captures the instantaneous trajectory of the satellite, discretizes the dynamic satellite, and then calculates the interference. Literature [8] submitted an approximate technology for "in-line" interference analysis between feeder links of NGSO MSS (mobile satellite services) service, in order to evaluate the feasibility of frequency sharing between two NGSO MSS networks; References [7,10] respectively realize the simulation of short-term interference of feeder links between two and four NGSO MSS networks; Literature [9] realized the C/I simulation of uplink and downlink of two NGSO networks for 10 d with period of 1 s. However, it takes a very long time to cover all possible interference scenarios through CS-BTF in case of large number of user terminals and NGSO satellites.

Some researchers have adopted analysis methods to calculate the statistical results of interference by means of probability theory and mathematical statistics. Literature [3] introduces an interference analysis method based on reference satellite position probability. Recently, some scholars have migrated the network model based on stochastic geometry in wireless network [1,2,5] to the massive NGSO constellations. Literature [12,13] proposed the analysis of downlink coverage and average rate of LEO satellite constellation based on stochastic geometry. Reference [15] proposed a mathematical framework to analyze uplink interference between ground network and satellite network by using stochastic geometry. However, a common problem of A-BMM is that the actual physical parameters have not been considered, such as antenna pattern and beam width. In addition, the theoretical model is too idealized, the calculation process of the formula derived is too complex, even there is not a closed form expression.

To shorten the simulation time of interference analysis in massive NGSO constellations, avoid the complexity of theoretical interference derivation and calculation, and can be applied to the in the real satellite systems, a Monte Carlo algorithm based on stochastic geometry for simulating satellite systems interference is proposed in this paper. Firstly, stochastic geometry is used to model the spatiotemporal characteristics of interference factors in the presence of massive satellites; Then, Monte Carlo method is used to simulate the model and calculate the cumulative probability distribution of interference indicator, which can be used as the statistical result of long-term interference of massive satellite constellations.

The contributions of this paper are as follows: (1) From the perspective of stochastic geometry, the satellites and users are modeled as a space point process for interference evaluation, which solves the difficulty of calculating the orbit equation of massive satellite constellation; (2) For the uplink and downlink interference scenarios, an interference evaluation algorithm suitable for uplink and downlink is proposed respectively; (3) We validate our interference prediction results based on the proposed algorithm by comparing it with the interference prediction results of the state-of-art satellite simulation software.

2 Problem Definition

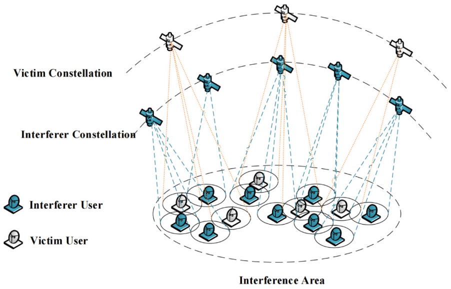
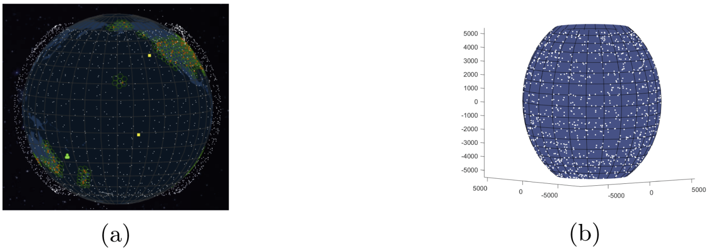


Fig. 1. Scenario of co-frequency interference between satellite constellations

As shown in Fig. 1, the satellite establishes a communication connection with the user through the antenna beam. Due to the dense distribution of satellites and users, the antenna beam reaching one user may cover the receiving area of other users, thereby causing interference to the received signals of other users. This paper believes that the two most critical factors affecting the calculation of massive constellation interference are the distribution of satellites and users, so we focus on the modeling of satellite and user distribution. First, we analyze the dynamic trajectory of the actual massive constellations, which is similar to the randomly distributed point process in space (Fig. 2); More importantly, relevant literatures [12,13] have also modeled the dynamic trajectory of satellites as a space point process. For the distribution model of users, we refer to the model of base stations in cellular networks [1,2,5], which also model users as a random point process on the Earth's surface.



(a)

(b)

Fig. 2. Starlink constellation vs our model. (a) Starlink constellation of 2806 satellites; (source: https://satellitemap.space/?constellation=starlink) (b) 2806 random points uniformly distributed on the surface of sphere.

3 Proposed Monte Carlo Algorithm Based on Stochastic Geometry

3.1 Calculation of C/I_{agg}

To simplify calculation, we assume that antenna can adjust its direction to capture satellites or user terminals. It means that in a communication link, satellite antenna and user antenna point at each other.

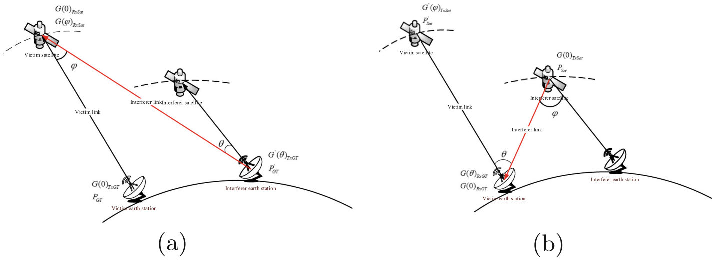


Fig. 3. Interference scenario. (a) uplink; (b) downlink.

Figure 3 is the simplest interference scenario of single interferer link, from which we can easily get C/I for uplink and downlink, respectively. Taking the uplink as an example, the calculation of C/I includes the following parts,

$$\begin{cases} C = P_{GT} + G(0)_{TxGT} + G(0)_{RxSat} - FSL \ (dBw) \\ I = P^{'}_{GT} + G^{'}(\theta)_{TxGT} + G(\varphi)_{RxSat} - FSL^{'} \ (dBw) \end{cases} \tag{1}$$

$$\left(\frac{C}{I}\right) = C - I \tag{2}$$

where P_{GT}, $P^{'}_{GT}$ are transmit power, $G(\cdot)_{TxGT}$, $G(\cdot)_{RxSat}$ are antenna gains of victim user terminals and satellite, respectively, and $G^{'}(\cdot)_{TxGT}$ is antenna gains of interferer user terminal.

According to literature [14], free space loss can be written as

$$\begin{cases} FSL = 20(log f_C + log d_C) + 32.45 \ (dB) \\ FSL^{'} = 20(log f_I + log d_I) + 32.45 \ \ (dB) \end{cases} \tag{3}$$

where, f_C, f_I are frequency victim link and interferer link respectively, in MHz. $d_C = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2}$, and $d_I = \sqrt{(x_1 - x_4)^2 + (y_1 - y_4)^2 + (z_1 - z_4)^2}$ which are distances of victim link and interferer link.

Let $\boldsymbol{a} = (x_1 - x_4, y_1 - y_4, z_1 - z_4)$, $\boldsymbol{b} = (x_2 - x_4, y_2 - y_4, z_2 - z_4)$, where (x_1, y_1, z_1), (x_2, y_2, z_2), (x_4, y_4, z_4) are position vectors of victim satellite, interferer satellite and interferer user terminal, respectively. Therefore, θ can be written as

$$\theta = cos^{-1}\frac{\boldsymbol{a} \cdot \boldsymbol{b}}{||\boldsymbol{a}|| \cdot ||\boldsymbol{b}||} \tag{4}$$

Let $\boldsymbol{c} = (x_3 - x_1, y_3 - y_1, z_3 - z_1)$, $\boldsymbol{d} = (x_4 - x_1, y_4 - y_1, z_4 - z_1)$, where (x_1, y_1, z_1), (x_3, y_3, z_3), (x_4, y_4, z_4) are position vectors of victim satellite, interferer user terminal and interferer user terminal, respectively. Therefore, φ can be written as

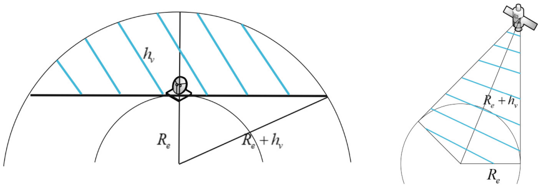
$$\varphi = cos^{-1}\frac{\boldsymbol{c} \cdot \boldsymbol{d}}{||\boldsymbol{c}|| \cdot ||\boldsymbol{d}||} \tag{5}$$

When interference scenario exists multiple interferer links, we need to consider aggregative C/I, which named C/I_{agg}. Based on C/I of single link, we summarize C/I_{agg} as follows

$$\frac{1}{C/I_{agg}} = \sum_{n=1}^{N} \frac{1}{C/I_n} \tag{6}$$

where N represents numbers of interferer links, C/I_n represents n-th interferer for victim.

Determine Number of Interferer Links N. Communication links are established in satellite systems when user terminal in the sight of satellite or satellite in the sight of user terminal. Figure 4, gives vision filed of victim system.



(a) (b)

Fig. 4. Vision field of user terminal and satellite. (a) downlink; (b) uplink. (Color figure online)

In Fig. 4, blue shadow part is vision field of an user terminal and a satellite. R_e is radius of earth, h_v is altitude of satellite. We use d_{max} represents maximum distance of user terminal or satellite can see, which can be written as

$$d_{max} = \sqrt{2R_e h_v + h_v^2} \tag{7}$$

Antenna Radiation Pattern. Antenna pattern refers to the law that the intensity of radio waves changes with the direction in space after being radiated by antenna. The most important parameter in antenna pattern is antenna gain $G(\cdot)$, which is used to measure the ability of the antenna to transmit and receive signals in a specific direction.

According to S.465 [6], the reference radiation pattern suitable for ground user terminal antenna is given by

$$G(\phi) = \begin{cases} 32 - 25log\phi \ dBi & \phi_{min} \leqslant \phi \leqslant 48^\circ \\ -10 \qquad\qquad dBi & 48^\circ \leqslant \phi \leqslant 180^\circ \end{cases} \tag{8}$$

where ϕ is off-axis angle of antenna. When the ratio of antenna diameter (D) and wavelength (λ) $D/\lambda < 50$, $\phi_{min} = max(2^\circ, 114(D/\lambda)^{-1.09})$.

According to ITU-R S.1528 [11], the reference radiation pattern for low Earth orbit satellite antenna, whose antenna diameter to wavelength ratio $D/\lambda < 35$, is given by

$$G(\phi) = \begin{cases} G_m & dBi \quad 0 \leq \phi \leqslant \phi_b \\ G_m - 3(\phi/\phi_b)^2 & dBi \quad \phi_b < \phi \leqslant Y \\ G_m + L_S - 25log(\phi/Y) & dBi \quad Y < \phi \leqslant Z \\ L_F & dBi \quad Z < \phi \leqslant 180^\circ \end{cases} \tag{9}$$

where G_m simplifies maximum gain in the main lobe, ϕ_b represents one half the 3 dB beamwidth in the plane of interest at the largest off-axis angle, L_s is main beam and near-in side-lobe mask cross point below peak gain, L_F is the far-out side-lobe level, equals to 0 for ideal patterns, and Y and Z are calculated by $Y = \phi_b\sqrt{-Ls/3}$ and $Z = Y \cdot 10^{0.04(G_m+L_sL_F)}$, respectively. Typically for a LEO satellite, $L_s = 6.75$ and $Y = 1.5\phi_b$.

3.2 Monte Carlo Method

The basic idea of Monte Carlo method is that when the problem to be solved is probability of a random event, the probability of this random event can be estimated by frequency of this event through some experimental simulation method. Therefore, the flow of interference calculation algorithm based on Monte Carlo is as follows. Firstly, establish the stochastic geometry model of interference scenario. And then the model is randomly simulated, statistically sampled to calculate the interference indicator C/I_{agg}. Finally, use the obtained results to calculate statistical estimate of C/I_{agg}, which is used as the approximate solution of the original problem.

To obtain statistical characteristics of C/I_{agg}, we define the cumulative distribution function of carrier interference ration probability. The probability is that C/I_{agg} less than a certain threshold T, that is:

$$p = \mathbb{P}\{C/I_{agg} < T\} \tag{10}$$

Before introduce proposed Monte Carlo algorithm based on stochastic geometry, the parameters used in the whole algorithm flow are described in the following Table 1:

Table 1. Parameters Description in the proposed algorithm

Parameter	Description	Parameter	Description
R_e	Earth radius	p_I	Interferer transmit power
h_V	Victim satellite altitude	f_V	Frequency of victim system
h_I	Interferer satellite	f_I	Frequency of interferer system
N_S	Number of interferer satellites	T	Threshold of C/I_{agg}
N_E	Number of interferer user terminal	num	Number of Monte Carlo simulations
p_V	Victim transmit power		

Based on the theory in Sect. 3.1, we use Monte Carlo Method to simulate interference scenario in satellite systems. Algorithm 1 and algorithm 2 describe the long-term interference estimation flow of uplink and downlink respectively.

```
Algorithm 1: Uplink interference estimation algorithm
Input: $R_e$, $h_V$, $h_I$, $N_S$, $N_E$, $p_V$, $p_I$, $f_V$, $f_I$, T, num.
Output: Output: carrier interference ration probability $p$.
for i=1:num do
    Generate victim satellite $S_V$ position $(x_{S_V}, y_{S_V}, z_{S_V})$ randomly;
    Calculate maximum sight of $S_V$ according to Eq. (12);
    Generate position $(x_{E_V}, y_{E_V}, z_{E_V})$ of victim user terminal $E_V$ randomly in the sight of $S_V$;
    Generate set of all interferer satellites positions randomly
    $C_{S_I} = \{(x^1_{S_I}, y^1_{S_I}, z^1_{S_I}), (x^2_{S_I}, y^2_{S_I}, z^2_{S_I}), (x^i_{S_I}, y^i_{S_I}, z^i_{S_I}), \ldots\}, i = 1, 2, \ldots, N_S$;
    Generate set of all interferer user terminals positions randomly
    $C_{E_I} = \{(x^1_{E_I}, y^1_{E_I}, z^1_{E_I}), (x^2_{E_I}, y^2_{E_I}, z^2_{E_I}), (x^i_{E_I}, y^i_{E_I}, z^i_{E_I}), \ldots\}, i = 1, 2, \ldots, N_E$;
    In the set $C_{E_I}$, find interferer user terminals which are in the sight of $S_V$ to compose set $C^{'}_{E_I}$;
    for $(x^j_{E_I}, y^j_{E_I}, z^j_{E_I})$ in $C^{'}_{E_I}$ do
        find the nearest satellite position $(x^k_{S_I}, y^k_{S_I}, z^k_{S_I})$ in $C_{S_I}$ to $(x^j_{E_I}, y^j_{E_I}, z^j_{E_I})$;
        After obtain $(x_{S_V}, y_{S_V}, z_{S_V})$, $(x_{E_V}, y_{E_V}, z_{E_V})$, $(x^j_{E_I}, y^j_{E_I}, z^j_{E_I})$ and $(x^k_{S_I}, y^k_{S_I}, z^k_{S_I})$, calculate $\theta$,$\varphi$ according to Eq. (4) and Eq. (5);
        Calculate $C/I_j$ according to Eq. (1) and Eq. (2);
    end
    Calculate aggregate interference $C/I_{agg}$ according to Eq. (6);
    Put $C/I_{agg}$ into CIRList;
end
for i in CIRList do
    if i¿T then
        count++;
    end
end
```

Algorithm 2: Downlink interference estimation algorithm

```
Input: $R_e$, $h_V$, $h_I$, $N_S$, $N_E$, $p_V$, $p_I$, $f_V$, $f_I$, T, num.
Output: Output: carrier interference ration probability $p$.
for $i$=1:num do
    Generate victim user terminal $E_V$ position $(x_{E_V}, y_{E_V}, z_{E_V})$ randomly;
    Calculate maximum sight of $E_V$ according to Eq. (12);
    Generate position $(x_{S_V}, y_{S_V}, z_{S_V})$ of victim satellite $S_V$ randomly in the sight of $E_V$;
    Generate set of all interferer satellites positions randomly
    $C_{S_I} = (x^1_{S_I}, y^1_{S_I}, z^1_{S_I}), (x^2_{S_I}, y^2 S_I, z^2_{S_I}), (x^i_{S_I}, y^i_{S_I}, z^i_{S_I}), \ldots, i = 1, 2, \ldots, N_S$;
    Generate set of all interferer user terminals positions randomly
    $C_{E_I} = (x^1_{E_I}, y^1_{E_I}, z^1_{E_I}), (x^2_{E_I}), y^2_{E_I}, z^2_{E_I}), (x^i_{E_I}), y^i_{E_I}, z^i_{E_I}), \ldots, i = 1, 2, \ldots, N_E$;
    Generate position $(x_{E_V}, y_{E_V}, z_{E_V})$ of victim user terminal $E_V$ randomly in the sight of $S_V$;
    Generate set of all interferer satellites positions randomly
    $C_{S_I} = \{(x^1_{S_I}, y^1_{S_I}, z^1_{S_I}), (x^2_{S_I}, y^2_{S_I}, z^2_{S_I}), (x^i_{S_I}, y^i_{S_I}, z^i_{S_I}), \ldots\}, i = 1, 2, \ldots, N_S$;
    Generate set of all interferer user terminals positions randomly
    $C_{E_I} = \{(x^1_{E_I}, y^1_{E_I}, z^1_{E_I}), (x^2_{E_I}, y^2_{E_I}, z^2_{E_I}), (x^i_{E_I}, y^i_{E_I}, z^i_{E_I}), \ldots\}, i = 1, 2, \ldots, N_E$;
    In the set $C_{E_I}$, find interferer user terminals which are in the sight of $S_V$ to compose set $C^{'}_{E_I}$;
    for $(x^j_{E_I}, y^j_{E_I}, z^j_{E_I})$ in $C^{'}_{E_I}$ do
        find the nearest satellite position $(x^k_{S_I}, y^k_{S_I}, z^k_{S_I})$ in $C_{S_I}$ to $(x^j_{E_I}, y^j_{E_I}, z^j_{E_I})$;
        After obtain $(x_{S_V}, y_{S_V}, z_{S_V})$, $(x_{E_V}, y_{E_V}, z_{E_V})$, $(x^j_{E_I}, y^j_{E_I}, z^j_{E_I})$ and $(x^k_{S_I}, y^k_{S_I}, z^k_{S_I})$, calculate $\theta$,$\varphi$ according to Eq. (4) and Eq. (5);
        Calculate $C/I_j$ according to Eq. (1) and Eq. (2);
    end
    Calculate aggregate interference $C/I_{agg}$ according to Eq. (6);
    Put $C/I_{agg}$ into CIRList;
end
for $i$ in CIRList do
    if $i$¿$T$ then
        count++;
    end
end
```

4 Performance Evaluation

4.1 Simulation Settings

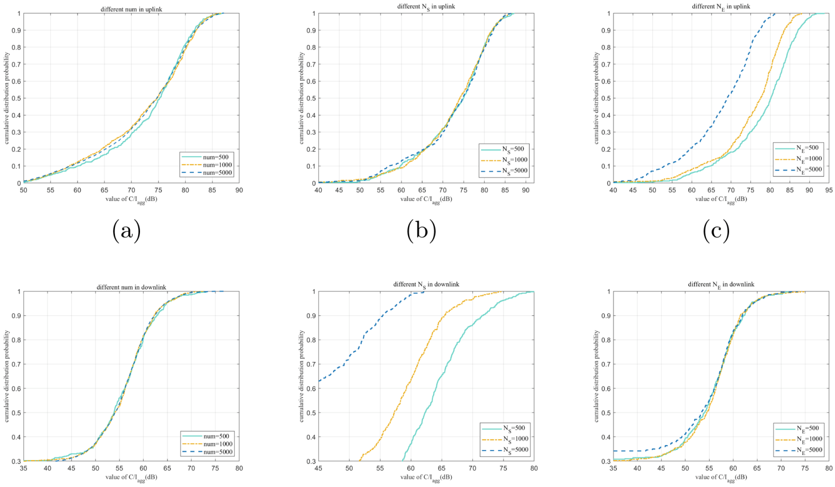
According to the algorithm proposed in Sect. 3.2, we use the parameters in the actual system to verify the performance of the algorithm [4]. The values of the parameters are shown in the following table.

Table 2. Constellation orbital configuration parameters and communication parameters of victim and interferer

Parameters	Victim constellation	Interferer constellation
Total number of satellites	720	1584
Orbital planes	18	72
Satellites per plane	40	22
Altitude	1200 km	550 km
Inclination	87.9°	53°
Frequency	14.25 GHz (uplink)	14.25 GHz (uplink)
	11 GHz (downlink)	11 GHz (downlink)
Maximum transmit gain of satellite	28.6 dBi	28.6 dBi
3dB half beamwidth of satellite transmitting antenna	6.5°	6°
Satellite antenna diameter	0.5 m	0.5 m
Antenna radiation pattern of satellite	ITU-R S.1528	ITU-R S.1528
Transmit power of satellite	14.3 dBW	11 dBW
Maximum receive gain of user terminal	49.2 dBi	44 dBi
User terminal antenna diameter	0.5 m	0.5 m
Antenna radiation pattern of user terminal	ITU-R S.465	ITU-R S.465

4.2 Simulation Results

In the numerical simulation, we obtain cumulative distribution probability p of C/I_{agg} corresponding to different thresholds T. Then, we discuss the influence of key parameters of the simulation on p, including the number of Monte Carlo num, the number of interferer satellites N_S and number of interferer user terminals N_E. Finally, based on the existing satellite interference simulation tools on the market, we compare the experimental results with them to verify the reliability of the algorithm proposed in this paper.



(a) (b) (c)

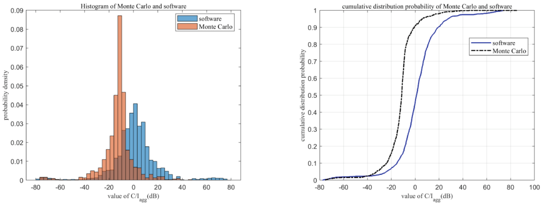
(d) (e) (f)

Fig. 5. Performance of key parameters in uplink and downlink.

Figure 5 shows uplink and downlink performance under different parameters. The results of different num are shown in Fig. 5(a) and (d). We can see that both of simulation results have converged when num = 500. Figure 5(b) presents uplink results under different N_S, with the increase of N_S, the change of cumulative distribution probability p is not significant. While Fig. 5(e) indicates that with the increase of N_S in downlink scenario, the curve about cumulative distribution probability tends to move to the left side, which implies that more interferer satellites, smaller C/I_{agg} in downlink. Figure 5(c) shows that with the increase of N_E in uplink scenario, the curve about p tends to move to the left side, which implies that more interferer user terminals, smaller C/I_{agg} in uplink. Figure 5(f) shows that number of N_E has little impact on cumulative distribution probability in downlink scenario.

Therefore, we conclude that number of interferer user terminals affect uplink interference, while number of interferer satellites affect downlink interference. The reason is that interference signal is transmitted from the user terminal in uplink, more interferer user terminal means more interference source. On the contrary, the interference source is on satellite in the downlink, increase of satellites means that the interference source also increases, so the interference signal received by the victim user terminal also increases.

In order to verify the reliability of the satellite systems interference evaluation algorithm proposed in this paper, we compare the algorithm proposed in this paper (Monte Carlo) with the existing satellite interference simulation software. Limited by the satellite interference simulation software, we only compared the downlink interference scenarios, and the satellite antenna beam was changed to point to the center of Earth rather than the user terminal, and user terminal antenna beam was changed to be vertical to the ground. The simulation results are shown in Fig. 6.



(a) (b)

Fig. 6. Downlink performance comparison between Monte Carlo algorithm based on stochastic geometry and software tool

From Fig. 6, we can find that the range of C/I_{agg} calculated by the algorithm proposed in this paper completely covers the results of software calculation, which proves that the algorithm can be applied to satellite systems interference simulation engineering in the real world. In addition, we find that the calculation results of the algorithm in this paper are generally smaller than those of the software. We believe that this may be because the software does not fully consider the complete long-term interference scenario of massive satellite systems due to the limitation of simulation time.

Finally, when the actual satellite system runs for 24 h, the simulation time of the software is 661.798 s; The simulation duration of Monte Carlo is 5.612 s. Therefore, the superiority of the algorithm proposed in this paper in simulation time exceeds the current mainstream software.

5 Conclusion

This paper presents a fast algorithm to evaluate the interference between different satellite systems. Firstly, the position of satellites is modeled randomly according to stochastic geometry, and then the probability distribution of C/I_{agg} is calculated by Monte Carlo simulation, which can represent the long-term interference by other satellite systems. Secondly, the key parameters which affect the interference result is analyzed and discussed. The results can be used as a reference for determining the optimal system scale in the construction of actual satellite system. Finally, the simulation results of the proposed method are compared with the current mainstream simulation software to illustrate the reliability of the proposed method.

References

1. Andrews, J.G., Baccelli, F., Ganti, R.K.: A tractable approach to coverage and rate in cellular networks. IEEE Trans. Commun. **59**(11), 3122–3134 (2011)
2. ElSawy, H., Hossain, E., Haenggi, M.: Stochastic geometry for modeling, analysis, and design of multi-tier and cognitive cellular wireless networks: a survey. IEEE Commun. Surv. Tutor. **15**(3), 996–1019 (2013)
3. Fortes, J.M.P., Sampaio-Neto, R., Amores Maldonado, J.: An analytical method for assessing interference in interference environments involving NGSO satellite networks. Int. J. Satell. Commun. Netw. **17**(6), 399–419 (1999)
4. Geng, J., Sun, D., Wang, W., Liu, Y.: Interference prediction between LEO constellations based on a novel joint prediction model of atmospheric attenuation. In: 2022 IEEE Wireless Communications and Networking Conference (WCNC), pp. 950–955. IEEE (2022)
5. Haenggi, M.: Stochastic Geometry for Wireless Networks. Cambridge University Press, Cambridge (2012)
6. ITU-R: Reference earth-station radiation pattern for use in coordiation and interference assessment in the frequency range from 2 to about 30 GHz. Recommendation S465-5, International Telecommunication Union, Geneva (1993)
7. ITU-R: Computer simulation of short-term interference between the feeder-links of two non-GSO MSS networks sharing the 5 and 7 GHz bands. Recommendation 4A/11, International Telecommunication Union, Geneva (1996)
8. ITU-R: “In-line” interference between the feeder-links of non-GSO MSS constellations. Recommendation 4A/3, International Telecommunication Union, Geneva (1996)
9. ITU-R: Interference between two NGSO FSS networks. Recommendation 4A/101, International Telecommunication Union, Geneva (1996)
10. ITU-R: Simulation of ‘in-line’ interference between the feeder-links of four separate non-GSO MSS constellations at 5/7 GHz. Recommendation 4A/109, International Telecommunication Union, Geneva (1996)
11. ITU-R: Satellite antenna radiation patterns for non-geostationary orbit satellite antennas operating in the fixed-satellite service below 30 GHz. Recommendation S1528, International Telecommunication Union, Geneva (2001)
12. Okati, N., Riihonen, T.: Nonhomogeneous stochastic geometry analysis of massive LEO communication constellations. IEEE Trans. Commun. **70**(3), 1848–1860 (2022)
13. Okati, N., Riihonen, T., Korpi, D., Angervuori, I., Wichman, R.: Downlink coverage and rate analysis of low earth orbit satellite constellations using stochastic geometry. IEEE Trans. Commun. **68**(8), 5120–5134 (2020)
14. Pahl, J.: Interference Analysis: Modelling Radio Systems for Spectrum Management. Wiley, Chichester (2016)
15. Yastrebova, A., et al.: Theoretical and simulation-based analysis of terrestrial interference to LEO satellite uplinks. In: GLOBECOM 2020 IEEE Global Communications Conference, pp. 1–6. IEEE (2020)

An Efficient Interference Calculation Model Based on Large Scale Constellations Probabilistic Analysis

Yiqing Liu[1,2], Weiqing Huang[1,2], Wen Wang[1](✉), Jingru Geng[1,2], and Zhaohua Qiu[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{liuyiqing,huangweiqing,wwen,gengjingru,qiuzhaohua}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Abstract. With the large scale NGSO constellation systems entering the blowout launch period, new challenges are expected for inter-system satellite coexistence due to the increase in scale of the constellations and the complexity of the interactions, where the problems of frequency compatibility between NGSO system and GSO system are the most prominent. However, the existing methods normally analyze the interference through real-time position acquisition, visible area judgment and single link calculation, as for the large scale constellations under construction, these methods still have great challenges to large amount of data stored in constellation positions, long simulation duration and high requirements for simulation equipment. In this circumstance, we establish a high efficient interference calculation model apply for large scale constellations. Our solution stems from the satellite position probability, by meshing the visible area and deriving the distribution of other interfering satellites in visible area through the initial satellite, we can directly analyze aggregate interference. The proposed model not only has the advantages of low algorithm complexity and high calculation efficiency, but also suitable for large scale constellations with different configurations as well as not depend on computer simulation equipment. The theoretical and simulation results both show that the proposed model has advantages in the accuracy of interference calculation and the duration of interference simulation.

Keywords: Large scale NGSO system · GSO system · High efficient interference calculation · Satellite position probability

This work is supported by the National Key Research and Development Program of China (Grant No. 2021YFB2910107).

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 64–79, 2022.
https://doi.org/10.1007/978-3-031-19214-2_6

1 Introduction

At present, SpaceX has launched 2566 *Starlink* non-geostationary satellite orbit ("NGSO") satellites in a new lower shell of the constellation, marking the arrival of the era of large scale constellations and ushering a change in human telecommunications. The large scale constellations have the advantages of wide transmission bandwidth, high communication quality, low transmission delay, and low terrain limitation [1,2]. Compared with the traditional geostationary earth orbit ("GSO") satellite, large scale NGSO constellations are more suitable for the future 6G communication network [3,4], so as to achieve ubiquitous, 100% geographic coverage with terrestrial-satellite communication networks [5].

The progress of large scale constellations in foreign countries has developed rapidly, the more representative systems are *Starlink*, *OneWeb* and *Kuiper*. *Starlink* system will eventually build nearly 42000 satellites, *OneWeb* system plans to launch 8368 satellites and *Kuiper* system has completed the application of 3236 satellites. Signal of the large scale constellations can overlap almost everywhere on the earth, and thus would have considerable effect on the existing systems in services. Analysing the information submitted to the International Telecommunication Union ("ITU") [6] and the Federal Communications Commission ("FCC") [7], the frequency resources used by these NGSO systems are concentrated mainly in the Ku and Ka frequency bands, which inevitably causes co-frequency conflicts to GSO systems [8]. According to Article 9 and 11 of the ITU Radio Regulations ("RR"), in the stage of satellite frequency and orbit coordination, the GSO satellite network has the priority in application and deployment, while the large scale NGSO satellite network shall send a coordination request to the GSO network, as well as using the interference distribution and the interference analysis to prove. However, in the process of coordination, the two parties will continue to discuss and consult, the traditional Monte Carlo interference calculation model based on time-slice often runs for several days, which is not suitable for interference calculation between large scale constellations and GSO systems [9].

Most of the current researches focus on obtaining more accurate interference analysis models in different scenarios while less research on simplifying the calculation model of large scale satellite system under real constellation configurations. The most commonly used and widely recognized interference calculation method is Monte Carlo calculation model based on time-slice [10], which storing the positions of satellites and earth stations at each time, then to judge the visual range and calculate the interference links. This method is suitable for scenarios with a small number of satellites, when conducting interference analysis between large scale constellations, the performance of computer equipment and simulation duration will become restrictive factors. Yastrebova [11] and Okati [12] used stochastic geometry and Poisson point process to construct satellite network and calculate interference and analysis, which is a mathematical analysis model and still has some differences with the actual satellite constellation configuration, whether it will become a general model still has some doubts. Fortes proposed to complete the interference analysis by the probability method [13,14]. By deriving the satellite probability density function and using the global integral to calculate and analyze interference distribution. However, there is no specific example analysis and meshing criteria. Lin carried out further analysis on this basis [15], the accuracy is verified by giving specific experimental methods, analysis and meshing standards. Nevertheless, tens of thousands of satellites in large scale constellation bring new problems to the construction and position data storage of the whole constellation. Therefore, the research on large scale constellations interference calculation needs to be more in-depth.

Contributions of this paper can be briefly summarized as follows:

- We propose an efficient and fast interference calculation model based on satellite probability analysis, which can be applied by different configurations of large scale constellations, as well as can provide a reliable reference for the frequency and orbit coordination between satellite systems.
- The proposed model can save the time when establishing the large scale constellation, and it saves the storage space and computation when calculate the position of whole system at each time, which can improve the computing efficiency and reduce the performance requirements of simulation equipment.
- Combined with actual systems parameters, the interference calculation model based on position probability and time-slice are used for calculation respectively. The theoretical and simulation results show that the accuracy proposed in this paper is the same or even better than that of the time-slice, and the total simulation duration has been improved at least 518 times.

The structure of this paper is as follows. Section 2 introduces the interference scenario and probability analysis method. Section 3 establishes the visual area and then calculates and statistics the aggregate interference. Section 4 describes the simulation results and effectiveness of the proposed model. Section 5 gives the conclusion.

2 Scenario Model

2.1 Interference Scenario

The large scale constellation satellites provide seamless coverage of service areas within their orbital inclination. The GSO system, due to its unique orbital characteristics and launch configuration, its earth stations are deployed mainly at latitude 30° within, and principally provides broadband services for middle and low latitudes, which makes a huge difference from the original design intention of the large scale NGSO systems to provide global services. Meanwhile, the NGSO constellation systems have a huge number of satellites, multiple NGSO satellites will appear in the visual area of the GSO earth station at any time, and limited by the current scarce satellite communication spectrum resources, the large scale NGSO constellation systems will inevitably interfere to the GSO system. This paper considers the interference scenario of spectrum coexistence between the GSO system and the large scale NGSO system, and the interference distribution

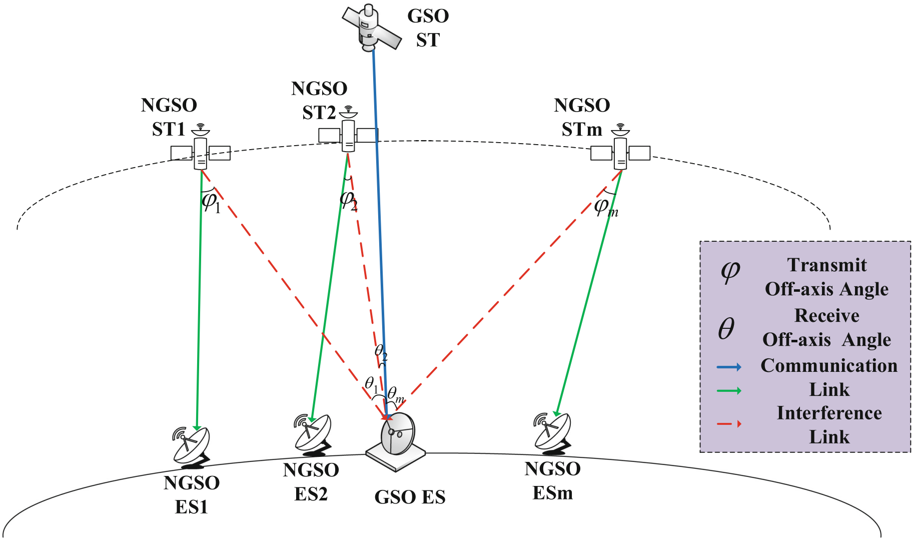


Fig. 1. Downlink interference scenario between the GSO system and the large scale NGSO system

is calculated fast and efficiently according to the corresponding mathematical model.

Figure 1 contains the GSO and the NGSO satellite systems. The GSO system consists of one satellite and one earth station, while the NGSO system consists of a NGSO constellation and several NGSO earth stations. In this paper, the interference scenario is established, then calculating the aggregate interference of the NGSO constellation to the GSO earth station based on the position probability of each initial NGSO satellite.

In the interference scenario, the links from GSO satellite to GSO earth station and NGSO satellites to NGSO earth stations are useful link, which represented by the solid blue line and solid green lines. And the links between NGSO satellites and GSO earth station are interference links, they are indicated by the red dotted lines. Angle φ represent the transmit off-axis angle and θ represent the receive off-axis angle.

2.2 Probability Method

Probabilistic analysis is to identify the sampling density required of each probability distribution and directly in sequence check all possible values to derive a quasi-analytic calculation of the S[X]. This methodology is the basis of Rec. ITU-R S.1529 [16], which can be used to analyze interference between NGSO Fixed-Satellite Service (FSS) systems and GSO or other NGSO systems. The key concept of the method is to identify the orbital shell, by controlling the simulation step, each cell should be sufficiently small that the link or interference metric would not vary significantly across it, then the probability that the satellite is in the cell can be calculated.

For example, the Fig. 2 draws the orbital shell of the interfering satellite within the visual area of the GSO earth station and meshes it into small cells. For the satellite communication system with determined operation mode and communication parameters, when the initial satellite is determined, the position distribution of the whole constellation can be determined. Setting the initial satellite in the center of the small cell to determine the distribution of interfering satellites in the visual area, then the aggregate interference can be calculated. Traversing all small cells in the visual area, and analyzing the interference distribution. In the Fig. 2, the blue square background is the initial satellite set in position 1, and the blue satellites are the distribution of interfering satellites at position 1, the orange satellites are at initial position 2 distribution.

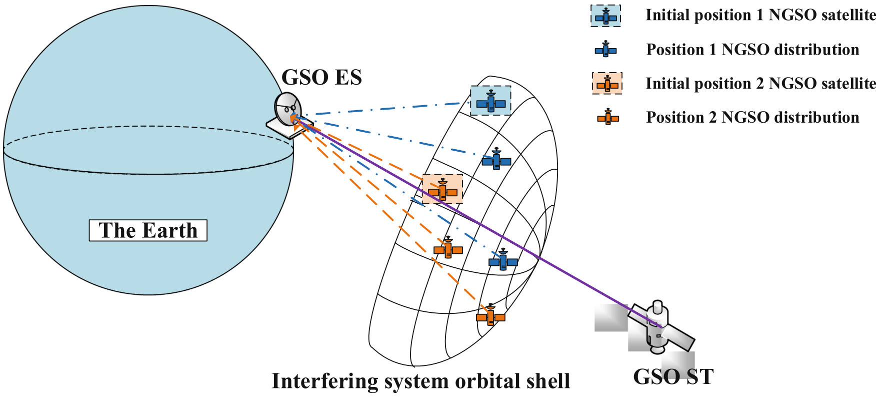

Initial position 1 NGSO satellite
Position 1 NGSO distribution
Initial position 2 NGSO satellite
Position 2 NGSO distribution
GSO ES
The Earth
Interfering system orbital shell
GSO ST

Fig. 2. Cells division of the visual area and the distribution of interfering satellites

3 Interference Calculation Model

3.1 Determine the Visual Area

GSO System Model

In general, the representation of the earth station and the satellite adopts a geographic coordinate system in which the positions' representation on the earth surface in form of longitudes and latitudes. For subsequent calculate the interference with the NGSO system, the geographic coordinate systems need to be converted into the ECF geocentric coordinate system.

Assume that the sub-satellite point of a GSO satellite, GSO earth station and NGSO earth station are $(\lambda lon_w, \lambda lat_w)$, converting it to a geocentric coordinate system, which can be expressed as Eq. (1):

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = (R_e + H_w) \begin{bmatrix} \cos \lambda lat_w \cos \lambda lon_w \\ \cos \lambda lat_w \sin \lambda lon_w \\ \sin \lambda lat_w \end{bmatrix} \tag{1}$$

In Eq. (1), R_e is the radius of the earth, H_w is the altitude above ground, and λlon_w and λlat_w express longitude and latitude respectively, where the GSO satellite is running over the equator, the latitude of the GSO satellite is at 0°. The altitude of a GSO earth station is negligible compared with the radius of the earth.

Visual Area Determination

The GSO earth station has its fixed visual field angle and is aligned with its own GSO satellite. As shown in Fig. 3, O is the geocentric point, the green area is the equatorial plane and the purple area is the visible range of the GSO earth station.

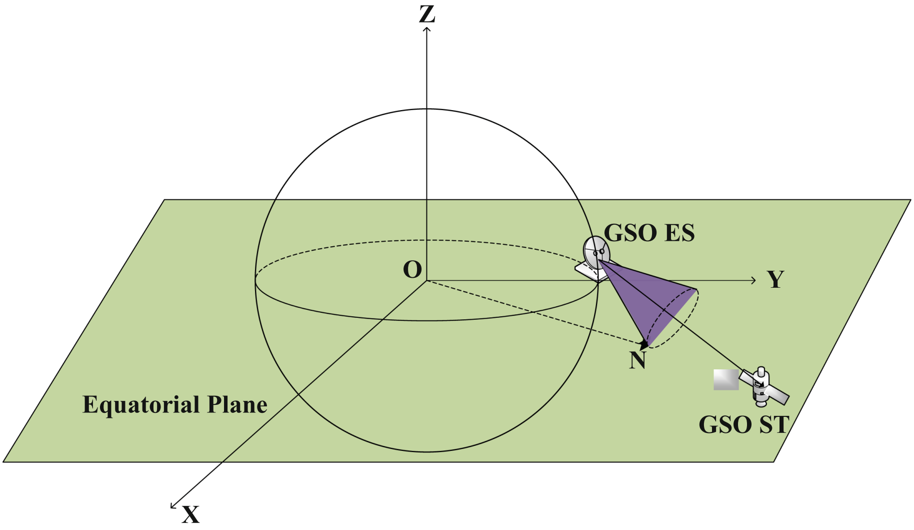


Fig. 3. Schematic diagram of the GSO earth station visual area

Through the coordinate conversion in the previous step, the visual area of the GSO earth station can be determined, and its expression is as follows:

$$\begin{cases} \cos \alpha = \dfrac{\vec{EG} \cdot \vec{EN}}{\left|\vec{EG}\right| \cdot \left|\vec{EN}\right|} \\ \left|\vec{ON}\right| = (R_e + H_w) \end{cases} \tag{2}$$

where, α is the maximum visual field angle of the GSO earth station, E is GSO earth station and G is GSO satellite, N is the intersection point of the edge of GSO earth station visible area and the NGSO satellite system orbital shell. Taking the north pole as the azimuth 0°and traversing all azimuth angles, it can obtain the intersection point set.

Visual Area Meshing

Based on the intersection point set, projecting the edge of GSO earth station visible area onto the earth's surface, the expression is as follows:

$$\begin{cases} \lambda lat_w = \arcsin(\frac{z_w}{R_e+H_w}) \\ \lambda lon_w = \arctan(y_w/x_w) \end{cases} \tag{3}$$

Meshing the projection area and calculating the satellite occurrence probability. The key step is dividing the projection area to sufficiently small cells, which makes the probability of NGSO satellite appearing at any position of the cell is the same. In this paper, the divided cells varies according to the change of simulation accuracy, and the cells are small enough. Therefore, the central coordinate of the cell represents the position of the cell.

The probability formula of the Walker Constellation in different longitudes and latitudes is as Eq. (4), where I represents the orbital inclination:

$$p_X(\lambda lon_w, \lambda lat_w) = \begin{cases} \frac{1}{2\pi^2} \frac{\cos(\lambda lat_w)}{\sqrt{\sin^2(I) - \sin^2(\lambda lat_w)}} & \begin{matrix} -\pi < \lambda lon_w < \pi \\ -I < \lambda lat_w < I \end{matrix} \\ 0 & \text{others} \end{cases} \tag{4}$$

3.2 Determine the Position of the NGSO Satellites in the Visual Area

Walker Constellation has the characteristics of uniform satellite distribution and good global coverage, which makes it the most commonly used constellation configuration for NGSO constellation application and initial deployment. Therefore, the scenario selected in this paper will be focused on the efficient interference analysis and calculation in the Walker constellation.

Initial Satellite Position Determination

When building Walker Constellation, the six Kepler orbital elements are often used to describe the satellite position. In ITU recommendation S.1325 [17], existing studies can convert the position of a NGSO satellite in the geodetic coordinate system into the ECF geocentric coordinate system. The expression of the NGSO satellite position in ECF coordinate system is:

$$\begin{bmatrix} x_{ns} \\ y_{ns} \\ z_{ns} \end{bmatrix} = r \begin{bmatrix} \cos \Omega \cos M - \sin \Omega \cos I \sin M \\ \sin \Omega \sin M + \cos \Omega \cos I \sin M \\ \sin I \sin M \end{bmatrix} \tag{5}$$

where (x_{ns}, y_{ns}, z_{ns}) is the NGSO satellite coordinate in ECF, r is the NGSO satellite orbital radius, Ω is the right ascension of ascending node (RAAN) of the orbit, M is the mean anomaly, and I is the inclination of the orbit.

According to the coordinate conversion formula (5), through the known of the initial NGSO satellite position in ECF coordinate system, that would determine the initial satellite position in geodetic coordinate system. It's important to note that the orbital inclination changes according to the change of orbit configuration, when the orbital configuration is determined, the I is a constant.

$$\begin{cases} M = \arcsin(\frac{z_{ns}}{r \cdot \sin I}) \\ \Omega = \arccos(\frac{x_{ns}/r \cdot \sin M + y_{ns}/r \cdot m}{\sin M \cos M + m^2}) \end{cases} \tag{6}$$

where $m = \cos I \sin M$.

Determine the Distribution Position of the NGSO Satellites Within the Visual Range

The Walker constellation was first proposed in 1970, which can be divided into star constellation and rose constellation, and it is currently adopted by most systems due to its characteristics of uniform distribution [18]. Walker Constellation can be fully described by three parameters: $T/P/F$, where T represents the total number of the constellation satellites, P represents the number of the orbital planes, F represents the phase factor of phase difference between adjacent satellites in adjacent orbital planes, which the value is usually between 1 to $P-1$.

When the initial satellite in Walker constellation is determined, which means the (Ω_0, M_0) is known, then according to the formula (7) of star constellation, the distribution of NGSO satellites in the GSO earth station visible area can be determined.

$$\begin{gathered} \begin{cases} \Omega_{i,j} = \Omega_0 \pm i(\pi/P) \\ M_{i,j} = M_0 \pm 2\pi(Fi/T + Pj/T) \end{cases} \\ i = 0, ..., roundup((dif\_lon \cdot P)/\pi) \\ j = 0, ..., roundup((dif\_lat \cdot T)/(2\pi \cdot P)) \end{gathered} \tag{7}$$

where, i represents i-th orbital plane, j represents j-th satellite in i-th orbital plane, roundup means rounding up function, dif_lon is the visible area maximum longitude difference and dif_lat is the visible area maximum latitude difference.

When the configuration of Walker Constellation is rose, the total period of RAAN difference between adjacent tracks is 2π, therefor, the formula can be expressed as:

$$\begin{gathered} \begin{cases} \begin{cases} \Omega up_{i,j} = \Omega_0 \pm i(\pi/P) \\ Mup_{i,j} = M_0 \pm 2\pi(Fi/T + Pj/T) \end{cases} \\ \begin{cases} \Omega down_{i,j} = \Omega_0 \pm (i + \frac{P}{2})(\pi/P) \\ Mdown_{i,j} = M_0 \pm 2\pi(Fi/T + P(j + \frac{T/P}{2})/T) \end{cases} \end{cases} \\ i = 0, ..., roundup((dif\_lon \cdot P)/\pi) \\ j = 0, ..., roundup((dif\_lat \cdot T)/(2\pi \cdot P)) \end{gathered} \tag{8}$$

Because the ascending orbits of rose constellations are distributed in the period range of 2π, the ascending orbit and the descending orbit will intersect. Therefore, within the visual range of GSO earth station, it should be discussed the distribution of interfering satellites in ascending orbit and descending orbit respectively.

3.3 Interference Calculation

This paper takes the downlink interference scenario of the NGSO system to the GSO system as an example. For each NGSO satellite in the visual area of the GSO earth station, it will interfere to the GSO earth station, the single link interference formula is as follows:

$$I\mathrm{m} = P_{t,m} \cdot O_{overlop} \cdot G_t\left(\varphi_m\right) \cdot G_r\left(\theta_m\right)/L_{fs,m} \tag{9}$$

where, I_m represents the interference power from the $m-th$ NGSO satellite to the GSO earth station, $P_{t,m}$ is the transmitting power of the $m-th$ NGSO satellite beam, O_p is the frequency overlap factor, $G_t\left(\varphi_m\right)$ is the gain of NGSO satellite where the off-axis angle between two links equals to φ_m and $G_r\left(\theta_m\right)$ is the gain of GSO earth station when the off-axis angle between two links equals to θ_m. L_{fs} is the free space path loss. Therefore, it needs to be adjusted by the frequency overlap factor. The formula for O_p is as follows:

$$\begin{gathered} O = \min\left\{B_g, B_n, \tfrac{(B_g+B_n)}{2} - \left|f_g - f_n\right|\right\} \\ O_p = \tfrac{O}{B_n} \end{gathered} \tag{10}$$

where, B_g is the GSO link bandwidth, B_n is the NGSO signal bandwidth, f_g is the GSO signal centre frequency and f_n is the NGSO signal centre frequency.

The free space path loss is related to the distance between the NGSO satellite and the GSO earth station, the formula is as Eq. (11):

$$L_{fs} = (\frac{4\pi r}{\lambda})^2 = (\frac{4\pi \cdot d_{km} \cdot f_{MHz} \cdot 10^9}{c})^2 \tag{11}$$

where, d_{km} is the distance between two separated location in km and f_{MHz} is the centre frequency in MHz.

Due to the large scale NGSO system, multiple NGSO satellites will appear simultaneously in the visual area of the GSO earth station. Therefore, the aggregate interference of the NGSO constellation to the GSO earth station will be calculated finally, the calculation formula is as follows:

$$I_{agg} = \sum_{m}^{M} 10^{I_m/10} \tag{12}$$

where, M is the number of NGSO satellites in the GSO earth station visible area. Aggregate interference-to-noise ratio (I_{agg}/N) is:

$$I_{agg}/N = I_{agg}/(K \cdot T \cdot B) \tag{13}$$

where, K is the Boltzmann constant, $K = 1.38 \times 10^{-23} J/K$, T is the equivalent noise temperature of the GSO earth station receiving system and B is the GSO earth station receiving bandwidth.

3.4 Interference Probability Distribution Statistics

In previous sections, firstly meshing the visual area of the GSO earth station, then according to the cell division accuracy, we can determine the coordinates of the center point of the small cell in the visible area of the GSO earth station. The initial satellite is placed at the center of the each cell respectively, Using Eq. (4) to determine the position occurrence probability of the initial satellite.

In order to overcome the difference of the cell division accuracy and make the statistical results of interference distribution more realistic, it is necessary to normalize the initial NGSO satellite position probability. The normalization formula is as follows:

$$p_X(\lambda lon_w, \lambda lat_w) = \frac{p'_X(\lambda lon_w, \lambda lat_w)}{\sum\limits_{w=1}^{W} p'_X(\lambda lon_w, \lambda lat_w)}, \quad \mathrm{w = 1, 2, ..., W} \tag{14}$$

where, $p_X(\lambda lon_w, \lambda lat_w)$ is the position occurrence probability of the $w-th$ initial NGSO interference satellite after normalization, $p'_X(\lambda lon_w, \lambda lat_w)$ is the position occurrence probability of the $w-th$ initial NGSO interfering satellite determined by the longitude and latitude of the cell center, W is the total number of small cells in the visual area of the GSO earth station.

When the NGSO constellation configuration and initial satellite are determined, the global distribution of interfering constellation is determined. The distribution and specific position of the interfering satellites in the visual area of the GSO earth station are derived from the initial NGSO satellite position. Using the calculation formula of interference-to-noise ratio, the aggregate interference I_{agg}/N corresponding to the $w-th$ initial satellite position can be calculated. Traversing and calculating the aggregate interference corresponding to all cells in the visual area of the GSO earth station, and statistical aggregate interference probability distribution. The statistical method is as follows: incorporation the $I_{agg,w}/N$ into corresponding interval I_{agg}/N, the probability corresponding to the $I_{agg,w}/N$ will be accumulated as the probability value of the current interval I_{agg}/N. Setting s dB as interval granularity, counting the probability of the $I_{agg,w}/N$ in interval $[a-s, a)$. Where a is the statistical of the $I_{agg,w}/N$ value, and the statistical formula is as Eq. (15):

$$p(a) = \sum_{w=1}^{W} p(I_{agg}\_w/N), \mathrm{a - s} \leq I_{agg}\_w/N < a \tag{15}$$

Repeating above process for all aggregate interference and position probability, and finally obtaining the probability distribution of the I_{agg}/N.

4 Experiment

4.1 Simulation Model

To verify the accuracy and efficiency of the proposed model, the interference scenario is constructed according to the parameters in Table 1. The interference calculation model based on satellite position probability and Monte Carlo interference calculation model based on time-slice are used for interference analysis, then comparing the simulation results and simulation duration.

Table 1. Satellite systems communication parameters

Parameters	SinoSat-5	Constellation B	Constellation C
Earth station field angle (°)	45	53	42
Constellation orbital altitude (km)	35794	800	550
Constellation orbital inclination (°)	0	66	53
Number of satellites	1	1584	3872
Number of orbital planes	1	72	44
Constellation phase factor	/	1	1
Constellation configuration	/	rose	star
Frequency (GHz)	12.5	12.5	12.5
Bandwidth (MHz)	250	250	250
Satellite transmitting antenna peak gain (dBi)	37	26.3	26.3
Half-power beam angle of satellite transmitting antenna (°)	0.82	3.2	3.2
Satellite transmitting antenna pattern	S.672	S.1528 [19]	S.1528
Satellite transmitting power (dBW)	16.3	8.5	8.5
Earth station receiving antenna peak gain (dBi)	52.9	19.9	19.9
Half-power beam angle of earth station receiving antenna (°)	0.4	8.8	8.8
Earth station receiving antenna pattern	S.1428 [20]	AP8	AP8
Earth station receiver system noise temperature (K)	240	120	120
Propagation model	P.525	P.525	P.525

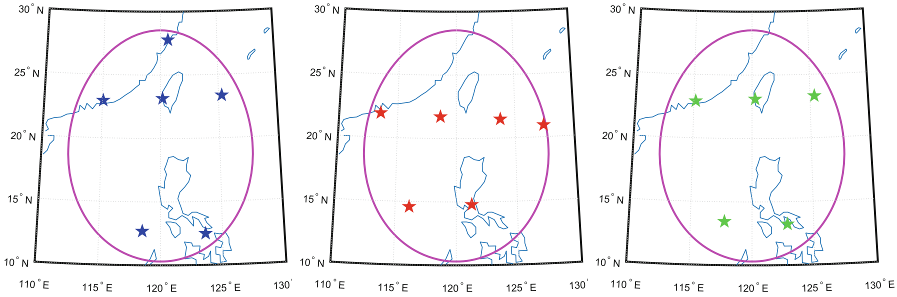
Table 1 shows the parameters of the interfered GSO system. In this paper, $SinoSat-5$ GSO system will be selected as an example. The $SinoSat$-5 satellite operates on the equatorial synchronous orbital plane and stationary relative to the earth station, the satellite transmitting antenna and the receiving antenna of the earth station are always targeting. Since the $SinoSat-5$ system has been in orbit, querying the data submitted to ITU and determining the system communication parameters, which will be better fit the actual scenario.

Table 1 also shows the parameters of the interfering NGSO constellation system B and C, where B and C choose different constellation configurations and orbital parameters respectively. In order to verify that the proposed calculation model is more suitable for large scale constellations, the communication parameters adopted in this paper will refer to the application documents submitted by

NGSO system to the ITU. Since the NGSO system configuration will be large, then the fixed antenna of satellites can provide global services.

4.2 Results and Discussion

Deploying the GSO earth station at $(120°E, 27°)$, and the GSO satellite operating at $(120°E, 0°)$. Due to the limitation of antenna pointing and visual field angle, the visual area of the GSO earth station is an irregular area. In Fig. 4, three initial positions are selected respectively, and the distribution of interfering satellites is drawn within the visual area of GSO earth station, where the pink circle is the visual area edge line of GSO earth station on the earth's surface, the blue star is the distribution of interfering satellites in the initial position 1, the red and green stars are at the initial positions 2 and 3 respectively.

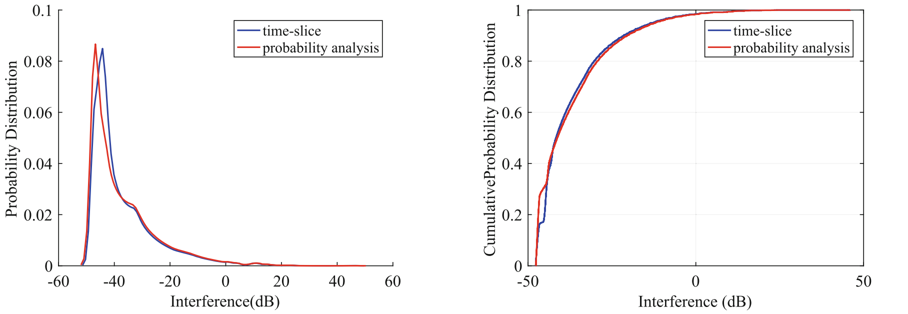


(a) blue stars in position 1 (b) red stars in position 2 (c) green stars in position 3

Fig. 4. The distribution of interfering satellites in the visual area (Color figure online)

As show in Fig. 5(a), the satellite aggregate interference distribution curves simulated by different models, in which the blue line is the Monte Carlo interference calculation model based on time-slice, and the red line is the proposed interference calculation model based on satellite position probability using in this paper. The experimental results of the two models fit well, and there is only a slight deviation near the peak of the probability distribution. Figure 5(b) draws the cumulative distribution curve by two models.

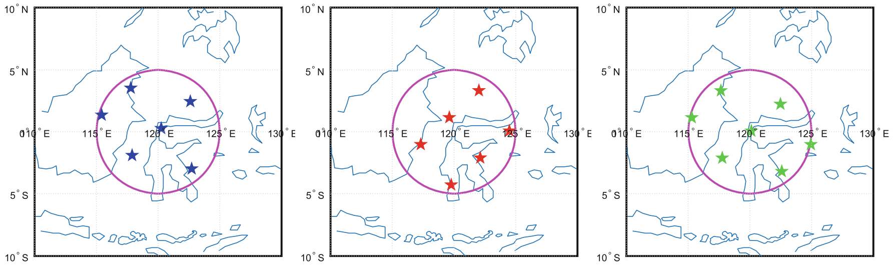
We deploy the GSO earth station at $(120°E, 0°)$, and analyzing the aggregate interference from constellation C to the GSO earth station when the GSO satellite is directly above the GSO earth station. Figure 6 shows the projection edge of the visual area and the distribution of interfering satellites of the GSO earth station. Since the GSO earth station is setting directly below the GSO satellite, the projection boundary of the visible area is a positive circle. Figure 7(a) and 7(b), show the aggregate interference probability distribution and cumulative distribution received by the GSO earth station respectively when the number of interfering constellations increases, in which the blue line using the Monte



(a) probability distribution diagram (b) cumulative probability distribution diagram

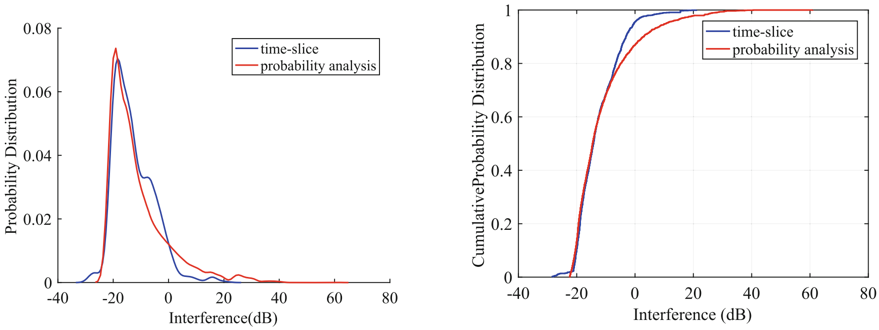
Fig. 5. Probability distribution and cumulative probability distribution of the interference

Carlo interference calculation model and the red line using the proposed model in this paper. In this experiment, the aggregate interference probability distribution of the two models is quite different between (−10) dB to 0 dB. This is because the calculation model based on time-slice is the real-time satellite position acquisition in the simulation scenario, and the satellite attitude and orbit control are not considered, which would caused satellite position deviation due to earth perturbation, atmospheric friction and other factors. In the actual operating environment, the satellite will operate according to the predetermined orbit. When the interfering link approaches to the GSO communication link, a slight position difference of interfering satellites will have a great impact on the antenna pointing and antenna angles, and then affect on the gain and aggregate interference. In addition, the calculation model based on satellite position probability can more comprehensively include extreme cases such as the worst interference scenario, and provide more reliable data support for interference analysis.



(a) blue stars in position 1 (b) red stars in position 2 (c) green stars in position 3

Fig. 6. The distribution of interfering satellites in the visual area (Color figure online)



(a) probability distribution diagram (b) cumulative probability distribution diagram

Fig. 7. Probability distribution and cumulative probability distribution of the interference

Table 2 compares the constellation scenario construct and aggregate interference calculate time used by interference calculation model based on time-slice and on satellite position probability respectively. The workstation used is Dell 5530, which can build up to about 4000 satellites. Using the calculation model based on time-slice, the total simulation time is 10 d. The proposed interference calculation model saves the construction time of interference constellation and increases the interference calculate time thousands of times. When the number of the interfering constellation increases, the constellation construction and simulation duration of the calculation model based on time-slice increases significantly, while the simulation duration of the calculation model based on satellite position probability hardly changes, and even the calculation time is shrinking. This is because the orbital altitude of constellation C is smaller than that of the constellation B compared with the increase of constellation size. With the decrease of the orbital altitude of NGSO constellation, the visual sphere of the GSO earth station will shrink. Therefore, the main factor affecting the calculation time of this model is the orbit altitude of interfering constellations, which is less related to the number of constellations. Most of the large scale constellations are Low Earth Orbit constellations, and the orbital altitude is low or even very low, then the greatest time-consuming impact on the proposed calculation model can be ignored.

Table 2. The simulation duration of two interference calculation models

	Constellation construction	Interference calculation
Interference of B to GSO based on time-slice	1716 (s)	10564 (s)
Interference of B to GSO based on probability distribution	0 (s)	23.7 (s)
Interference of C to GSO based on time-slice	24684 (s)	163687 (s)
Interference of C to GSO based on probability distribution	0 (s)	14.8 (s)

5 Conclusion

In this paper, we propose a new efficient interference calculation model, which can calculate the occurrence probability of the initial satellite and deduce the distribution of interfering satellites in the visual area through the initial satellite, then the model can calculate the aggregate interference from the interfering constellation to the GSO system fast. Compared with the traditional interference calculation model based on time-slice, the proposed model shows a great improvement in reducing model complexity, improving model calculation efficiency, lessening simulation equipment performance requirements and being suitable for a variety of large scale constellation configurations. By analyzing the experimental results, it can be seen that the calculation result distribution of the proposed model is consistent with that of the traditional model, and is not limited to the total simulation duration, which can include the worst co-line interference scenario, and it can better reflect the impact of extreme simulation scenarios on GSO system. What's more, in the same simulation scenario, the time-consuming of the proposed model is at least 518 times faster than that of the traditional model. With the multiplication of the number of interference constellations, the simulation duration of the traditional time-slice model increases by 15.8 times, while the simulation duration of the proposed model changes slightly.

References

1. Liu, J., Wei, Z., Zhao, B., Su, J., Xin, Q.: A probabilistic resilient routing scheme for low-earth-orbit satellite constellations. In: Liu, Z., Wu, F., Das, S.K. (eds.) WASA 2021. LNCS, vol. 12939, pp. 254–261. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86137-7_28
2. Zhou, D., Sheng, M., Luo, J., Liu, R., Li, J., Han, Z.: Collaborative data scheduling with joint forward and backward induction in small satellite networks. IEEE Trans. Commun. **67**(5), 3443–3456 (2019)
3. Cao, X., Yang, P., Alzenad, M., Xi, X., Wu, D., Yanikomeroglu, H.: Airborne communication networks: a survey. IEEE J. Sel. Areas Commun. **36**(9), 1907–1926 (2018)
4. Tang, Z., et al.: A quasi-dynamic inter-satellite link reassignment method for LEO satellite networks. In: Xu, K., Zhu, H. (eds.) WASA 2015. LNCS, vol. 9204, pp. 497–507. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21837-3_49
5. Kuang, L., Jiang, C., Qian, Y., Lu, J.: Terrestrial-Satellite Communication Networks. WN, Springer, Cham (2018). https://doi.org/10.1007/978-3-319-61768-8
6. Radio regulations articles. International Telecommunication Union (2016)
7. Protection of GSO networks by NGSO systems[EB/OL]. U.S Federal Communications Commission (2016)
8. Del Portillo, I., Cameron, B.G., Crawley, E.F.: A technical comparison of three low earth orbit satellite constellation systems to provide global broadband. Acta Astronaut. **159**, 123–135 (2019)
9. Interference analysis to accompany the request for modification of the steam-2b non-geostationary satellite system. CRc4422M3 interference analysis with respect to compliance with RoP, no. 9.27 STEAM-2B MOD report, p. 29

10. Braun, C., Voicu, A.M., Simić, L., Mähönen, P.: Should we worry about interference in emerging dense NGSO satellite constellations? In: 2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), pp. 1–10. IEEE (2019)
11. Yastrebova, A., et al.: Theoretical and simulation-based analysis of terrestrial interference to LEO satellite uplinks. In: GLOBECOM 2020 IEEE Global Communications Conference, pp. 1–6. IEEE (2020)
12. Okati, N., Riihonen, T., Korpi, D., Angervuori, I., Wichman, R.: Downlink coverage and rate analysis of low earth orbit satellite constellations using stochastic geometry. IEEE Trans. Commun. **68**(8), 5120–5134 (2020)
13. Fortes, J., Sampaio-Neto, R., Maldonado, J.: An analytical method for assessing interference in an environment involving NGSO satellite networks. Int. J. Satell. Commun. Netw. **17**(6), 5–9 (1999)
14. Fortes, J., Sampaio-Neto, R., Goicochea, J.: Fast computation of interference statistics in multiple non-GSO satellite systems environments using the analytical method. IEE Proc. Commun. **151**(1), 44–49 (2006)
15. Jin, J., Lin, Z., Yan, J., Kuang, L.: Method for calculating probability distribution of equivalent power flux density. J. Tsinghua Univ. Nat. Sci. Ed. **62**(1), 7 (2022)
16. Analytical method for determining the statistics of interference between non geostationary satellite orbit fixed-satellite service systems and other non geostationary satellite orbit fixed satellite service systems or geostationary satellite orbit fixed satellite service networks. International Telecommunication Union, ITU-R S.1529, Gevena (2001)
17. Simulation methodologies for determining statistics of short-term interference between co-frequency, codirectional non-geostationary-satellite orbit fixed-satellite service systems in circular orbits and other non-geostationary fixed-satellite service systems in circular orbits or geostationary-satellite orbit fixed-satellite service networks. International Telecommunication Union, ITU-R S. 1325-3, Gevena (2003)
18. Song, Z., Dai, G., Wang, M.: Theoretical analysis of walker constellation coverage to area target. Comput. Eng. Des. **35**(10), 3639–3644 (2014)
19. Satellite antenna radiation patterns for non-geostationary orbit satellite antennas operating in the fixed-satellite service below 30 GHz. International Telecommunication Union, ITU-R S. 1528-1, Gevena (2001)
20. Reference FSS earth-station radiation patterns for use in interference assessment involving non-GSO satellites in frequency bands between 10.7 GHz and 30 GHz. International Telecommunication Union, ITU-R S. 1428-1, Gevena (2001)

FedALP: An Adaptive Layer-Based Approach for Improved Personalized Federated Learning

Zaipeng Xie[1,2](✉), Yao Liu[2], Zhihao Qu[2], Bin Tang[2], and Weiyi Zhao[3]

[1] State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering,
Hohai University, Nanjing, China
[2] College of Computer and Information, Hohai University, Nanjing, China
{zaipengxie,lyao,quzhihao,cstb}@hhu.edu.cn
[3] The University of Hong Kong, Hong Kong, China
wyzhao99@connect.hku.hk

Abstract. Personalized federated learning (PFL) is an improved framework that can facilitate the handling of data heterogeneity by learning personalized models. As personalization performance directly depends on the global model, it is desired to acquire a global model with a decent generalization capability under data heterogeneity. This paper proposes a novel PFL scheme, FedALP, integrating the clustering method with an adaptive layer-based fusion algorithm. Experiments are performed using various neural network models on three standard datasets. Experimental results demonstrate that, compared with the FedAvg method, our scheme can significantly improve the local model's performance with a negligible decrease in the generalization capability of the global model. Furthermore, our scheme is customizable for specific PFL applications; hence it may provide a flexible strategy to effectuate a balanced performance for both the global and the local models.

Keywords: Personalized federated learning · Adaptive · Layer-based · Non-IID

1 Introduction

Federated Learning (FL) is a distributed deep learning framework [11] that allows multiple clients to jointly train a shared global model under the coordination of a central server while keeping the participants' data private. Most of the existing training methods are variants of the Federated Averaging (FedAvg) introduced by McMahan et al. [14]. However, in the presence of statistical data

Supported by The Belt and Road Special Foundation of the State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering under Grant 2021490811, the National Natural Science Foundation of China under Grant No. 61872171, and the Fundamental Research Funds for the Central Universities No. B210201053.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 80–92, 2022.
https://doi.org/10.1007/978-3-031-19214-2_7

heterogeneity [20], such as non-IID and imbalanced data, it is difficult for FL to train a single model that works well for all clients. Optimizing the global model independently may result in poor performance in the local models [8,20].

Personalized federated learning (PFL) [16] has been proposed as a solution to mitigate the aforementioned issues. Many efforts [13] have been made to explore a scheme that exhibits sound global generalization properties and well personalized local matching properties. Wu et al. [17] proposed a tailored hierarchical communication architecture that introduced an intermediate layer of servers between the cloud and the clients for asynchronous training. Arivazhagan et al. [1] proposed a neural network architecture where the base layer is trained on a centralized server using FedAvg, and the top layer is trained locally using a gradient descent variant. Liang et al. [9] proposed a new FL algorithm that learns a compact local representation and a global model across all clients. However, these personalization methods are usually focused on enhancing local representations, and the generalization capability of the global model is of less concern.

In the PFL process, many clients may share some similarities in the data distribution. If these clients can be aggregated for mutual benefits, the performance may outperform localized adaptation schemes [10]. Briggs et al. [2] proposed a hierarchical clustering strategy to separate client clusters by comparing their local updates with the global model. Ma et al. [12] proposed a personalized FL method that incorporates attention-based clustering to facilitate collaborations among similar clients. Zhang et al. [19] proposed a PFL framework that can calculate optimal weighted model combinations for each client. Huang et al. [5] proposed a attentive message passing mechanism that can assist the collaboration among clients significantly. Instead of maintaining a single global model, this mechanism retains a personalized cloud model for individual client.

Sattler et al. [15] proposed clustered federated learning paradigm that exploits geometric properties of the FL loss surface to group the clients into clusters. However, these approaches do not consider the relationship between the global model and the personalized local models. The generalization performance of the global model can be impaired because of the limited communication between client groups. Wu et al. [17] pointed out that the comprehensive knowledge from the global model may be beneficial in situations when limited local data is acquired for training. On the other hand, the global model with a decent generalization performance can serve as an unbiased initialization for new users. Hence, it is desirable to explore a novel PFL training framework that ensures adequate performance for the global model.

This paper proposes a hierarchical PFL framework FedALP named *federated learning with adaptive layer-based personalization*. The focus of this framework is on addressing the above issues. Our contributions can be summarized as follows:

- We propose a novel federated learning framework, which integrates a client clustering method and an adaptive layer-based fusion algorithm. This framework does not require manual efforts, and it can adaptively allocate layers for

the personalization model to maintain a decent performance for both local and global models.
- Our proposed scheme is fully customizable for specific PFL applications; hence it can provide a flexible strategy to effectuate a balanced performance for both the global and the local models.
- Experiments have been performed on models with datasets including MNIST, FashionMNIST, and CIFAR-10. The results demonstrate that FedALP can improve the performance of the local model by maximum 31.5% with at most 6.8% decrease in the performance of the global model. FedALP on non-IID data can achieve a comparable or even better performance than the FedAvg framework on IID data for the same dataset. And as β varies, FedALP can provide a dynamic performance between an optimized global and personalized local performance.

2 Methodology

2.1 Motivation

While FL has been shown to be effective in training a single accurate global inference model, it may not generate a satisfactory global model shared by all nodes on non-IID dataset. In recent years, in the explorations of PFL, many researchers have focused on two possible solutions:

1) **Clustering-based personalization** [2,15,19]. Instead of expecting the global model to perform well on all clients, this method trains dedicated models for sharing within a group of clients with similar data distribution.
2) **Layer-based personalization** [1,3,9]. These method personalizes some layers of the local model, while the rest are derived from the global model.

However, current clustering-based personalization approaches rarely focus on model sharing between groups. Consequently, they may compromise the generalization performance of the global model. Meanwhile, current layer-based personalization approaches lack flexibility and adaptability because they usually adopt predefined layering. Therefore, they may end up with a suboptimal solution, leading to an unbalanced performance for both the global and the local models.

The proposed scheme, FedALP, employs an adaptive layer-based PFL scheme that incorporates a clustering method. In this method, the layer-based personalization scheme is applied to a group of clients. Each client can return the performance feedback within the group to regulate the layer-based personalization training.

2.2 Algorithm Design

Algorithm 1 describes the proposed scheme of FedALP, and a summary of the symbols is listed in Table 1. In general, FedALP's workflow consists of three phases. The first phase is the warm-up phase, where the FedALP initiate the global model on the global server and push it to every participating client. The training process at this phase follows the FedAvg [14] scheme and runs for T_{pre} rounds. The second phase performs the clustering and the layer-wise personalization based on the results from the warm-up phase. The third phase is the main body of FedALP when the groups and the layers are set. In this phase, the hierarchical PFL training is carried out for $(T - T_{pre})$ rounds until the models achieve satisfactory performance results. Details of the process of FedALP are elaborated as follows:

Table 1. Notation

Symbol	Explanation	Symbol	Explanation
T	Iteration number of overall training	$w_g^{(t)}$	Global model parameters of iteration t
T_{pre}	Iteration number of the warm-up phase	$w_m^{(t)}$	The m^{th} group's model parameters of iteration t
K	Number of clients	$\mathcal{G}_m$	Set of client index in m^{th} group
M	Number of groups	$\mathcal{D}_m$	Average gradient of clients within m^{th} group
ρ	Cosine similarity matrix, $\rho \subset \mathbb{R}^{K \times K}$	Ψ_m	Personalization weight of m^{th} group
l	Number of model layers involved in training	α	Dirichlet distribution parameters, $\alpha \in [0, +\infty)$
γ_k	Samples number weight of client k	β	Personalization factor, $\beta \in [0, 1]$

Warm-up Phase: At the beginning of FedALP, each client will initialize a model that is trained and shared at a given frequency (every 20 epochs in our setting). Meanwhile, the Global server receives clients' gradients to update the global model; then it pushes the latest global model to the clients. The process at the current phase is the same as the standard FedAvg's setting; the model training at iteration $t + 1$ will only begin after successfully receiving $w^{(t)}$. We train the global model for T_{pre} iterations, where T_{pre} is a predefined setting which is typically set to be 40% to 70% of the overall training iterations T. At this phase, the global objective function of FedAvg is given by

$$\underset{w}{min} \left\{ f(w) \triangleq \sum_{k=1}^{K} \gamma_k F_k(w) \right\}, \tag{1}$$

where K is the number of clients, γ_k is the weight of the k-th client, $\gamma_k \geq 0$, $\sum_k \gamma_k = 1$, and $F_k(w)$ is the local objective functions. The local objective functions is given by

$$F_k(w) \triangleq \mathbb{E}_{(x,y) \sim p_{data}^k} L(x, y; w), \tag{2}$$

where $p_{data}^{(k)}$ is the data distribution of client k, $L(\cdot)$ is the loss function of the predictions on examples (x, y) made with model parameters w. A global model, $w_g^{(T_{pre})}$, can be obtained after T_{pre} rounds and is shared by each client.

Layer-wise Personalization with Clustering: This phase starts with a global model $w_g^{(T_{pre})}$ where its gradient updates $\{\Delta w_k^{(T_{pre})}\}_{k=1}^K$ are noted as ΔW. Algorithm 2 describes the process of this phase. A pairwise cosine similarity matrix $\rho \subset \mathbb{R}^{K\times K}$ is constructed with cosine similarity kernel S as follows:

$$\rho = S(\Delta W),\, \rho_{ij} = S_C(i,j), \tag{3}$$

where the cosine similarity $S_C(\cdot,\cdot)$ between the gradient updates of any two clients i and j is defined by:

$$S_C(i,j) \triangleq \frac{< \Delta w_i^{(T_{pre})}, \Delta w_j^{(T_{pre})} >}{||\Delta w_i^{(T_{pre})}||\,||\Delta w_j^{(T_{pre})}||}, \tag{4}$$

where $i, j \in \{1, 2, \cdots, K\}$. Then we use a top-down hierarchical clustering algorithm [4] to cluster K clients into M groups based on ρ, and thus produce a group list, denoted by $\{\mathcal{G}_m\}_{m=1}^M$. A single process is designated as the group server for coordinating among clients within the group. The group server maintains a group model w_m, while the global model is denoted as w_g.

Algorithm 1: FL with Adaptive Layer-based Personalization (FedALP)

```
Procedure FedALP SERVER TRAINING:
    Input: Round number $T_{pre}$, $T$, local epochs $E$, batch size $B$, learning rate $\eta$
    Output: Global model $w_g^{(T)}$ and group models $\{w_m^{(T)}\}_{m=1}^M$
    Get $w_g^{(T_{pre})}$ by FedAvg [14] with $E$, $B$, $\eta$, $T_{pre}$
    Execute FEDALP INITIALIZATION (Algorithm 2)
    Initialize group model $\{w_m^{(T_{pre})}\}_{m=1}^M$ with $w_m^{(T_{pre})} \leftarrow w_g^{(T_{pre})}$
    for each global round $t = T_{pre+1}, T_{pre+2}, \cdots, T$ do
        for $m = 1, 2, \cdots, M$ do
            $\mathcal{W}_{new}$=MixByLayer($w_m^{(t-1)}, w_g^{(t)}, \Psi_m$)
            Server broadcasts $\mathcal{W}_{new}$ to client $k \in \mathcal{G}_m$
            for each client $k \in \mathcal{G}_m$ do
                $\Delta w_k^{(t+1)} \leftarrow$ClientUpdate($k, \mathcal{W}_{new}$)
            $w_m^{(t+1)} \leftarrow w_m^{(t)} + \sum_{k\in\mathcal{G}_m} \gamma_k \Delta w_k^{(t+1)}$
        $w_g^{(t+1)} \leftarrow \sum_{m=1}^M \left\{ (\sum_{k\in\mathcal{G}_m} \gamma_k) w_m^{(t+1)} \right\}$
Function MixByLayer($w_m, w_g, \Psi_m$):
    for each model layer $l = 1, 2, \cdots, L$ do
        $w^{(l)} \leftarrow \Psi_m^{(l)} w_m^{(l)} + (1 - \Psi_m^{(l)}) w_g^{(l)}$
    return $w \leftarrow \{w^{(l)}\}_{l=1}^L$
Function ClientUpdate($i, w$):
    $\hat{w} \leftarrow w$
    for each local epoch $e = 1, 2, \cdots, E$ do
        $w \leftarrow w - \eta \cdot \nabla L(b; w)$ for local batch $b \in B_i$
    return $\Delta w \leftarrow w - \hat{w}$
```

Next, the group server adopts the proposed adpative layer-based fusion algorithm to generate a layer-wise weight list $\varPsi$. The procedure of getting $\varPsi$ is given as follows: Given a group $\mathcal{G}_m$, we first calculate the average gradient updates $\mathcal{D}_m$ within each group as given by:

$$\mathcal{D}_m = \sum_{k \in \mathcal{G}_m} \gamma_k \varDelta w^{(k)}. \tag{5}$$

Then the updates can be divided into individual sets of layers as given by

$$\mathcal{D}_m = \left[\mathcal{D}_m^{(1)}, \mathcal{D}_m^{(2)}, \cdots, \mathcal{D}_m^{(l)}\right], \tag{6}$$

where l represents the total number of model layers involved in the training.

We define a tensor δ_m to represent the Euclidean distance of each layer in the model within m^{th} group. δ_m can be derived from $\mathcal{D}_m$ as given by

$$\delta_m \leftarrow \left\{||\mathcal{D}_m^{(1)}||_2, ||\mathcal{D}_m^{(2)}||_2, \cdots, ||\mathcal{D}_m^{(l)}||_2\right\}, \tag{7}$$

where the Euclidean norm, $||\mathcal{D}_m^{(n)}||_2$, represents the update distance of the n^{th} layer and $n \in \{1, \cdots, l\}$. It is worth noting that $||\mathcal{D}_m^{(n)}||_2$ is proportional to the degree of the personalization for the layer.

We define a personalization factor β and then the layer-based personalization weights $\varPsi_m$ is calculated as given by

$$\varPsi_m = \beta \cdot \delta_m / \mathbf{max}(\delta_m). \tag{8}$$

The personalization factor β is a parameter that can have an impact on the personalization degree of FedALP. When $\beta = 0$, FedALP turns into FedAvg; when $\beta = 1$, some layers are completely localized at the expense of the generalization capability of the global model.

FedALP Hierarchical PFL Training: In this phase, the group server takes over the global sever as the organizer within each group, where clients' gradients are sent to update the group model w_m, and the latest group model is sent back to the clients. While the global server only communicates with the group servers. The global model w_g is updated by averaging the w_m at every global iteration. Figure 1 describes current phase of FedALP.

At the beginning of each iteration, the global server sends the latest global model w_g to each group server. Then, the global model is weighted and fused with the group model w_m layer-by-layer. The model parameter of the n^{th} layer at group m is given by

$$\mathcal{W}_{new}^{(n)} = \varPsi_m^{(n)} w_m^{(n)} + (1 - \varPsi_m^{(n)}) w_g^{(n)}, \tag{9}$$

process.pdf

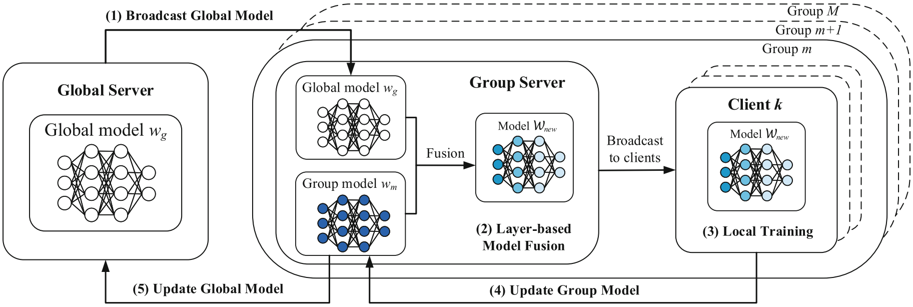


Fig. 1. FedALP training process

where $\mathcal{W}_{new}^{(n)}$ represents the n-th layer of $\mathcal{W}_{new}$ and $n \in \{1, \cdots, l\}$. For each group, the model $\mathcal{W}_{new}$ has l layers and is given by

$$\mathcal{W}_{new} \leftarrow \left\{\mathcal{W}_{new}^{(1)}, \mathcal{W}_{new}^{(2)}, \cdots, \mathcal{W}_{new}^{(l)}\right\}. \tag{10}$$

Here $\mathcal{W}_{new}$ serves as the starting point for the next iteration and is broadcasted within the group. The client trains the model $\mathcal{W}_{new}$ for several epochs (we pick 20 as our setting) and they are aggregated to update the group model w_m.

The training process repeats until the desired number of iterations or the accuracy reaches a given threshold. Thus it concludes the FedALP process. In our method, all model aggregations are weighted based on the amount of data owned by the client to optimize the model performance further.

Algorithm 2: Layer-wise Personalization Algorithm with Clustering

```
Procedure LAYER-WISE PERSONALIZATION WITH CLUSTERING
    Input: Group number $M$, personalization factor $\beta$, gradients $\{\Delta w_k^{(T_{pre})}\}_{k=1}^K$
    Output: $\{\mathcal{G}_m\}_{m=1}^M$ and $\{\Psi_m\}_{m=1}^M$
    Estimated hierarchical clustering $P$ with Ward method from the similarity
      matrix $\rho$, where $\rho_{i,j} = S_C(\Delta w_i^{(T_{pre})}, \Delta w_j^{(T_{pre})}), i, j \in \{k\}_{k=1}^K$ (Eq. 4)
    Intersect $P$ to determine $M$ groups $\{\mathcal{G}_m\}_{m=1}^M$, $\mathcal{G}_m = \{k \mid \textit{client k in group m}\}$
    for $m = 1, 2, \cdots, M$ do
        $\mathcal{D}_m \leftarrow \sum_{k \in \mathcal{G}_m} \gamma_k \Delta w_k^{(T_{pre})}$
        $\Psi_m \leftarrow$ LayersWeight$(\mathcal{D}_m, \beta)$
Function LayersWeight($\mathcal{D}, \beta$):
    $\mathcal{D} = [\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \cdots, \mathcal{D}^{(L)}]$
    for each model layer $l = 1, 2, \cdots, L$ do
        $\delta^{(l)} \leftarrow ||\mathcal{D}^{(l)}||_2$
    $\delta \leftarrow \{\delta^{(1)}, \delta^{(2)}, \cdots, \delta^{(L)}\}$
    return $\Psi \leftarrow \beta \cdot \delta / \mathbf{max}(\delta)$
```

In summary, the proposed method can adaptively get the optimized layer-based personalization for various models. Compared to personalizing the entire model, our layer-based personalization can improve the performance of the local model with a minimal impact on the global model generalization performance. While implementing personalization, we also optimize the global model. Hence, the global model facilitates communication among groups, and every client may obtain knowledge from the global model and avoid overfitting and a locally optimal result. The global model provides a generalization capability for other applications that may exploit its ability. It is worth mentioning that, since our layer-wise algorithm is personalized, each group can have its own layer-based weights Ψ_m, which will allow using different personalization within different groups.

Our approach is flexible compared with some state-of-the-art layer-based personalization schemes [1,3,9]. This is because the proposed layer-wise algorithm is adaptable by incorporating the personalization factor, β. A balanced performance can be achieved for both the global and the local models that are tailored for specific PFL applications. For example, when $\beta = 0$, FedALP turns into FedAvg; when $\beta = 1$, FedALP becomes a variant of [1], some layers are completely localized at the expense of the generalization capability of the global model.

3 Experiments

3.1 Datasets and Model Architectures

We evaluated the performance of FedALP with four models on three non-IID datasets based on MNIST, FashionMNIST, CIFAR-10. It is worth noting that, various kinds of non-IID data partition scheme exist and our data partition is the same as in [4].

A. **MNIST** [7]. We generated a non-IID dataset consisting of 100 clients, where each client has 500 training samples and 100 test samples that consist of only one digit. Each digit is owned by 10 clients.
B. **FashionMNIST** [18]. We follow the same procedure as MNIST to create a non-IID dataset using FashionMNIST.
C. **CIFAR-10** [6]. We partition the CIFAR-10 dataset using the Dirichlet distribution, **DIR**(α), to provide the corresponding cross-category partition for each client. The parameter α controls the heterogeneity of the generated dataset. When $\alpha = 0$, it means that each client gets only one category of sample, and when $\alpha \to +\infty$, it means that all categories of sample are uniformly distributed on each client. We consider 100 clients and assign datasets to clients under the IID and Dirichlet distribution with $\alpha \in \{0.001, 0.01, 0.1\}$. Clients will have unbalanced amount of samples.

The experiments utilized four models to evaluation the FedALP scheme. The first model is a fully connected network with only one hidden layer, named MNIST-NN. The second model is a CNN named MNIST-CNN. It consists of 3 convolutional layers and 2 fully connected layers. The third one is also a CNN named CIFAR10-CNN and it consists of 3 convolutional layers and 2 fully connected layers. The fourth one is an AlexNet model accustomed to the CIFAR-10 dataset, and it consists of 5 convolutional layers and 3 fully connected layers. Both MNIST-NN and MNIST-CNN were used on the MNIST and FashionMNIST datasets, while both CIFAR10-CNN and AlexNet were used on CIFAR-10.

3.2 FedALP Evaluation

In our experiments, the FedAvg algorithm [14] is used as the baseline for training on both the IID dataset and the non-IID dataset. In each experiment, the global model maintained by the FedALP algorithm is named *FedALP_global*. Three sets of experiments have been performed.

1. Experiment 1 describes the comparison of the accuracy performance between FedAvg and FedALP on both IID and non-IID datasets.
2. Experiment 2 compares the model's accuracy performance of FedAvg and FedALP on both IID and non-IID CIFAR-10 by varying α, the degree of non-IID in datasets.
3. Experiment 3 compares the model's accuracy performance of FedALP on non-IID CIFAR-10 by varying β, the personalization factor of FedALP.

Experiment 1: Fig. 2 describes the experimental results and the accuracy values are listed in Table 2. This experiment evaluates the accuracy performance of all three datasets in four different cases: (1) FedAvg on IID datasets (green line), (2) FedAvg on non-IID datasets (orange line), (3) the **global model** performance

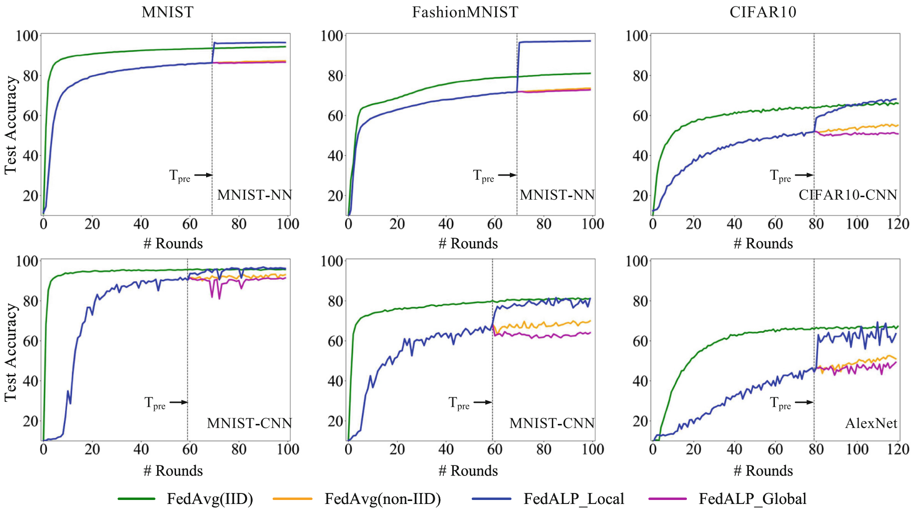


Fig. 2. Comparison of FedAvg on IID, FedAvg on non-IID, and FedALP on non-IID datasets ($\alpha = 0.001$, $\beta = 0.6$) with various models. (Color figure online)

Table 2. The accuracy comparison of FedAvg and FedALP with various models.

Dataset	Model	Accuracy				Ratio
		Non-IID ($\alpha = 0.001$)			IID	
		FedAvg	FedALP_global	**FedALP**	FedAvg	
MNIST	MNIST-NN	87.25	(↓ 0.69%)86.65	(↑ 10.56%)94.46	94.37	102.21%
	MNIST-CNN	92.55	(↓ 0.75%)91.86	(↑ 4.45%)96.67	96.04	100.66%
FashionMnist	MNIST-NN	73.48	(↓ 0.91%)72.81	(↑ 31.52%)96.64	81.05	119.24%
	MNIST-CNN	68.86	(↓ 5.63%)65.01	(↑ 15.65%)79.67	80.99	98.37%
CIFAR10	CIFAR10-CNN	54.86	(↓ 6.33%)51.39	(↑ 23.57%)67.79	65.82	102.99%
	CIFAR10-AlexNet	51.73	(↓ 6.79%)48.22	(↑ 22.83%)63.54	66.82	91.60%

of FedALP on non-IID datasets, $\alpha = 0.001$ (magenta line), (4) the average **local model** performance of FedALP on non-IID datasets, $\alpha = 0.001$ (blue line).

We observe that FedAvg on non-IID data significantly decreases its accuracy performance compared with FedAvg on IID data. Since the starting point of the FedALP is set to be at iteration T_{pre}, a notable performance enhancement is demonstrated, as shown in Fig. 2. We found that our FedALP outperforms FedAvg on non-IID by maximum 31.5% in the average local model performance, while a slight decrease (maximum 6.8%) is observed in the global model performance. The adaptively layer-based fusion method can accommodate some layers by adjusting their personalization contribution, preventing the overall model from deviating too far from the global model. Interestingly, our FedALP method on non-IID datasets and the FedAvg method on IID datasets are comparable in accuracy performance. An intuitive explanation is that since our approach can adaptively adjust the personalized layering scheme for each group, it may boost the accuracy performance even with data discrepancy.

Experiment 2: Experimental results is summarized in Fig. 3 and the accuracy values are listed in Table 3. This experiment evaluates the test accuracy performance of CIFAR-10 by setting α to three different values $\{0.001, 0.01, 0.1\}$. Results are collected for four cases: (1) FedAvg on IID datasets (green line), (2) FedAvg on non-IID datasets (orange line), (3) the **global model** performance of FedALP on non-IID datasets (magenta line), (4) the average **local model** performance of FedALP on non-IID datasets (blue line).

We observe that in all cases with non-IID datasets, our FedALP outperforms the FedAvg methods. In addition, we observe that our FedALP approach demonstrates excellent effectiveness in the accuracy performance as α decreases. This is because α is a parameter that determines the degree of non-IID, and the reduction in α will produce a performance degradation on FedAvg. At the same time, our FedALP method can mitigate the data discrepancy and boost performance.

Experiment 3: In this experiment, we evaluate the accuracy of both the global and local models by varying the personalization factor, β, as shown in Fig. 4. Two models, including CIFAR10-CNN and AlexNet, are employed on the non-IID CIFAR-10 datasets ($\alpha = 0.001$). This experiment aims to demonstrate the dynamic performance of FedALP that may be regulated for a balanced global and personalized local performance.

In Fig. 4, we compare the accuracy versus rounds by varying β from 0 to 1 with 0.3 as the step. It is worth noting that the solid lines describe the average result for the local models, and the dash-dotted lines illustrate the results for the global model.

We observe that the average local model's accuracy improves significantly as β increases-i.e., the degree of personalization of each layer increases, resulting in an improved local model. Meanwhile, the global model's accuracy degrades slightly. When $\beta = 0$, our proposed scheme produces the exact results as FedAvg. This is because all the layers contribute to the training of the global model. When $\beta = 1$, the personalization layers do not contribute to the training of the global model; hence, a maximum local model accuracy can be attained with a 12% decrease in the global accuracy performance compared with FedAvg.

In conclusion, by carefully choosing β, our scheme can significantly improve the local model's performance with a negligible decrease in the global model's accuracy. Our proposed method can adaptively accommodate specific PFL applications, providing flexibility to produce a balanced performance for both the global and the local models.

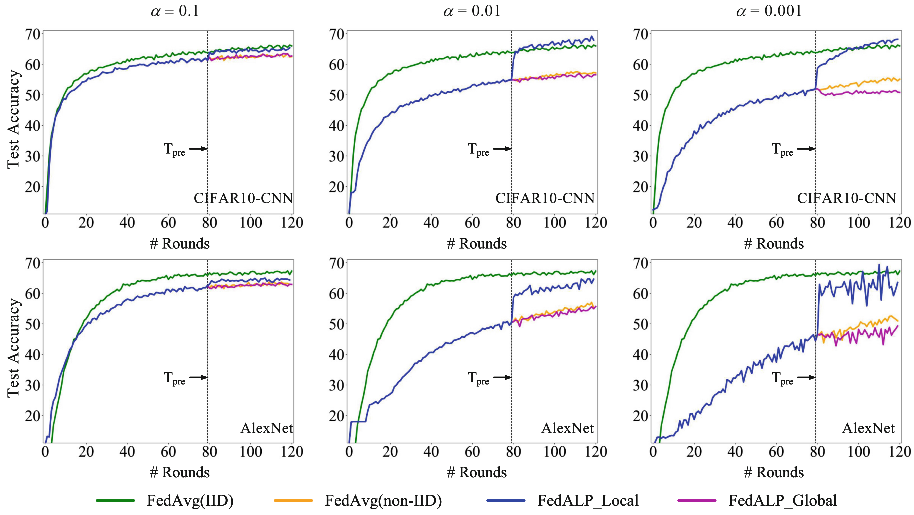


Fig. 3. Comparison of FedAvg on IID and non-IID data, FedALP on non-IID CIFAR-10 datasets with two models by varying $\alpha \in \{0.001, 0.01, 0.1\}$. (Color figure online)

Table 3. The accuracy comparison of FedAvg and FedALP by varying α.

Dataset-model	α	Accuracy		
		FedAvg	FedALP_global	FedALP
CIFAR10-CNN	0.1	62.81	(↑ 1.11%)63.51	(↑ 3.30%)64.88
	0.01	57.03	(↓ 0.79%)56.58	(↑ 19.48%)68.14
	0.001	54.86	(↓ 6.33%)51.39	(↑ 23.57%)67.79
CIFAR10-AlexNet	0.1	63.19	(↑ 0.43%)62.92	(↑ 1.88%)63.38
	0.01	56.06	(↓ 2.23%)54.81	(↑ 13.66%)63.72
	0.001	51.73	(↓ 6.79%)48.22	(↑ 22.83%)63.54

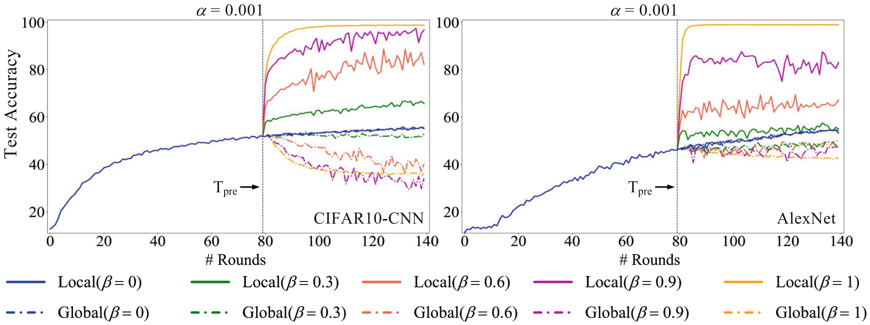


Fig. 4. FedALP on non-IID CIFAR-10 with various $\beta \in \{0, 0.3, 0.6, 0.9, 1\}$.

4 Conclusion

This study describes a novel personalization federated learning method that utilizes adaptive layer-based personalization and a clustering method. Experimental results show that the proposed method can significantly improve the local model's performance with a negligible decrease in the generalization capability of the global model. The training results on non-IID data with FedALP are comparable to a standard FedAvg on the IID data. Results also reveal that our scheme can provide a flexible strategy that effectuates a balanced performance for both the global and the local models for specific PFL applications.

References

1. Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S.: Federated learning with personalization layers. arXiv preprint arXiv:1912.00818 (2019)
2. Briggs, C., Fan, Z., Andras, P.: Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–9 (2020)
3. Bui, D., et al.: Federated user representation learning. arXiv preprint arXiv:1909.12535 (2019)

4. Fraboni, Y., Vidal, R., Kameni, L., Lorenzi, M.: Clustered sampling: low-variance and improved representativity for clients selection in federated learning. In: Proceedings of the 38th International Conference on Machine Learning, ICML, 18–24 July 2021, vol. 139, pp. 3407–3416 (2021)
5. Huang, Y., et al.: Personalized cross-silo federated learning on non-IID data. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI. pp. 7865–7873 (2021)
6. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Technical report, University of Toronto (2009)
7. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
8. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of FedAvg on non-IID data. In: 8th International Conference on Learning Representations (ICLR). arXiv preprint arXiv:1907.02189 (2020)
9. Liang, P.P., et al.: Think locally, act globally: federated learning with local and global representations. arXiv preprint arXiv:2001.01523 (2020)
10. Liu, B., Guo, Y., Chen, X.: PFA: privacy-preserving federated adaptation for effective model personalization. In: The Web Conference 2021, pp. 923–934 (2021)
11. Lo, S.K., Lu, Q., Wang, C., Paik, H., Zhu, L.: A systematic literature review on federated machine learning: from a software engineering perspective. ACM Comput. Surv. **54**(5), 95:1-95:39 (2021)
12. Ma, Z., Lu, Y., Li, W., Yi, J., Cui, S.: PFedAtt: attention-based personalized federated learning on heterogeneous clients. In: Asian Conference on Machine Learning, ACML, vol. 157, pp. 1253–1268 (2021)
13. Mansour, Y., Mohri, M., Ro, J., Suresh, A.T.: Three approaches for personalization with applications to federated learning. arXiv preprint arXiv:2002.10619 (2020)
14. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS, vol. 54, pp. 1273–1282 (2017)
15. Sattler, F., Müller, K., Samek, W.: Clustered federated learning: model-agnostic distributed multitask optimization under privacy constraints. IEEE Trans. Neural Netw. Learn. Syst. **32**(8), 3710–3722 (2021)
16. Tan, A.Z., Yu, H., Cui, L., Yang, Q.: Towards personalized federated learning. IEEE Transactions on Neural Networks and Learning Systems, pp. 1–17 (2022)
17. Wu, R., Scaglione, A., Wai, H., Karakoç, N., Hreinsson, K., Ma, W.: Federated block coordinate descent scheme for learning global and personalized models. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI (2021)
18. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747 (2017)
19. Zhang, M., Sapra, K., Fidler, S., Yeung, S., Alvarez, J.M.: Personalized federated learning with first order model optimization. In: 9th International Conference on Learning Representations (ICLR) (2021)
20. Zhu, H., Xu, J., Liu, S., Jin, Y.: Federated learning on non-IID data: a survey. Neurocomputing **465**, 371–390 (2021)

Recovering the Weights of Convolutional Neural Network via Chosen Pixel Horizontal Power Analysis

Sihan He[1], Weibin Wu[1], Yanbin Li[2], Lu Zhou[1], Liming Fang[1], and Zhe Liu[1](✉)

[1] Nanjing University of Aeronautics and Astronautics, Nanjing, China
{h_sihan,wuweibin,lu.zhou,fangliming,zhe.liu}@nuaa.edu.cn
[2] College of Artificial Intelligence, Nanjing Agricultural University, Nanjing, China
yanbinli@njau.edu.cn

Abstract. In many scenarios, people have a demand for deploying the artificial intelligence applications on the edge device of IoT. For some special applications, these embedded devices are always required realtime reponse; hence, it is necessary to process machine learning algorithms on microprocessors. However, these devices may be subjected to side-channel attacks (SCA). During the execution, these devices will generate the leakage information can be captured to get the secret data. In this work, we investigate how to reverse engineer the weights of a convolutional neural network (CNN) which is deployed on ARM Cortex-M3 using Chosen Pixel Horizontal Power Analysis (CP-HPA).

We conduct the experiment on ELMO emulating leaks for the ARM Cortex-M3. ARM Cortex-M3 microprocessors are often used to deploy CNNs. Here, we show that it is possible to recover the weights of a CNN using CP-HPA assuming that the adversary only has the knowledge of the architectures. We increase the accuracy of our attack through setting up chosen input pixel to correlate the selected multiplication. We are able to successfully recover the weights of a CMSIS-NN implementation CNN, and accuracy of our attack is 84.625%.

Keywords: SCA · CNN · ARM Cortex-M3 · CP-HPA

1 Introduction

With the continuous development of machine learning algorithms, artificial intelligence has gradually become mainstream across industries. Numerous examples prove its validity for several applications, such as image recognition [14], robotics [11], natural language processing [20], etc. Deep research in machine learning

This work is supported by the National Key R&D Program of China (Grant No. 2020AAA0107703), the National Natural Science Foundation of China (Grant No.62132008, 62072247, 62071222), the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20220075).

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 93–104, 2022.
https://doi.org/10.1007/978-3-031-19214-2_8

leads to more development of machine learning algorithms on edge devices of internet of things (IoT). Therefore, more neural network (NN) models have been implemented on low-cost micro-controllers, such as cortex core, or neural network hardware accelerators, such as FPGA.

The owners of NN models tend to spend plenty of time, fund, and human resource to collect and process data and consume a lot of computing power to train models. Hence, as the network is of commercial interest, its details are always kept secret. For some special applications, the NN models may contain private information which are sensitive to consumers. In these scenarios above, the NN models must not be disclosed. In recent years, many attack techniques for the neural network have been proposed. Side-channel attacks (SCA) are effective attack techniques for embedded systems that adversary can use information leakage, such as power consumption, electromagnetic emanations, timing, etc., generated during the execution to recover the secret data. For embedded neural network implementations, SCA is also a matter of concern.

There has been many work about SCA for attacking NN. Previous work in this field primarily focused on recovering model architectures or the information of inputs at the inference stage. Hua et al. [9] reversed engineered AlexNet and SqueezeNet deployed on CNN accelerator using memory and timing side-channel leakages generated by off-chip memory access patterns during dynamically zero pruning. The method of [9] relied on the memory access of adaptive zero pruning techniques that feature maps contain a large number of zeros after executing ReLU function, and the accelerator only reads and writes none-zero values. Thus, this attack assumed that the adversary possesses the knowledge of memory access patterns of the targeted accelerator. Batina et al. [2] completely recovered the NN architectures through electromagnetic side-channel using correlation power analysis (CPA), which is a special case of DPA, over the multiplication operations. In [2], they needed to correlate a large number of Waveform, which may bring adversary significant computation overheads. Batina et al. [3] used horizontal power analysis (HPA) to predict the input using electromagnetic emanations (EM) leakage by calculating the correlations from side-channel samples with Hamming weight of each multiplication result. Batina et al. [3] targeted the information of inputs and demonstrate the attacks on the MNIST dataset. The attacks for weights had not been discussed in [3]. Maji et al. [18] recovered the neural networks (floating point, fixed point, binary NNs) and the inputs (MNIST, CIFAR-10 and ImageNet) using timing/SPA attacks.

The main contributions of our work are as follows:

1) Our work recovers weights of the NN model which is CMSIS-NN implementation on ARM Cortex -M3 core processor using CP-HPA, and we present a comparison about requirement for trace number of respectively using CPA and CP-HPA to attack the weights of targeted CNN. Traces requirement of using CP-HPA to attack one weight is only one-twentieth that of CPA.
2) CP-HPA use a method of setting up special input values to correlate the selected multiplications to increase the accuracy of our attack. CP-HPA will be lead to a higher accuracy than HPA.

3) We reverse engineer the weights respectively using CPA, HPA, CP-HPA. Compared with CPA, CP-HPA have a lower traces requirement. In HPA experiment, we face a problem that the recovery of current weight is related to the previous weight; therefore, errors in the recovery of one weight leads to errors in the following series of weights. Hence, the accuracy of HPA is low. We solve the problem through CP-HPA, and this method leads to a high accuracy.

The paper is organized as follow: Sect. 2 introduces the theory behind side-channel attack and CNN. Then, in Sect. 3, we describe details about the targeted model and our attack method. At last, we describe our experimental results in Sect. 4 and conclude our work in Sect. 5.

2 Background

In this section, we introduce the concept of side-channel attack and several side attack methods used in this paper. Next, we give details about CMSIS-NN and CNN.

2.1 Side-Channel Attack

SCA is a commom attack method against embedded devices that adversary uses the energy consumption characteristic in the executing device to obtain the secret information of an algorithm. Based on existing analysis techniques, SCA has different variants. Simple Power Analysis (SPA) reveals the sensitive information using only a small amount of energy traces, utilizing the characteristics that equipment energy consumption depends on different operations. Correlation Power Analysis (CPA) [4] and Differential Power Analysis (DPA) [12] are advanced form of SCA. In addition to the two above, TA (template attack) [5] is a popular form of SCA as well. SCA and defense for it has been an important branch of cryptography study, and much SCA attacks for symmetric cryptographic algorithm (such as AES [17], DES [10]) and traditional public key cryptography algorithm (such as ECC [7], RSA [1]) has been proposed.

Correlation Power Attack. CPA uses many energy traces to analyze the consumption of the device for one operation and get the secret data by computing the correlation between candidates data and side-channel measurements using hamming weight model. It is assumed that CPA targets an operation f(m,s) of a known input m with a secret value s. The adversary calculates the hamming weight of f for predicted m with all the hypothesis values s. Then the attack computes ρ(hw,p) that ρ is the Pearson correlation coefficient, hw is the hamming weight for all the hypothesis s, and p is side-channel measurement. The correct value of s will lead to a higher correlation than other candidates, shown in Fig. 1.

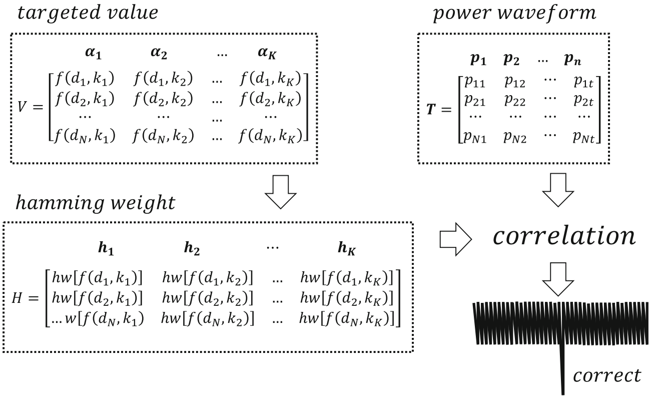

targeted value
α_1 α_2 ... α_K
$V = \begin{bmatrix} f(d_1,k_1) & f(d_1,k_2) & \ldots & f(d_1,k_K) \\ f(d_2,k_1) & f(d_2,k_2) & \ldots & f(d_2,k_K) \\ \ldots & \ldots & \ldots & \ldots \\ f(d_N,k_1) & f(d_N,k_2) & \ldots & f(d_N,k_K) \end{bmatrix}$
power waveform
p_1 p_2 ... p_n
$T = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1t} \\ p_{21} & p_{22} & \cdots & p_{2t} \\ \ldots & \ldots & \ldots & \ldots \\ p_{N1} & p_{N2} & \cdots & p_{Nt} \end{bmatrix}$
hamming weight
h_1 h_2 ... h_K
$H = \begin{bmatrix} hw[f(d_1,k_1)] & hw[f(d_1,k_2)] & \ldots & hw[f(d_1,k_K)] \\ hw[f(d_2,k_1)] & hw[f(d_2,k_2)] & \ldots & hw[f(d_2,k_K)] \\ \ldots w[f(d_N,k_1) & hw[f(d_N,k_2)] & \ldots & hw[f(d_N,k_K)] \end{bmatrix}$
correlation
correct

Fig. 1. Correlation power analysis

Horizontal Power Attack. HPA [6] is an improvement of CPA. As the secret s perform multiple f(m,s) with the different input m, one waveform can be split into multiple blocks according to f(m,s), shown in Fig. 2. We use these blocks to launch CPA, which is introduced above so that we can reverse engineer the secret value using a few side-channel measurements.

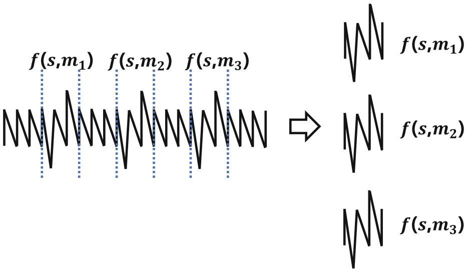


Fig. 2. One wareform consised of 4 f(m,s) aligned horizontally can be splited to 4 blocks which has one f(m,s).

2.2 Convolutional Neural Network and CMSIS-NN

Inspired by Hubel and Wiesel's research on biological processes of animals' visual cortex, CNN (Convolutional Neural Network) was proposed [16]. In recent years, CNN has been continuously developing in multiple directions and making breakthroughs in image recognition, natural language learning, motion analysis, etc. From the perspective of computation, CNN is not much different from ordinary networks, and they are made of many 2-dimensional layers, each of which consists of neurons. CNNs mainly use three types of layers: convolutional layers, pooling layers, and fully-connected layers.

In a convolutional layer, the convolution operation is performed on two matrices. A convolution layer extracts a new feature map by computing a dot product between the kernel and the input feature map, shown in Fig. 3. Pooling layers, which are always inserted between convolution layers, are used to reduce the feature dimensions in order to compress the images. Average pooling, which calculates the average value, and max pooling, which calculates the maximum value, are two common types of pooling layers, shown in Fig. 4. Fully-connected layer combines multiple feature maps after convolution and pooling and connects each neuron with its respective weight, shown in Fig. 5. Next, the results of the full-connected layer are given to the Softmax function for classification and CNN output final results.

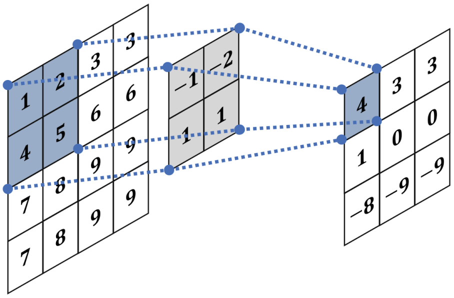


Fig. 3. On the left is input feature map, the convolution kernel is in the middle and the output feature map is on the right. The convolution kernel slides on the input feature map with fixed stride and performs dot product operation to obtain the output feature map.

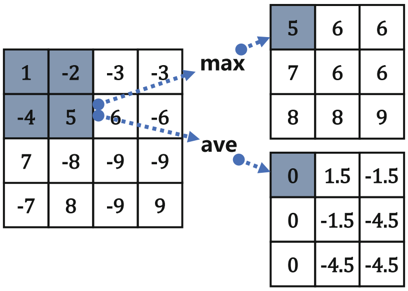


Fig. 4. Average pooling, which calculates the average value, and max pooling, which calculates the maximum value, are two commom types of pooling layers.

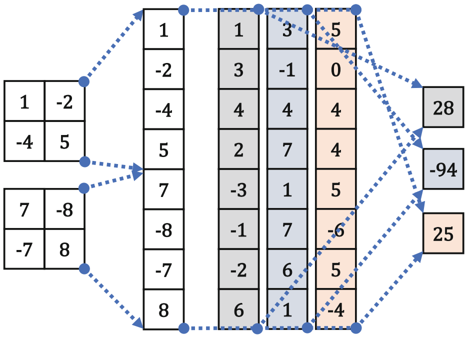


Fig. 5. In fully-connected layer, multiple feature maps are combined to a column matrix and each neuron connect with weights respectively.

Nowadays, a large number of edge devices of the IoT are put into use. Most of them adopt the cloud data processing method that collected data are uploaded to the cloud, processed on the server, and then returned to the microprocessor. This approach is not suitable for devices with real-time reponse requirements. Hence processing machine learning algorithms, especially CNN, on microprocessors has become a technological necessity. CMSIS-NN is one of the solutions. CMSIS-NN software library is a efficient kernels which is developed to apply machine learning applications to ARM Cortex-m series processors core. This library contains a number of functions, each covering a specific category: convolution functions, activation functions, fully-connected layer functions, pooling functions, SVDF layer functions, softmax functions, basic math functions, enable Arm Cortex-M processors core to implement neural network applications with maximum performance and minimize memory footprint. Additional information about CMSIS-NN can be acquired in [15].

3 Chosen Pixel Horizontal Power Analysis

In this section, we give details about the targeted model we use and propose the method about recovering the weights paraments of CNN through CP-HPA.

3.1 Targeted Model

Our targeted network is CNN, implemented by CMSIS-NN on ARM Cortex-M3 core processor. This CNN, totally 7 layers, consists of 3 convolutional layers: 3 max-pooling layers, and 1 fully-connected layer. The targeted CNN is applied to image classification, input is CIFAR-10 dataset [13] which consists of 60000 color images in classes.

For the neural network, the weights of each layer are of value. In this example, these weights are stored using 8-bits data(int8_t). In this CNN, weights computed in convolutional layer which call functions arm_convolve_HWC_q7_RGB() and arm_convolve_HWC_q7_fast(). These two methods are originated from CMSIS-NN both and used to implement convolution operation respectively for the first convolutional layer and the second and third convolutional layer. These two keep consistent in algorithm logic and implementation details, only have difference in data size of input parameter.

3.2 Correlating Selected Multiplications to Reverse the Weights of CNN

As storing of resulting multiplication is existed during the execution, targeting this temporary variable is feasible. In this example, as weights is 8-bits data, this hypothetical value of the parameter ranges from 0 to 256. For each hypothesis, the attack calculates the temporary variable using the weight w and the input and gets their correlation coefficient. The correct assumption has a higher correlation.

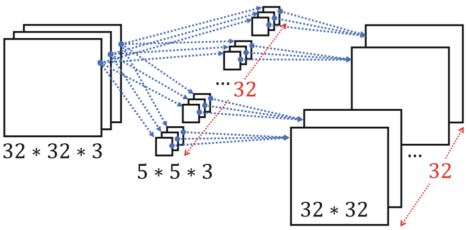


Fig. 6. In the first convolutional layer, 32 kenerls output 32 feature which is $32 * 32$ for a $32 * 32 * 3$ input feature as padding is 2 and stride is 1.

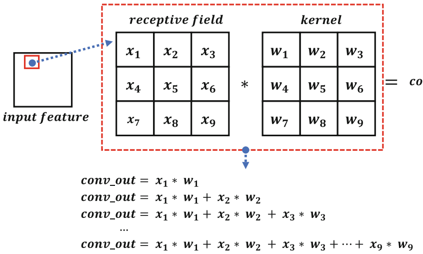

input feature
receptive field
kernel
x1 x2 x3
x4 x5 x6
x7 x8 x9
*
w1 w2 w3
w4 w5 w6
w7 w8 w9
= conv_out
conv_out = x1 * w1
conv_out = x1 * w1 + x2 * w2
conv_out = x1 * w1 + x2 * w2 + x3 * w3
…
conv_out = x1 * w1 + x2 * w2 + x3 * w3 + ⋯ + x9 * w9

Fig. 7. The output value of each multiplication result is related to the previous multiplication in a receptive field of the input feature map.

CPA targets only one multiplication on each trace. Hence, this method requires a large number of traces. Apart from the attack points and their adjacent field, the rest of the side-channel waveform is unused. In this example, The execution contain many multiplications in the convolutional layer. As one weight perform multiplications with different input respectively during the execution, multiple attack points can be found on one trace. Splitting one trace into multiple blocks according to multiplication and launching CPA using these blocks can reverse engineer weights of this CNN.

In the first convolutional layer of the targeted NN model, the size of the input feature is $32 * 32 * 3$; the output size is $32 * 32 * 32$, size of kernels is $5 * 5 * 3 * 32$. Each $5 * 5 * 3$ kernel and input feature performs a convolution operation to generate an output feature which is $32 * 32$, shown in Fig. 6. Each weight and respective input data perform a multiplication calculation as a kernel slides once. In our targeted NN model, each weight will perform such multiplication 784 times during the first convolutional layer. In our Experiment, we use 9 trace and correlate 784 multiplications for each trace, we found that the accuracy of attack is low in this way. As we correlate 784 multiplications to reverse Engineering kernels, recovering any weight incorrectly will lead to subsequent weights can not being recovered correctly as well. The output value of each multiplication result is related to the previous multiplication in a receptive field of the input feature map, shown in Fig. 7. If we incorrectly recover w_1, we will not get the correct result of the multiplication for w_2. Hence, we can not incorrectly recover w_2, the same is for w_3, and the mistake will continue until the last weight. For the whole reverse Engineering, methods with low fault tolerance are unacceptable. Therefore, we need to make some change in our attack, we set up special input values in order that we only trace chosen multiplication. We set only one non-zero value in a receptive field, and the rest are set at zero value so that kernel slides several times to generate a non-zero multiplication result for each weight. The purpose of this is to make each multiplication result we use only related to one targeted weight. Applying it to our targeted CNN, one non-zero value is set for every 75 bytes only in order that the kernel slides 5 steps to get a non-zero multiplication result that is related to the targeted weight only, shown in Fig. 8, and we can use 36 multiplication in the first convolutional layer totally. Then, we merge and correlate multiplications of more traces to recover each weight.

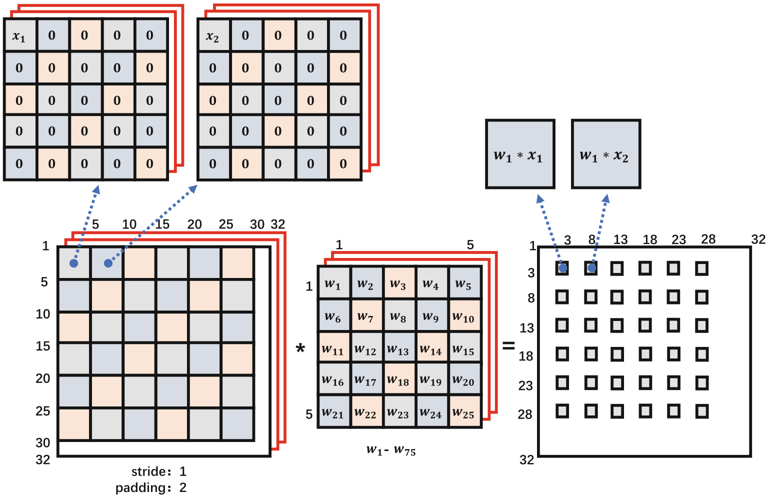

x_1
x_2
0
$w_1 * x_1$
$w_1 * x_2$
5 10 15 20 25 30 32
1 5 10 15 20 25 30 32
stride: 1
padding: 2
1 5
w_1 w_2 w_3 w_4 w_5
w_6 w_7 w_8 w_9 w_{10}
w_{11} w_{12} w_{13} w_{14} w_{15}
w_{16} w_{17} w_{18} w_{19} w_{20}
w_{21} w_{22} w_{23} w_{24} w_{25}
w_1- w_{75}
*
=
1 3 8 13 18 23 28 32
1 3 8 13 18 23 28 32

Fig. 8. As w_1 is targeted, one value which is respective to w_1 in a receptive field is set to non-zero for every 75 bytes only in order that kernel slides 5 times to get a non-zero multiplication result which is related to targeted weight only.

Launching CPA requires abundant consumption traces. As the SNR of measurements is low, the adversary needs more traces, sometimes millions. Not only numerous traces increase the acquisition time, but also processing masses of waveform requires more computation complexity. Hence, attacks for each weight involve time and computation overheads due to processing abundant measurements. However, even a small-scale network owns thousands of weights, the time required for recovery of all weights is overcharged. In HPA, by reasons of correlating multiple multiplications in each trace, the requirement of waveforms could be greatly reduced. However, the accuracy of HPA is low as discussed above. For using CP-HPA to reverse engineer the entire network, time, computation complexity, accuracy are all acceptable for adversary.

4 Experiments

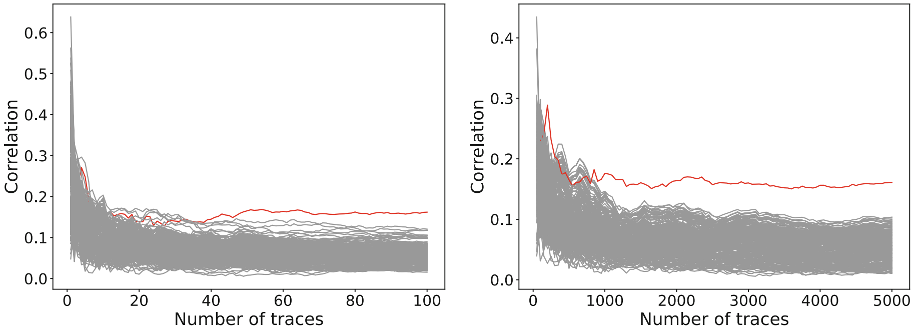
In the previous section, we discuss the threat model, which is CMSIS-NN implementation, and the purpose of the methodology to reverse engineer the weights of the targeted model. After that, we present a complexity analysis of our method. In this section, we conduct the experiment on emulating leaks for the ARM Cortex-M3.

We use the ELMO, which is a power trace simulator, to generate waveforms. ELMO is able to simulate power traces for any given Thumb binary, and its source code is available in [8]. ELMO can evaluate Leaks for the ARM Cortex-M0 and Cortex-M3 based on the Thumb instruction set. Additional information for the theory and development of the ELMO power model can be acquired in [19]. We simply remove the defines FIXEDVSRANDOM, MASKFLOW, ENERGYMODEL and define the coeffs_M3.txt as COEFFSFILE from the elmodefines.h file [8] in order to generate waveforms of the target model for ARM Cortex-M3. After that, we add gaussian noise on each trace to generate the final waveform which is used for our attack.

Table 1. Assembly for multiplication from input x and weight w: con_out $+= x * w$, the result is stored in a 32-bit register.

#	Instruction	Comment
237	bl starttrigger	trigger up
...	...	taking address for x
261	ldr r3, [r7,#12]	loading x
262	adds r3,r3,r2	loading x
263	ldrb r3,[r3]	loading x
264	lsls r3,r3,#24	loading x
265	asrs r3,r3,#24	loading x
266	move r0,r3	loading x
...	...	taking address for w
304	ldr r3, [r7,#4]	loading w
305	adds r3,r3,r2	loading w
306	ldrb r3,[r3]	loading w
307	lsls r3,r3,#24	loading w
308	asrs r3,r3,#24	loading w
309	muls r3,r0	multiplication of x and w
310	ldr r2, [r7,#24]	taking preious result
311	adds r3,r2,r3	accumulation
312	str r3,[r7,#24]	storing result
313	bl endtrigger	trigger down

For this experiment, we target the multiplication operation from the weight and the input. This result of the multiplication is stored in a temporary int variable (see in Table 1). Each measurement that is generated by ELMO is divided into many blocks for multiplication (see in the previous section). In our experiment, we use 100 traces respectively for each weight of the first convolutional layer of the targeted NN model, and we divide these traces into 3600 blocks totally. Each block has 121 points, and we correlate the field which ranges from point 104 to point 108. These five points in measurements reflect the power consumption of storing the multiplication result. In Fig. 9(a), around 40 traces will be required before the correct weights can be distinguished from other candidates. It can be seen that the correlation of correct weights and wrong hypothesis can be distinguished around 1000 traces, shown in Fig. 9(b). The targeted CNN has three convolutional layers; the attack for each convolutional layer are the same. The input of the next convolutional layer can be chosen based on the previous convolutional layer which has been recovered in order that we can also implement the method discussed in Sect. 3.2 for the subsequent convolutional layer. In our experiment, we first launch the HPA to recover the weight of the first convolutional layer of the targeted CNN, and the accuracy of this method is 18.7917%. Then, we reverse engineer the weights of the first convolutional layer of the targeted CNN using CP-HPA, and the accuracy of our attack is 84.625%.



(a) CP-HPA (b) CPA

Fig. 9. The correlation of correct weight and other candidates respectively using horizontal CP-HPA and CPA can be distinguished. The red one represents correct weight and the rest represents other candidates. (Color figure online)

5 Conclusion

Numerous design strategies for the neural network have been proposed with the popularity of neural network algorithms, and the trained weights have become one of the main factors in determining the efficiency of neural networks. In this work, we demonstrate reverse engineering of weights of a CNN using side-channel analysis techniques. Concrete attacks are performed on simulation data which is generated by ELMO for chosen neural network implemented on ARM Cortex-M3. We conclude that all of the weights of the first convolutional layer can be recovered using the CP-HPA technique. The proposed attacks draw on the previous work, we target the other parameters and make changes in the method according to our target compared to them.

References

1. Amiel, F., Feix, B., Villegas, K.: Power analysis for secret recovering and reverse engineering of public key algorithms. In: Adams, C., Miri, A., Wiener, M. (eds.) SAC 2007. LNCS, vol. 4876, pp. 110–125. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77360-3_8
2. Batina, L., Bhasin, S., Jap, D., Picek, S.: {CSI}{NN}: reverse engineering of neural network architectures through electromagnetic side channel. In: 28th USENIX Security Symposium (USENIX Security), pp. 515–532 (2019)
3. Batina, L., Bhasin, S., Jap, D., Picek, S.: Poster: recovering the input of neural networks via single shot side-channel attacks. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp. 2657–2659 (2019)
4. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28632-5_2
5. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Kaliski, B.S., Koç, K., Paar, C. (eds.) CHES 2002. LNCS, vol. 2523, pp. 13–28. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36400-5_3
6. Clavier, C., Feix, B., Gagnerot, G., Roussellet, M., Verneuil, V.: Horizontal correlation analysis on exponentiation. In: Soriano, M., Qing, S., López, J. (eds.) ICICS 2010. LNCS, vol. 6476, pp. 46–61. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17650-0_5
7. Coron, J.-S.: Resistance against differential power analysis for elliptic curve cryptosystems. In: Koç, Ç.K., Paar, C. (eds.) CHES 1999. LNCS, vol. 1717, pp. 292–302. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48059-5_25
8. Gao, S.: sca-research/ELMO. https://github.com/bristol-sca/ELMO (2021)
9. Hua, W., Zhang, Z., Suh, G.E.: Reverse engineering convolutional neural networks through side-channel information leaks. In: 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), pp. 1–6. IEEE (2018)
10. Kim, J., Hong, S., Han, D.G., Lee, S.: Improved side-channel attack on des with the first four rounds masked. ETRI J. **31**(5), 625–627 (2009)
11. Kober, J., Bagnell, J.A., Peters, J.: Reinforcement learning in robotics: a survey. Int. J. Robot. Res. **32**(11), 1238–1274 (2013)

12. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_25
13. Krizhevsky, A., Nair, V., Hinton, G.: CIFAR-10. Canadian Institute for Advanced Research (2009)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems **25** (2012)
15. Lai, L., Suda, N., Chandra, V.: CMSIS-NN: efficient neural network kernels for Arm Cortex-M CPUs. arXiv preprint arXiv:1801.06601 (2018)
16. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. The Handbook of Brain Theory and Neural Networks **3361**(10), 1995 (1995)
17. Lo, O., Buchanan, W.J., Carson, D.: Power analysis attacks on the AES-128 S-box using differential power analysis (DPA) and correlation power analysis (CPA). J. Cyber Secur. Technol. **1**(2), 88–107 (2017)
18. Maji, S., Banerjee, U., Chandrakasan, A.P.: Leaky nets: recovering embedded neural network models and inputs through simple power and timing side-channels-attacks and defenses. IEEE Internet Things J. **8**(15), 12079–12092 (2021)
19. McCann, D., Oswald, E., Whitnall, C.: Towards practical tools for side channel aware software engineering: 'grey box' modelling for instruction leakages. In: 26th USENIX Security Symposium (USENIX Security), pp. 199–216 (2017)
20. Teufl, P., Payer, U., Lackner, G.: From NLP (natural language processing) to MLP (machine language processing). In: Kotenko, I., Skormin, V. (eds.) MMM-ACNS 2010. LNCS, vol. 6258, pp. 256–269. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14706-7_20

MineSOS: Long-Range LoRa-Based Distress Gesture Sensing for Coal Mine Rescue

Yuqing Yin[1], Xiaojie Yu[1], Shouwan Gao[1](✉), Xu Yang[1,2], Pengpeng Chen[1,3], and Qiang Niu[1,3]

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China
{yinyuqing,yuxiaojie,gaoshouwan,yang_xu,chenp,niuq}@cumt.edu.cn

[2] Xuzhou Kerui Mining Technology Co., Ltd, Xuzhou 221000, China

[3] Mine Digitization Engineering Research Center of Ministry of Education of the People's Republic of China, China University of Mining and Technology, Xuzhou 221116, China

Abstract. Distress signal identification has great significance in saving lives in coal mine rescue. In response to the rescue in the long-distance coal mine tunnel full of dust and dangerous gas, we propose a distress gesture sensing system utilizing LoRa technology, called MineSOS. Inspired by the Morse code "SOS" and binary code, we first present a set of distress gestures with an error-check design, only requiring one hand movement. For signal processing, we propose a novel metric to choose the optimal LoRa attribute due to the observation of the complementary relationship between amplitude and phase variation. Finally, a double-check mechanism is presented to recognize and verify the distress information. We conducted extensive experiments to evaluate MineSOS's performance, and results show that it can achieve high accuracy for gesture recognition in a coal mine lab. MineSOS system also has the capability of long-range sensing, which is believed to benefit emergency coal mine rescue.

Keywords: LoRa · Distress gesture recognition · Coal mine rescue

1 Introduction

Underground coal mines are hazardous environments facing numerous problems such as ground movements, gas explosions, and air blasts, resulting in great potential for large-scale environmental damage and loss of life [9]. When a mine accident occurs, the most urgent and essential task is to explore the vital signs and distress information of the trapped miners, including the thermal radiation

This work was supported by the National Natural Science Foundation of China under Grant 51904294 and the Natural Science Foundation of Jiangsu Province under Grant BK20221109.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 105–116, 2022.
https://doi.org/10.1007/978-3-031-19214-2_9

signs of the personnel, distress cries or tapping sounds, and photoelectric signals using wireless equipment.

One of the promising life detectors is based on infrared sensors or thermal infrared cameras through detecting radiant heat radiated by human bodies even in the darkness [4]. However, dusty and foggy environments and the construction of long and narrow tunnels limit its detection range and make it fail when rock obstructions are blocking in between. Acoustic life detector is used to refine the sonic caused by vocal signs or knocking, while its detection quality is severely affected by environmental noises [10]. Radiofrequency signals such as Wi-Fi can also monitor human's vital signs [8,11], but it suffers from the limitation of short sensing range.

This paper proposes a novel contactless sensing system based on the emerging LoRa technology, which is more suitable for long-range emergency coal mine rescues. LoRa, a Low Power Wide Area Network (LPWAN) technology, is designed to allow long-range wireless communications in low-power and low-cost ways [2,5]. LoRa can enable effective data transmission in urban areas to a few kilometers and has strong penetration capability through obstacles. This paper explores the opportunities and feasibility of LoRa technology for human life detection in coal mine rescues.

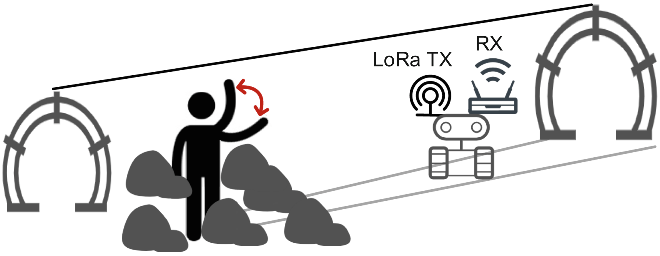


Fig. 1. Concept of MineSOS system for coal mine rescues.

The high-level idea of our designed system is shown in Fig. 1. When a person regains consciousness after the accident, he/she can make some distress gestures. The rescue robot carrying a pair of LoRa transceiver devices runs along the mine tunnel to collect echo signals. By analyzing the characteristics of the signal reflected by human activities, we can parse out the distress information embedded in gestures. However, it does not make sense to make gestures randomly. How to design a series of emergency gestures that carry certain distress information and are suitable for long-range mine rescues becomes a challenging issue.

Inspired by that the international distress signal "SOS" is an unbroken sequence of three dots, three dashes and three dots, we realize that we can splice several sub-gestures together to present one piece of information. In this paper, we first give the design of sub-gestures and then the distress signal "SOS" is composed of the combination of corresponding sub-gestures. In response to the issue that existing research always uses one attribute of LoRa signal, either amplitudes [12] or phases [6] for processing, resulting in performing not well at different locations, we carefully analyze the effect of the relationship between amplitudes and phases on sensing performance in theory and propose a novel method to choose the optimal attribute of LoRa signals for gesture identification. Another issue is that the gesture patterns are not clear and inconsistent with the ideal ones as signals get attenuated and become blurred after long-range propagation, causing identification errors. In response to this problem, we propose a gesture identification algorithm combining gesture pattern recognition and correlation coefficient comparisons. Finally, we introduce the verification mechanism with check codes to double-check the distress information. The main contributions of this paper are summarized as follows.

- We propose MineSOS, a distress gesture sensing system for emergency rescues, which only requires only one arm movement. By combining with LoRa sensing technology, this system can effectively work in the long-range coal mine environment.
- We design a series of distress gestures with a verification mechanism to be friendly used for the trapped user and to be reliably recognized after long-distance propagation. We also introduce a novel method to choose the optimal LoRa attribute for sensing and a double-check approach for gesture recognition.
- We prototype MineSOS and evaluate it in both indoor scenarios and a real coal mine environment with different users. Extensive experimental results show that MineSOS can recognize distress gestures with an accuracy of 94%.

2 System Overview

Figure 2 illustrates the outline of our proposed system. A set of user-friendly distress gestures are the foundation. We pick up several common sub-gestures to present the information elements like dot and dash. With a certain coding rule, several sub-gestures can build up a piece of distress information. A check code mechanism is added to make the information robust after identification. When the transceiver devices collect a period of LoRa signals, the system will do a series of data processing methods for distress information analysis. After denoising raw signals, we propose a metric, MVSR, to judge whether amplitude or phase is the optimal attribute for sensing. Then, we segment the LoRa signals into several fragments to better identify each sub-gesture. Finally, we restore the information of this set of gesture sequences and utilize a double-check mechanism to verify the correctness of the distress information.

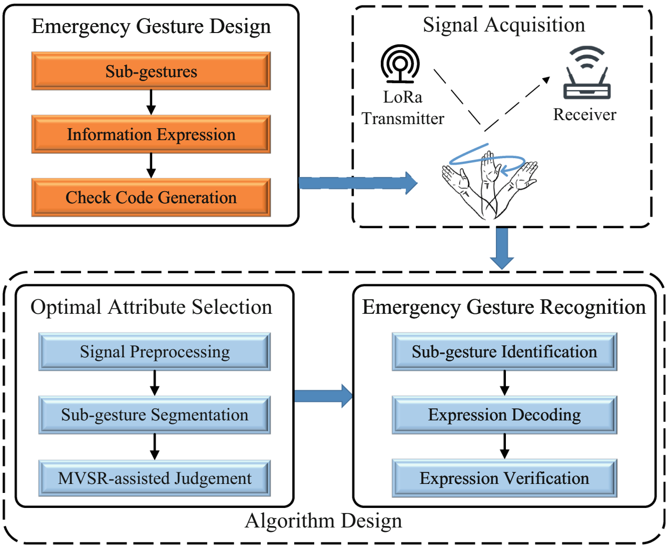


Fig. 2. Overview of the MineSOS system.

3 Gesture Design

3.1 Sub-gestures

Morse code is a method to encode text characters as standardized sequences of two different duration called dot and dash [3]. In the international Morse code, three dots form the letter "S" and three dashes make the letter "O", so the distress signal "SOS" is correspondingly the order of dots and dashes. Binary code is another encoding text method that is often used in computing and telecommunications, where the two symbols are "0" and "1". To better express the distress information in a united way, we map the "dot" and "dash" in Morse codes to the symbols "0" and "1", respectively. As shown in Fig. 3, we design a series of sub-gestures to represent basic symbols requiring only one hand. To clarify the completeness of a piece of information, we design the *hand-up* gesture [Fig. 3(a)] as the start point and the *hand-down* gesture [Fig. 3(b)] as the end point. The gesture *wave-to-one-side* [Fig. 3(d)] corresponds to the symbol "0" and the gesture *wave-left-and-right* [Fig. 3(e)] corresponds to the symbol "1". The gesture *hand-upright* [Fig. 3(c)] represents the pause between symbols. These sub-gestures utilize the natural hand actions and can easily make up a piece of distress information.

3.2 Distress Information Expression

There are two categories of information that a trapped miner can delivery: 1) the SOS signal and 2) the health status to express emergency level. We transfer

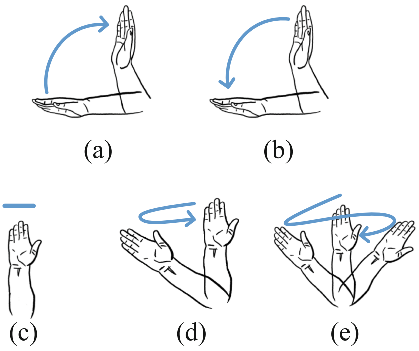
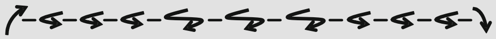
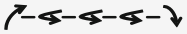
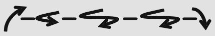
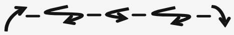
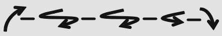


Fig. 3. Sub-gestures.

the Morse codes of "SOS" into binary codes and then the gestures sequence is shown in Fig. 4(a). For the consideration of allocation of rescue resources, health statuses are classified into four levels: *1) bad, 2) fair, 3) good* and *4) excellent*. Corresponding to 2-bit binary numbers, they are "00", "01", "10", and "11", respectively. To avoid fault identification of one bit in sub-gestures, we add an even parity bit for error detection. For a given set of bits, if the occurrence count of the bit "1" is odd, the parity bit value is set to 1, making the total occurrence count of the bit "1" in the whole set (including the parity bit) an even number. If the count in a given set of bits is already even, the parity bit's value is 0. Finally, the gestures sequences for health statuses are illustrated in Fig. 4(b).

Information	**Binary Expression**	**Gestures**
SOS	000111000	

(a) expression of distress signal and gestures

Information		**Binary Expression**	**Added Parity Bit**	**Gestures**
Status	Bad	00	00 **0**	
	Fair	01	01 **1**	
	Good	10	10 **1**	
	Excellent	11	11 **0**	

(b) expressions of health status and gestures

Fig. 4. Distress information expressions.

3.3 Observation

It is known that the larger the signal variation, the better the recognition performance [6]. However, we observe that for one set of LoRa signals, variations of its two attributes (i.e., amplitude and phase) have different changing scales. The rationale of this situation can be analyzed by vector expressions as shown in Fig. 5. From sent by a LoRa node to collected by a receiver, LoRa signals propagate along with two categories of paths: static and dynamic paths. Static paths are composed of the line-of-sight (LoS) path and reflected paths from walls or grounds, represented as a static vector V_s in I/Q space. Dynamic paths are the motion-reflected signal paths, represented as a dynamic vector V_d rotating with respect to the static vector. The composite signal is what we retrieve from the receiver, and its vector representation V_c is the sum of V_s and V_d. We record the average vector of the rotating V_d as V_{da}. It is observed that with different angle differences between V_s and V_{da}, the variations of amplitude and phase present different scales. This angle difference can be a random value ranging from 0 to 360 °C as a fine environmental change can cause a casual change of the initial phase. The angle difference is 90° [Fig. 5(a)], the amplitude varies larger than the phase, while the situation becomes the opposite when the difference is 180° [Fig. 5(b)].

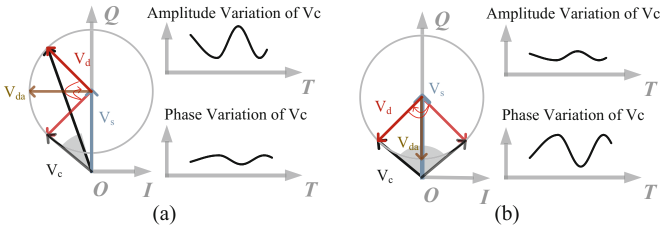


Fig. 5. Rational of LoRa sensing in the vector space.

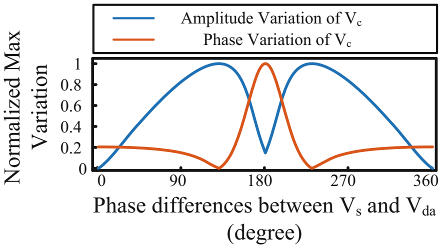


Fig. 6. Impact of angle differences on amplitude and phase variations.

We simulate all the situations of signal variation induced by different angle differences and show the maximized amplitude and phase variations in Fig. 6. It can be inferred that the amplitude and phase are complementary in the variation scale, and we can utilize this characteristic to choose an attribute with larger variation to recognize gestures better.

4 MineSOS System Design

In this section, we will introduce core modules of MineSOS system: 1) optimal attribute selection and 2) emergency gesture recognition.

4.1 Optimal Attribute Selection

Signal Preprocessing. In MineSOS, the receiver is equipped with two antennas; thus, two LoRa streams can be obtained simultaneously from sampled packets. We first calculate the ratio of two signal streams to remove random phase offsets for further processing [6]. Then, we separately calculate the normalized attributes, i.e., amplitude and phase of the signal-ratio. We plot the normalized amplitude in Fig. 8 and consider it as an instance to introduce methods of segmentation and MVSR-assisted selection.

Sub-gesture Segmentation. Figure 7 shows the amplitude variation scales that are calculated as the differences between the maximum and minimum amplitudes in each 0.5-s sliding window with a step of 0.2 s. As we can seen from Fig. 8, the amplitude variation of the static gesture *hand-upright* is almost 0.25. In order to distinguish different consecutive gestures, we apply a threshold to separate sub-gestures. The threshold is set to 0.25 in our design and is denoted as the horizontal red line in Fig. 7. Then the signal is accordingly segmented to different parts for sub-gestures.

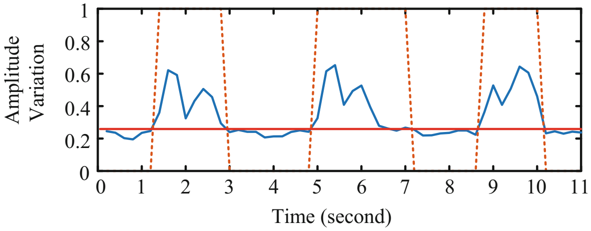


Fig. 7. Sub-gesture segmentation.

MVSR-Assisted Selection. As mentioned in Fig. 6, amplitude and phase are complementary in variation scale, and we should choose the attribute with larger variation for gesture sensing. We create a new metric, named Motion gesture induced Variation to Static gesture induced variation Ratio (MVSR), to quantify the variation performance. Taking Fig. 8 as an example, one LoRa stream is separated into different parts for sub-gestures. We calculate the max-min variation of each sub-gesture, and then MVSR is defined as the ratio of variation sum of motion gestures to that of static gestures. Finally, we select amplitude or phase with larger MVSR for further processing.

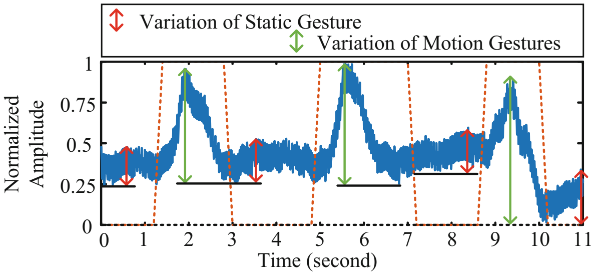
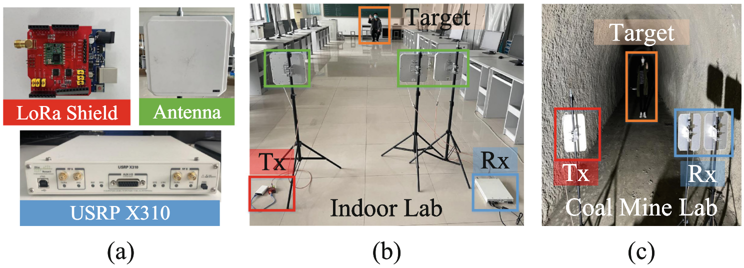


Fig. 8. Signals for three consecutive gestures *wave-to-one-side.*

4.2 Emergency Gesture Recognition

Sub-gesture Identification. We propose a double-check mechanism for sub-gesture identification. The first-level identification is to compare the similarity of the segmented signal with the signal pattern, where the signal patterns for sub-gestures are obtained from a pre-experiment. By leveraging the dynamic time warping (DTW) algorithm [7], we can find the best-matching reference pattern for one segmented signal. After one round of sub-gesture identification, the second-level identification for symbols “0” and “1” starts working. We calculate the correlation coefficient between every two-segmented signals that are recognized as the same symbol. If one symbol “0” is misidentified as “1”, the correlation coefficient between the error symbol and each other symbol “1” may be much lower than the correlation coefficient between correct symbols “1”. When the correlation coefficient is more than twice as small as others, we shall correct it to the opposite symbol.

Expression Decoding and Verification. The expression can be obtained by splicing the identified symbols, and the distress information is decoded by matching the binary expressions. When a piece of health information is got, we can perform the exclusive or (XOR) operations among all bits to verify its correctness. If the XOR result is “0”, the information is decoded correctly, while the result is “1”, it means that the information is invalid.



(a) (b) (c)

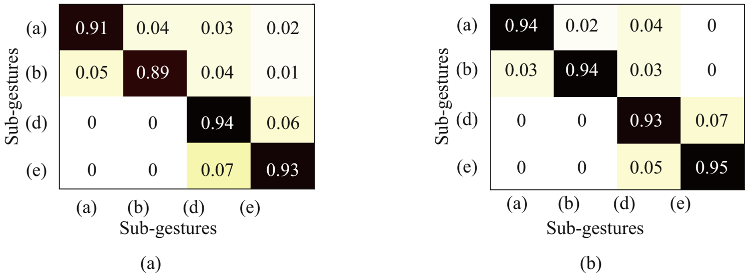
Fig. 9. Experimental implementation: (a) hardware, (b) indoor lab scenario, and (c) coal mine lab scenario.

5 System Evaluation

5.1 Implementation

Figure 9 presents the system implementations, including hardware equipment and experimental scenarios. The transmitter is a Semtech SX1276 LoRa shield [1], which is connected to a 9-dbi directional antenna. A USRP X310 connecting with two antennas is served as the receiver, working with the Labview software to retrieve LoRa signals. As shown in Fig. 9 (b) and (c), the transmitting antenna and receiving antennas are put on one side of the lab, while a target is operating gestures on the other side of the lab. The default distance between the transceiver pair is 1.4 m, and the default height is 1.2 m. The target makes gestures 10 m away from the transceiver pair in the indoor lab while 5 m in the coal mine lab. In two scenarios, we recruit four volunteers and ask each volunteer to repeat each sub-gesture and distress information over one hundred times.

5.2 Overall Performance



	(a)	(b)	(d)	(e)
(a)	0.91	0.04	0.03	0.02
(b)	0.05	0.89	0.04	0.01
(d)	0	0	0.94	0.06
(e)	0	0	0.07	0.93

(a)

	(a)	(b)	(d)	(e)
(a)	0.94	0.02	0.04	0
(b)	0.03	0.94	0.03	0
(d)	0	0	0.93	0.07
(e)	0	0	0.05	0.95

(b)

Fig. 10. Confusion matrix of sub-gesture recognition in (a) indoor lab and (b) coal mine lab.

This section shows MineSOS's overall performance of gesture recognition in two different scenarios. We utilize recognition accuracy as the metric to measure the system performance, where the accuracy is the percentage of correctly recognized sub-gestures or distress information. Figure 10 shows the confusion matrices of recognition accuracy of four sub-gestures. It is noted that gesture *hand-upright* is used to separate other sub-gestures, and its accuracy is not counted in the results. Figure 11 shows the recognition accuracy of different distress information. It can be inferred that MineSOS performs well for all distress information, achieving 93.6% and 94.3% recognition accuracy on average for indoor scenario and coal mine scenario, respectively.

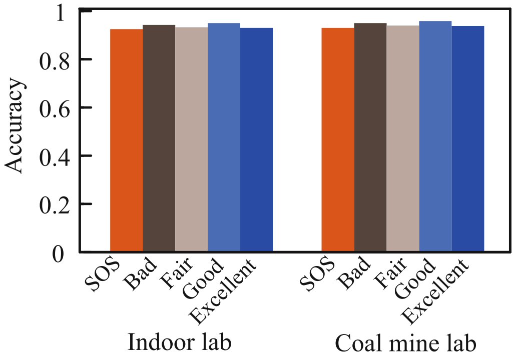


Fig. 11. Accuracy of distress information.

5.3 Evaluation of Designed Methods

This experiment evaluates the proposed optimal attribute selection method. Figure 12 shows the accuracy of gesture recognition by comparing methods of only using amplitude or phase for gesture recognition.

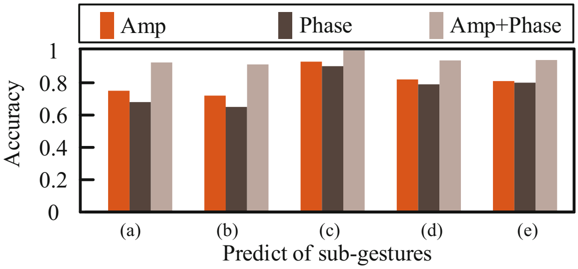


Fig. 12. Evaluation of optimal attribute selection.

For sub-gesture recognition, average accuracies obtained by the three methods are 80.6%, 76.4% and 94.2%, respectively. It can be seen from the experimental results that the complementary nature of amplitudes is helpful in obtaining optimal attribute signals with strong sensing performance, thereby improving the accuracy of gesture recognition.

We also evaluate the double-check mechanism and present the gesture recognition accuracies in Fig. 13. Compared with only using DTW for gesture matching, the accuracy of the double-check mechanism has been improved by 9.6%. It can be seen that the double-check mechanism can effectively correct individual wrongly recognized sub-gestures and improve the performance of gesture recognition.

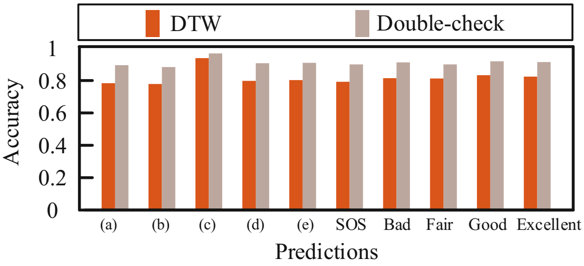


Fig. 13. Evaluation of double-check mechanism.

5.4 Exploring the Limit of Sensing Range

In this section, we explore the limit of MineSOS's sensing range in an LoS environment. We carry out the experiments in an indoor corridor, where the transceiver setup is similar to in the coal mine lab. We vary the distance between the target and the transceiver pair from 10 m to 40 m in a step of 10 m. The accuracy decreases as the distance increases, and the corresponding accuracies are 94%, 92%, 90% and 88%, respectively. When we try to enlarge the distance to 50 m, the signal patterns are not clear anymore and then we conclude that our designed system MineSOS can achieve a limit of 40 m in gesture sensing range.

6 Conclusion

In this paper, we have designed a LoRa-based contactless gesture recognition system for coal mine rescue. Combining the Morse code and binary code, we design a series of distress gestures for SOS distress information and expressions of health status. Through an in-depth analysis of the rationale of LoRa-based activity sensing, we observe a complementary relationship between LoRa amplitude and phase. We then propose a novel metric, MVSR, to choose the optimal attribute for gesture recognition. After utilizing a double-check mechanism, MineSOS can identify different sub-gestures and decode the distress information. Extensive experiments have verified the effectiveness of our system in both indoor and coal mine environments. It can achieve a 40 m sensing range with good recognition accuracy.

Acknowledgements. The authors would like to thank the volunteers that had helped to conduct the experiments.

References

1. Lora shield. https://www.dragino.com/products/lora/item/102-lora-shield.html
2. Liando, J.C., Gamage, A., Tengourtius, A.W., Li, M.: Known and unknown facts of LoRa: experiences from a large-scale measurement study. ACM Trans. Sens. Netw. **15**(2), 1–35 (2019)
3. Niu, K., et al.: WiMorse: a contactless Morse code text input system using ambient WiFi signals. IEEE Internet Things J. **6**(6), 9993–10008 (2019)
4. Shetty, A.D., Shubha, B., Suryanarayana, K., et al.: Detection and tracking of a human using the infrared thermopile array sensor- "grid-eye". In: 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), pp. 1490–1495. IEEE (2017)
5. Talla, V., Hessar, M., Kellogg, B., Najafi, A., Smith, J.R., Gollakota, S.: LoRa backscatter: enabling the vision of ubiquitous connectivity. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **1**(3), 1–24 (2017)
6. Xie, B., Yin, Y., Xiong, J.: Pushing the limits of long range wireless sensing with LoRa. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **5**(3), 1–21 (2021)
7. Yadav, S.K., Tiwari, K., Pandey, H.M., Akbar, S.A.: A review of multimodal human activity recognition with special emphasis on classification, applications, challenges and future directions. Knowl.-Based Syst. **223**, 106970 (2021)
8. Yang, X., Yin, Y., Chen, P., Niu, Q.: A device-free intelligent alarm system based on the channel state information. IEEE Trans. Veh. Technol. **69**(10), 11419–11427 (2020). https://doi.org/10.1109/TVT.2020.3010645
9. Yang, X., Yu, X., Zhang, C., Li, S., Niu, Q.: MineGPS: battery-free localization base station for coal mine environment. IEEE Commun. Lett. **25**(8), 2579–2583 (2021)
10. Ye, H.: Life detection technique in earthquake search and rescue. In: 2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control, pp. 664–666. IEEE (2012)
11. Yin, Y., Yang, X., Xiong, J., Lee, S.I., Chen, P., Niu, Q.: Ubiquitous smartphone-based respiration sensing with Wi-Fi signal. IEEE Internet Things J. **9**(2), 1479–1490 (2022). https://doi.org/10.1109/JIOT.2021.3088338
12. Zhang, F., et al.: Exploring LoRa for long-range through-wall sensing. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **4**(2), 1–27 (2020)

Weighted Data Loss Minimization in UAV Enabled Wireless Sensor Networks

Zhengzhong Xiang[1], Tang Liu[2], and Jian Peng[1](✉)

[1] College of Computer Science, Sichuan University, Chengdu, China
xiangzz@stu.scu.edu.cn, jianpeng@scu.edu.cn

[2] College of Computer Science, Sichuan Normal University, Chengdu, China
liutang@sicnu.edu.cn

Abstract. With high mobility and adaptability, the Unmanned Aerial Vehicle (UAV) has provided a promising solution for data collection in Wireless Sensor Networks (WSNs). However, few existing works considered that data overwritten would occur if the UAV can not collect data from sensors in time, which will cause data loss in WSNs. Moreover, the importance of data stored in different sensors may vary significantly according to the application scenario. In this paper, we formulate a novel Loss Minimization Problem (LMP) in a UAV-enabled WSN. The objective is to minimize the volume of weighted data loss in the WSN by jointly considering the UAV hovering locations and hovering durations, subject to the limited energy capacity. We first devise a novel one-to-many data collection scheme that enables the UAV to collect data from multiple sensors simultaneously. Then we discrete the infinite hovering locations of the UAV into finite to reduce computational complexity. We instead propose efficient heuristic and approximation algorithms for the optimization problem. Finally, we evaluate the performance of the proposed algorithms through extensive experimental simulations. Simulation results demonstrated that the proposed algorithms are promising.

Keywords: Unmanned aerial vehicle · Wireless sensor network · Data collection · Data loss minimization · Trajectory optimization

1 Introduction

Wireless Sensor Networks (WSNs) have recently risen to prominence as a promising technology for various applications, including smart cities [13], industrial monitoring [4] and so on. In a WSN, numerous low-cost sensors are deployed in detection areas to monitor the environment and periodically generate massive sensing data. However, constrained by the portable size, most sensors typically have limited storage capacities and are powered by batteries with limited energy sources. Therefore, to prolong the lifetime of WSNs, it's unrealistic to transmit their sensing data to the base station (BS) directly because of the significant transmission energy consumption [5]. Furthermore, sensors continuously generate massive sensing data with limited storage, rendering it a critical issue to collect data timely in WSNs to avoid data loss and data overwritten [12].

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 117–129, 2022.
https://doi.org/10.1007/978-3-031-19214-2_10

With the high mobility and adaptability, Unmanned Aerial Vehicles (UAVs) have recently emerged as a promising solution for data collection in WSNs [6,10]. In a UAV-enabled WSN, when sensors request data collection, the BS dispatches UAVs to fly close to and hover at specific locations to collect sensing data and deliver it back to the BS. Since sensors are generally deployed in complex environments such as cities and hilly terrain, UAVs with high mobility can easily reach destinations compared to conventional terrestrial collectors. On the other hand, the communications between UAVs and sensors are less affected by channel impairments such as shadowing and fading, enabling UAVs to reach better communication quality and a more extended transmission range [14].

In recent years, extensive studies have been conducted to explore data collection in UAV-enabled WSNs [5,8]. However, most existing studies face the potential data loss risk. When the size of the WSN is large, sometimes there may be a significant number of sensors requesting data collection concurrently. Limited by the number of available UAVs, we cannot always collect data from all sensors in time. By most existing data collection schemes, people tend to prioritize sensors that have stored more data to maximize the gains of a single trip, ignoring that data overwritten would occur while the UAV can not collect data from sensors in time, which may incur critical data loss. Moreover, unlike existing studies that consider all sensors as isomorphic and ignore the different importance of data stored in different sensors, we notice that the importance of data stored in sensors may vary significantly, determined by the scenario they monitor. For example, sensors monitoring vehicle traffic are more critical than monitoring city temperatures.

As the above observation, we formulate a novel Loss Minimization Problem (LMP): schedule hovering locations and hovering durations for an energy-constrained UAV to collect data from sensors such that the volume of weighted data loss in the WSN is minimized. Tackling this problem faces many challenges. For example, in WSNs, a UAV is typically powered by an energy-limited battery, so the data collection path should be a closed tour so that the UAV can complete the task before the battery dead. Furthermore, while planning the UAV hovering locations and durations, we should trade off the remaining storage capacities of sensors and the UAV moving time, making the problem further complex. Moreover, there are infinite potential hovering locations for the UAV in the area, which impose a significant computational burden to determine the finalized path. To this end, we propose a spatial discretization algorithm to reduce computational complexity, and we devise efficient heuristic and approximation algorithms for LMP, which significantly decline the data loss in the WSN compared with three baseline algorithms.

The main contributions of this paper are summarized as follows:

- We formulate a novel Loss Minimization Problem (LMP) to minimize the volume of weighted data loss in the WSN under the assumption that the energy capacity of the UAV is limited. We consider a one-to-many data collection scheme that enables the UAV to collect data from multiple sensors simultaneously, improving the data collection efficiency significantly.

- To make the problem tractable, we first discrete the infinite hovering locations of the UAV into finite. Then we devise an efficient heuristic algorithm for the optimization problem by jointly considering the energy capacity of the UAV and the priorities of sensors.
- We finally evaluate the performance of the proposed algorithm through extensive experimental simulations. Simulation results demonstrate that the proposed algorithm is promising.

2 Related Work

This section reviews the literature related to the scheduling of the UAV for data collection in WSNs. Based on the number of UAVs in WSN, we classify the investigated problems into two types: single-UAV WSN and multiple-UAV WSN.

Single-UAV WSN: Li et al. [5] investigated the problem of employing an energy-constrained UAV to collect data from IoT sensors in a sparse WSN, aiming to maximize the accumulative volume of data collected per tour. They constructed a closed tour for the UAV to fully or partially collect data from sensors, consisting of hovering locations and the sojourn duration at each hovering location. In [11], Dai et al. assigned dynamical priorities to the data according to the importance of reconnaissance areas. They aimed to maximize the overall reconnaissance utility subject to the UAV energy constraint and priorities. In [12] Samir et al. studied a time-constrained IoT devices data collection problem, where each device has its own target data upload deadline. They jointly optimized the trajectory of a UAV and the radio resource allocation to maximize the number of served IoT devices.

Multiple-UAV WSN: In [16], Xu *et al.* considered a problem of employing the minimum number of UAVs to collect data from sensors and constructed closed tours for them, subject to the constraint that the duration of each data collection tour is no longer than a given delay. In [8] Luo et al. investigated a novel fine-grained trajectory plan problem in which multiple UAVs are employed for data collection. The objective of the problem is to minimize the maximum flight time while guaranteeing all sensing data of WSN is fully collected. In [17] Zhan et al. studied the data collection problem in WSNs enabled by multiple UAVs to minimize the maximum mission completion time. They jointly optimized the UAV path and sensors wake-up scheduling while guaranteeing that UAVs can successfully collect data from each sensor with a given energy budget. In [15], Xiang et al. jointly optimize the route selection, sensing time, and delivery weight allocation while accounting for interdependency between flying/sensing and the UAV delivery weight, to maximize delivery and sensing utility with the limited energy of UAVs.

However, few existing works considered that data overwrite would occur while the UAV can not gather data from sensors timely. In this paper, we investigates the Loss Minimization Problem, which aims to minimize the volume of weighted data loss in WSN subject to the UAV energy capacity.

3 Preliminaries

In this section, we first introduce the system model and then formulate the loss minimization problem precisely.

3.1 System Model

We consider a WSN where a set of N sensors $V = \{v_i | 1 \leq i \leq N\}$ are distributed over a two-dimensional to-be-monitored area on the ground. Denote by $(x_i, y_i, 0)$ the coordinate of sensor $v_i \in V$. Assume that each sensor v_i continuously monitors its surroundings and generates r_i units of data per second, with D_i units of data stored locally waiting to be collected, and the storage capacity is capped at C_i. Denote by rt_i the remaining time of sensor v_i, where $rt_i = \frac{C_i - D_i}{r_i}$, implying that data overwritten will occur if data collection on v_i cannot begin within this duration. We employ a weight variable w_i to measure the importance of data generated by sensor v_i, where $0 < w_i \leq 1$. A higher value of w_i means that the data is of greater importance, so we should try to avoid or reduce data loss in such a sensor. The value of w_i is predetermined according to the importance of the monitoring scenario of the sensor v_i. When the storage capacity C_i falls below a predefined threshold θ, sensor v_i will send a data collection request to the BS, and the BS will insert v_i into a to-be-collected queue Q_t.

When the UAV is available, the BS will construct a data collection plan according to the information from the sensors in Q_t. The UAV takes off from the BS and moves along the path to collect data from sensors at a constant speed s and a fixed altitude H above the ground. Since a UAV is typically powered by an energy-limited battery, it must return to the base station for recharging before its battery dead, indicating that the data collection path must be closed. Assuming the energy capacity of the UAV is $\mathcal{E}$, the energy consumption in the tour includes the UAV moving consumption E_f and the energy consumption E_h for hovering to collect data from sensors.

Supporting by the Orthogonal Frequency Division Multiple Access (OFDMA) technique [9], the UAV can simultaneously collect data from multiple sensors within its communication range R. Specifically, each sensor v_i can transmit its data to the UAV when the distance between v_i and the UAV is no greater than R. The UAV hovers at specific locations to collect data from sensors. Denote by $\mathcal{H} = \{h_0, h1, ..., h_K\}$ the sequence of hovering locations for the UAV in the tour, where h_0 is the BS, and $\{X_k, Y_k, H\}$ denotes the coordinate of h_k, where $0 < H \leq R$. Denote by $V(h_k)$ the set of sensors whose data can be collected while the UAV is hovering at h_k, i.e.,

$$V(h_k) = \{v_i | dis(v_i, h_k) \leq R^2, v_i \in V\}, \tag{1}$$

where $dis(v_i, h_k)$ is the distance between hovering location h_k and sensor v_i.

Let B_{ik} denotes the data transmission rate of sensor $v_i \in V(h_k)$ when the UAV hovers at h_k. Following the equation in [1,8], B_{ik} can formulate as

$$B_{ik} = \log(1 + \frac{\sigma_0}{dis(v_i, h_k)^\alpha}), \tag{2}$$

where σ_0 is the transmission power of v_i, and α is the path loss exponent.

When the UAV hovers at h_k to collect data from sensors in $V(h_k)$, its hovering duration $t(h_k)$ can be calculated as

$$t(h_k) = \max_{v_i \in V(h_k)} \{\frac{D_i}{B_{ik}}\}, \tag{3}$$

where D_i is the volume of data stored in v_i. Denote by $f(h_{k-1}, h_k) = \frac{dis(h_{k-1}, h_k)}{s}$ the moving time for the UAV to fly from location h_{k-1} to h_k, where s is the moving speed of the UAV. The total length of the data collection tour for the UAV is $Dis(\mathcal{H}) = \sum_{i=0}^{K} dis(h_i, h_{i+1})$, where h_0 and h_{K+1} are the BS.

3.2 Problem Formulation

Based on the models above, we define a novel Loss Minimization Problem (LMP). We aim to schedule the hovering locations and hovering duration for a UAV to minimize the volume of weighted data loss in the WSN while guaranteeing the battery energy of the UAV is not exhausted during the whole data collection tour. We define $S(v_i)$ as the state of v_i. A sensor have two states: remaining spare storage space ($S(v_i) = 0$) and running out of storage ($S(v_i) = 1$, overflow), we can express it as

$$S(v_i) = \begin{cases} 0, & T_i \le rt_i \\ 1, & T_i > rt_i. \end{cases} \tag{4}$$

T_i is the arrival time that the UAV starts collecting data from sensor v_i, rt_i refers to the remaining time of v_i. We employ a metric DL to measure the volume of weighted data loss in a data collection tour, i.e.,

$$DL = \sum_{i=1}^{N} S(v_i)(T_i - rt_i)w_i r_i, \tag{5}$$

where N is the number of sensors in the WSN, r_i is the data generating rate of v_i, w_i is the priority of v_i with $0 < w_i \le 1$, the more critical the data stored in v_i, the higher its value of w_i. The objective of our LMP can be formalized as

$$minimize\ DL \tag{6}$$

subject to:

$$E_f = \sum_{k=0}^{K} f(h_k, h_{k+1})\eta_f, \tag{7}$$

$$E_h = \sum_{k=1}^{K} t(h_k)\eta_h, \tag{8}$$

$$E_f + E_h \le \mathcal{E}, \tag{9}$$

Constraint (7) and (8) shows the accumulative energy consumption of the UAV for moving and hovering over the tour, where η_f and η_h are the energy consumption rate of the UAV for moving and hovering, respectively. Constraint (9) shows that the energy capacity $\mathcal{E}$ constrains the energy consumption of the UAV in the data collection tour.

4 The Proposed Scheme

This section details our schemes for LMP and shows theoretical analysis.

4.1 Potential Hovering Locations Discretization

To solve the problem, we need to plan the sequence of hovering locations for the UAV to collect data from sensors. However, the potential hovering locations in the given area are infinite, posing heavy computational complexity. To tackle this challenge, inspired by the work in [5], we discretize the objective area to reduce the infinite numbers of potential hovering locations into finite.

For each sensor v_i, let $v_i^{'}$ be its projection on the UAV moving plane, whose coordinate is (x_i, y_i, H), where H is the flight altitude of the UAV. We can get a set of circles $\Phi(v_i^{'})$ centered at $v_i^{'}$ with radius $R_0 = \sqrt{R^2 - H^2}$, where R is the data transmission range. The UAV can collect data from v_i when it hovers within $\Phi(v_i^{'})$. We partition the circle $\Phi(v_i^{'})$ into finite numbers of equal squares with edge length $l > 0$. The locations within a square can be regarded as indistinguishable when the value of l is small enough, so we can assume that the UAV only hovers at the center of the square area. Consequently, we can discretize the given area into a set of finite potential hovering locations $\mathcal{P} = \{p_1, p_2, ..., p_M\}$.

4.2 Data Collection Tour Planning Algorithm

We construct a feasible data collection tour by selecting a sequence of hovering locations in $\mathcal{P}$ iteratively. The Data Collection Tour Planning algorithm (DCTP) proceeds as follows.

At first, the BS responds to the data collection requests and inserts them into a to-be-collected queue Q_t. We discretize the UAV hovering area into a sequence of potential hovering locations $\mathcal{P}$ to reduce the computational complexity.

Denote by $RT(p_i)$ the remaining time of each hovering location $p_i \in \mathcal{P}$, i.e.,

$$RT(p_i) = \min_{v_j \in V(p_i)} rt_j, \tag{10}$$

where $V(p_i)$ is the set of sensors whose data can be collected when the UAV hovers at p_i, rt_j is the remaining time of sensor $v_j \in V(p_i)$, $RT(p_i)$ is determined by the minimum remaining time of the sensor in $V(p_i)$. Since the primary objective of this work is to minimize the volume of weighted data loss, the UAV should preferentially visit the hovering location with the most urgent remaining

Algorithm 1: Data Collection Tour Planning

```
Input: A to-be-collected queue $Q_t$.
Output: A data collection tour $\mathcal{H}$ and the volume of weighted data loss $DL_{\mathcal{H}}$.
Discrete the potential hovering locations of sensors into $\mathcal{P}$;
Sort all hovering locations in $\mathcal{P}$ in increasing order of their remaining time;
$\mathcal{H} \leftarrow \{h_0\}$;
while $Q_t \neq \emptyset$ and $E(\mathcal{H}) < \mathcal{E}$ do
    Select the first hovering location $p_j$ from $\mathcal{P}$; $\mathcal{H}^{'} \leftarrow \mathcal{H} \cup \{p_j\}$;
    Calculate the time $T_j^{arr}$ for the UAV to arrive at $p_j$;
    if $T_j^{arr} < RT(p_j)$ and $E(\mathcal{H}^{'}) \leq \mathcal{E}$ then
        $\mathcal{H} \leftarrow \mathcal{H}^{'}$;
    else
        Call Algorithm 2 to optimize $\mathcal{H}$;
        for $i \leftarrow 1$ to $|\mathcal{H}|$ do
            Insert $p_j$ as the $i$th hovering location $h_i$ to join $\mathcal{H}$;
            Form a tour $\mathcal{H}^{'}$ by minimizing the volume of weighted data loss and
              promise $E(\mathcal{H}^{'}) \leq \mathcal{E}$;
        $\mathcal{H} \leftarrow \mathcal{H}^{'}$;
    Update the state of each sensor in $V(p_j)$;
    $Q_t \leftarrow Q_t - V(p_j)$;
    Update the hovering locations set $\mathcal{P}$;
Finally obtain the data collection tour $\mathcal{H}$ of the UAV.
```

time. We sort the hovering locations in $\mathcal{P}$ in increasing order of their remaining time, where a shorter remaining time indicates a higher visit priority.

Initially, the data collection tour only includes the BS h_0. For each iteration, we expand $\mathcal{H}$ by selecting an unvisited hovering location p_j with the most urgent remaining time as the new hovering location to insert into $\mathcal{H}$. The procedure will repeat until the to-be-collected queue Q_t is empty or the total energy consumption $E(\mathcal{H})$ of data collection tour exceeds the UAV energy capacity $\mathcal{E}$. Supported by the OFDMA technology, the UAV can simultaneously collect data from multiple sensors in $V(p_j)$ when it hovers at p_j. So the arrival time T_i of each sensor $v_i \in V(p_j)$ equals the arrival time T_j^{arr} of p_j, which can formulate as

$$T_j^{arr} = \sum_{i=0}^{K-1} (f(h_i, h_{i+1}) + t(h_i)). \tag{11}$$

If the UAV can timely reach p_j and meets both the energy and temporal constrain, p_j will be inserted at the end of $\mathcal{H}$ as the following hovering location to expand the data collection tour. If it fails, we invoke Algorithm 2 to optimize the current data collection tour by adjusting the order of hovering locations. We scan each sojourn location from h_1 to h_K in the optimal tour to find the best insertion position for p_j. The final chosen insertion position should satisfy the two conditions: (i) The energy consumption of the tour does not exceed the

Algorithm 2: Data Collection Tour Optimization

```
Input: An original data collection tour $\mathcal{H}$.
Output: An optimized data collection tour $\mathcal{H}_{opt}$.
$count \leftarrow 0$, $K \leftarrow |\mathcal{H}|$;
while $count < \mathcal{M}$ do
    Initialize an empty queue $\mathcal{H}_{temp}$;
    Randomly select two non-negative integers $i$ and $j$ $(i < j \leq K)$ from $\mathcal{H}$;
    Take the sub-path from $h_0$ to $h_{i-1}$ and add them in order to $\mathcal{H}_{temp}$;
    Take the sub-path from $h_i$ to $h_j$ and add them in reverse order to $\mathcal{H}_{temp}$;
    Take the sub-path from $h_{j+1}$ to $h_K$ and add them in order to $\mathcal{H}_{temp}$;
    if $Dis(\mathcal{H}_{temp}) < Dis(\mathcal{H})$ and $DL_{\mathcal{H}_{temp}} \leq DL_{\mathcal{H}}$ then
        $\mathcal{H} \leftarrow \mathcal{H}_{temp}, count \leftarrow 0$;
    else
        $count \leftarrow count + 1$;
$\mathcal{H}_{opt} \leftarrow \mathcal{H}$;
```

energy capacity of the UAV; (ii) The volume of weighted data loss should be minimal under the constrain (i).

After expanding the tour $\mathcal{H}$, we update the state $S(v_i)$ of each sensor $v_i \in V(h_k)$, where h_k is the hovering location in $\mathcal{H}$. Then update the to-be-collected queue Q_t to avoid repeated visits to the same sensors. Finally, we update the potential hovering locations set $\mathcal{P}$ based on the updated Q_t and sort it.

The detailed process of DCTP algorithm is given in Algorithm 1.

4.3 Data Collection Tour Optimization Algorithm

As demonstrated in line 10 of Algorithm 1, we use Algorithm 2 to optimize obtained path to minimize the weighted data loss further. In Algorithm 1, we first select the hovering location with the most urgent remaining time and insert it after the sequence without considering the spatial consumption, which may incur a redundant moving distance. To this end, we introduce a Data Collection Tour Optimization algorithm (DCTO) to reduce the detour by optimizing the order of hovering locations. The DCTO algorithm proceeds as follows.

Initially, there are K hovering locations in an original data collection tour $\mathcal{H}$ waiting for optimization. We first construct an empty queue $\mathcal{H}_{temp}$ to store the optimized tour, then we randomly select two non-negative integers i and j, where $i < j \leq K$. We intercept the sub-path $h_0 \rightarrow h_{i-1}$ from $\mathcal{H}$ and add them into $\mathcal{H}_{temp}$ in order. Then we take the sub-path $h_i \rightarrow h_j$ and add them into $\mathcal{H}_{temp}$ in reverse order. Subsequently, we also intercept the sub-path $h_{j+1} \rightarrow h_K$ and add then into $\mathcal{H}_{temp}$ in order. After reordering $\mathcal{H}$, we check the traveling length of the optimized tour $\mathcal{H}_{temp}$ and update $\mathcal{H}$ if the traveling length is shorter without causing more data loss in the network. The optimization process will repeat until continuous $\mathcal{M}$-times rounds can not lead to a better result.

The details of DCTO algorithm are presented in Algorithm 2.

4.4 Algorithm Analysis

We prove that the number of partitioned hovering locations is finite and show the theoretical analysis of the proposed algorithms.

Theorem 1. *The number of potential hovering locations in $\mathcal{P}$ is no greater than $(\frac{\pi R_0^2}{l^2}+1)|Q_t|$.*

Proof. Since the number of sensors in the to-be-collected queue is $|Q_t|$, the size of the UAV hovering area is no greater than $\pi R_0^2|Q_t|$, where R_0 is the radius of the projection circles. We can partition this contiguous area into a maximum of $\sum_{v\in Q_t}\lceil\frac{\pi R_0^2}{l^2}\rceil \leq (\frac{\pi R_0^2}{l^2}+1)|Q_t|$ identical squares with length l. So we can reduce the potential hovering locations into a finite number which is linearly related to the size of Q_t.

Theorem 2. *The solution of DCTO gets roughly $\sqrt{K}$ approximation ratio to the optimal solution, and its time complexity is bounded by $O(K^{10}\log K)$, where K is the number of hovering locations in the data collection tour $\mathcal{H}$.*

Proof. The DCTO algorithm adapted from a local search algorithm 2-opt, a well-known algorithm proposed to solve the Traveling Salesman Problem (TSP). The algorithm iteratively selects two hovering locations and reverses the order between them to optimize the data collection tour. In [7], Liu et al. show that we can achieve an approximation ratio of roughly $\sqrt{K}$, with a similar theoretical derivation as shown in [2]. Furthermore, in [3], Englert et al. proved the time complexity of 2-opt is bounded by $O(K^{10}\log K)$, whose proof process is rather complex. Here we omit the specific proof process due to space limitation.

5 Performance Evaluation

In this section, we carry out experimental simulations to evaluate the critical performance metrics of the proposed algorithms. And we investigate the impacts of essential parameters on the performance under different settings.

5.1 Simulation Setup

We consider a WSN consisting of 100–600 sensors randomly deployed within a $1500\,\mathrm{m}\times 1500\,\mathrm{m}$ square area and the BS located at the centre of the ground. The storage capacity C_i of each sensor is 1000 MB. The volume of data D_i stored in each sensor is randomly drawn from 0 MB to 1000 MB. The data generate rate r_i of each sensor ranges from 0 KB/s to 100 KB/s. The weight priority w_i of each sensor is randomly set between 0 to 1. A sensor will immediately send a data collection request to the BS when its storage capacity falls below the threshold $\theta = 25\%$. Here we employ a single UAV for data collection tasks, hovering at the altitude of $H = 60\,\mathrm{m}$, with a const moving speed $s = 10\,\mathrm{m/s}$ [16]. The UAV equips with a battery whose energy capacity $\mathcal{E} = 3 \times 10^5\,\mathrm{J}$, and the moving and hovering energy consumption rates are $\eta_f = 100\,\mathrm{J/s}$ and $\eta_h = 150\,\mathrm{J/s}$, respectively [5]. The data transmission range R and the channel bandwidth W are set as 100 m and 20 MB/s, respectively. We set the reference SNR at transmission distance 1 m to $\gamma_0 = 80\,\mathrm{dB}$ and the path loss exponent to $\alpha = 3$ [8].

Since LMP is a novel problem, to evaluate the performance of the proposed algorithm *lmpAlg* for the weighted data loss minimization problem, we introduce a benchmark heuristic *greAlg*. The algorithm first discrete the monitoring area into a series of hovering locations, then expands the path by selecting the location with the most urgent remaining time until the energy of the UAV is exhausted or all sensors have been visited. We mainly contrast our algorithm to *appAlg* [5] and *cAlg* [1]. *appAlg* constructs a closed tour for an energy-constrained UAV to maximize the volume of collected data. *cAlg* iteratively expand the data collection tour and then prune the path according to the data collection efficiency ratio. For each parameter setting, we obtain the average results from 50 instances.

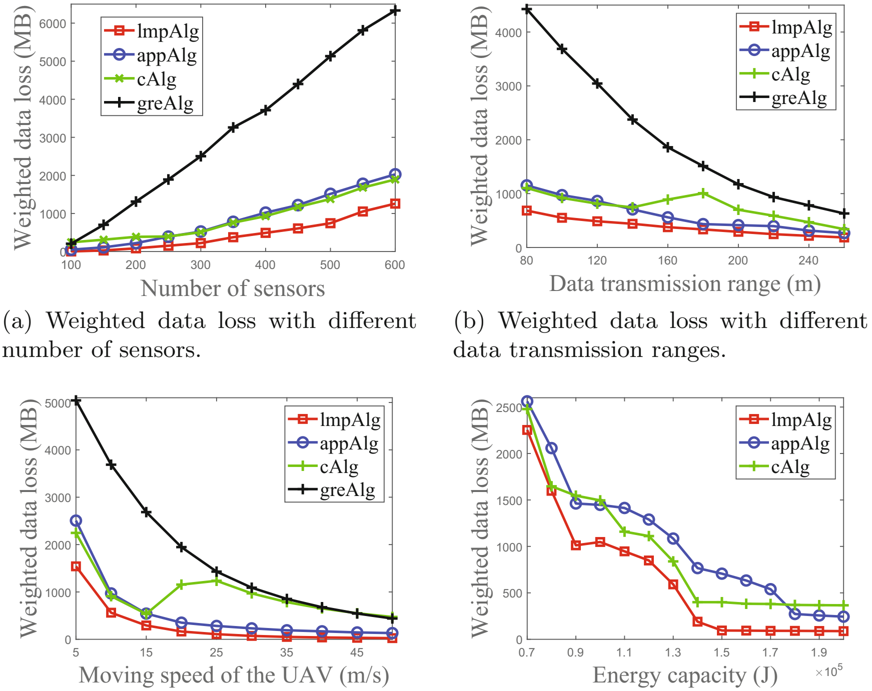
5.2 Performance Evaluation of Different Algorithms

Following, we investigate the performance impact of the number of sensors, the data transmission range, the moving speed, and the energy capacity of the UAV.

We first vary the number of sensors in the WSN from 100 to 600. As shown in Fig. 1(a), *lmpAlg* significantly outperforms other algorithms. The average weighted data loss of *lmpAlg* is about 37% to 96% less than those three algorithms. Figure 1(a) also shows that the weighted data loss of algorithms is proportional to the number of sensors in the network. The reason is that with the increase in sensors, the WSN generates more data simultaneously. The UAV can not timely collect data from some sensors, which will incur more data loss.

Then we vary the data transmission range from 80 m to 260 m, with 400 sensors distributed in the network randomly. Figure 1(b) depicts that the weighted data loss of *lmpAlg*, *appAlg* and *greAlg* monotonically declines with the data transmission range increase. The reason is that a longer data transmission range makes more sensors covered by the UAV at the same hovering location, thus leading to fewer weighted data loss. However, as shown in Fig. 1(b), the weighted data loss of *cAlg* increases when the data transmission range is between 140 m and 180 m. The reason is that *cAlg* will optimize the data collection path only when the energy consumption of the data collection tour exceeds the battery capacity. When the data transmission range is larger than 140 m, the UAV energy consumption to collect data from sensors is small enough that the total energy consumption of the UAV will not exceed the energy limit, so Alg1 does not optimize the path. Moreover, when the data transmission range is larger than 180 m, the UAV can complete the data collection task by hovering at a few hovering locations. So the weighted data loss decreased again because of a shorter moving distance.

Subsequently, we randomly deploy 400 sensors in the WSN and increase the value of the UAV flying speed from 5 m/s to 50 m/s. Figure 1(c) shows that the weighted data loss of *lmpAlg*, *appAlg* and *greAlg* decline against the increase



(a) Weighted data loss with different number of sensors.

(b) Weighted data loss with different data transmission ranges.

(c) Weighted data loss with different moving speeds.

(d) Weighted data loss with different energy capacities of the UAV.

Fig. 1. Performance evaluation of algorithms.

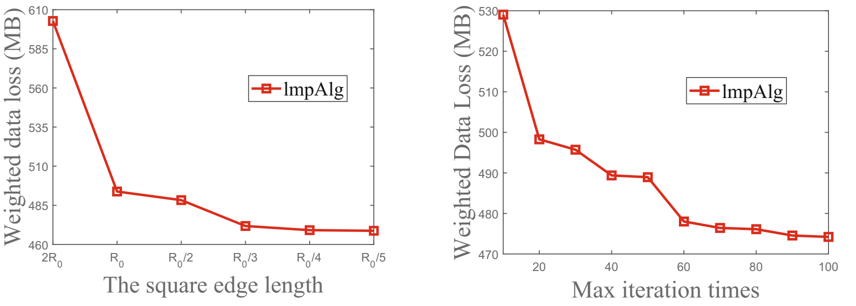
of the UAV moving speed. The reason is that with a faster speed, the UAV consumes less time moving between sensors and can collect data in a more timely manner, which helps reduce data overwritten. However, when the moving speed of the UAV is between 15 m/s and 20 m/s, the volume of weighted data loss of *cAlg* temporarily increases, for a similar reason in Fig. 1(b).

Finally, we vary the energy capacity from 0.7×10^5 J to 2×10^5 J, with 200 sensors distributed in the network randomly. Figure 1(d) depicts the weighted data loss monotonically declines with the increase of the UAV energy capacity. A UAV with low energy capacity needs to return to the BS frequently to recharge, which prevents it from accessing sensors on time. In addition, *greAlg* has tens of times more data loss than the others, so it does not appear in the diagram.

5.3 Impacts of Parameters on the Performance of Algorithm

In this section, we investigate the impact of parameters on performance in *lmpAlg* by randomly deploying 400 sensors in the network.

The edge length l of discretization squares determines the number of potential hovering locations (in step 1 of *Algorithm* 1), which will impact the algorithm



(a) Weighted data loss with different square edge lengths of *lmpAlg*.

(b) Weighted data loss with different iteration times of *lmpAlg*.

Fig. 2. The impact of simulation parameters on the performance of lmpAlg.

performance. Specifically, we decrease the length l of squares from $2R_0$ to $\frac{R_0}{5}$, and present the results in Fig. 2(a). The weighted data loss decreases from 603 MB to 494 MB when the square edge length l declines from $2R_0$ to R_0. As shown in theorem 1, when $l = 2R_0$, the UAV can only hover above sensors, while the value of l ranges from R_0 to $\frac{R_0}{5}$, the UAV can hover in the neighbors of sensors. In addition, the smaller l is, the finer the division of hovering location is. Consequently, the UAV can trade off the moving distance and hovering duration. Figure 2(a) also shows that a further decrease in the l will not lead to a better result in the same WSN.

Then we investigate the impact of the max iteration times $\mathcal{M}$ in data collection path optimization (in step 2 of *Algorithm* 2). Figure 2(b) depicts that the weighted data loss decreased from 529 MB to 478 MB against the max iteration times growth from 10 to 60. Because more iterations mean more optimization schemes, the data collection path is more likely to converge to the best result.

6 Conclusion

This paper investigates the Loss Minimization Problem (LMP), which minimizes the weighted data loss in a UAV-enabled WSN by scheduling the UAV hovering locations and durations, subject to the UAV energy capacity. We first devise a novel one-to-many data collection scheme by adopting the OFDMA technique and then discrete the number of infinite potential hovering locations into finite. We instead design efficient heuristic and approximation algorithms for it and evaluate the performance of proposed algorithms through extensive experimental simulations. Simulation results show that the proposed algorithms are promising.

Acknowledgments. This research is supported in part by the National Natural Science Foundation of China (62072320), the National Key R&D Program of China (2020YFB0704502), the Natural Science Foundation of Sichuan Province (2022NSFSC0569), the Key R&D Program of Sichuan Province (22ZDYF3599), and the Cooperative Program of Sichuan University and Yibin (2020CDYB-30).

References

1. Chen, M., Liang, W., Li, J.: Energy-efficient data collection maximization for UAV-assisted wireless sensor networks. In: 2021 19th IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–7. IEEE (2021)
2. Engels, C., Manthey, B.: Average-case approximation ratio of the 2-opt algorithm for the TSP. Oper. Res. Lett. **37**(2), 83–84 (2009)
3. Englert, M., Röglin, H., Vöcking, B.: Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP. Algorithmica **68**(1), 190–264 (2013). https://doi.org/10.1007/s00453-013-9801-4
4. Guo, J., Wang, T., He, Y., Jin, M., Jiang, C., Liu, Y.: TwinLeak: RFID-based liquid leakage detection in industrial environments. In: 2019 38th IEEE International Conference on Computer Communications (INFOCOM), pp. 883–891. IEEE (2019)
5. Li, Y., et al.: Data collection maximization in IoT-sensor networks via an energy-constrained UAV. IEEE Trans. Mob. Comput. (2021)
6. Liang, Y., et al.: Nonredundant information collection in rescue applications via an energy-constrained UAV. IEEE Internet Things J. **6**(2), 2945–2958 (2018)
7. Liu, T., Wu, B., Zhang, S., Peng, J., Xu, W.: An effective multi-node charging scheme for wireless rechargeable sensor networks. In: 2020 39th IEEE International Conference on Computer Communications (INFOCOM), pp. 2026–2035. IEEE (2020)
8. Luo, C., Satpute, M.N., Li, D., Wang, Y., Chen, W., Wu, W.: Fine-grained trajectory optimization of multiple UAVs for efficient data gathering from WSNs. IEEE/ACM Trans. Networking **29**(1), 162–175 (2020)
9. Mozaffari, M., Saad, W., Bennis, M., Debbah, M.: Mobile internet of things: can UAVs provide an energy-efficient mobile architecture? In: 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2016)
10. Qin, Z., Li, A., Dong, C., Dai, H., Xu, Z.: Completion time minimization for multi-UAV information collection via trajectory planning. Sensors **19**(18), 4032 (2019)
11. Qin, Z., et al.: Task selection and scheduling in UAV-enabled MEC for reconnaissance with time-varying priorities. IEEE Internet Things J. **8**, 17290–17307 (2021)
12. Samir, M., Sharafeddine, S., Assi, C.M., Nguyen, T.M., Ghrayeb, A.: UAV trajectory planning for data collection from time-constrained IoT devices. IEEE Trans. Wirel. Commun. **19**(1), 34–46 (2020)
13. Wang, P., Yu, R., Gao, N., Lin, C., Liu, Y.: Task-driven data offloading for fog-enabled urban IoT services. IEEE Internet Things J. **8**(9), 7562–7574 (2021)
14. Wu, Q., Liu, L., Zhang, R.: Fundamental trade-offs in communication and trajectory design for UAV-enabled wireless network. IEEE Wirel. Commun. **26**(1), 36–44 (2019)
15. Xiang, C., et al.: Reusing delivery drones for urban crowdsensing. IEEE Trans. Mob. Comput. 1–17 (2022). (Early Access)
16. Xu, W., et al.: Minimizing the deployment cost of UAVs for delay-sensitive data collection in IoT networks. IEEE/ACM Trans. Netw. **30**, 812–825 (2021)
17. Zhan, C., Zeng, Y.: Completion time minimization for multi-UAV-enabled data collection. IEEE Trans. Wirel. Commun. **18**(10), 4859–4872 (2019)

Robust Adaptive Cubature Kalman Filter for Attitude Determination in Wearable Inertial Sensor Networks

Hongkai Zhao[1,2], Huihui Wang[3], Zhelong Wang[1,2], Long Liu[3], and Sen Qiu[1,2](✉)

[1] School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, China
qiu@dlut.edu.cn

[2] Key Laboratory of Intelligent Control and Optimization for Industrial Equipment of Ministry of Education, Dalian University of Technology, Dalian 116024, China

[3] Dalian Neusoft University of Information, Dalian 116024, China

Abstract. Attitude analysis and recognition can be applied in wearable computing for medical assistance, motor-function assessment and dexterous human-robot interaction. The main problems, however, are serious drift and instability during traditional motion measurement fusion methods due to the high dynamic complexity of limb movements. To the best of our knowledge, it is the first attempt to employ an adaptive robust cubature Kalman filtering algorithm in the human attitude analysis based on wearable inertial sensors with time-varying state-process noise. Experiment results show that the adaptive robust CKF algorithm based on quaternion and gyroscope error modeling proposed in this paper can solve motion attitude solution. Lastly, we compare our method with CKF and EKF algorithm, the proposed algorithm can effectively improve the precision of attitude analysis.

Keywords: Cubature Kalman filter · Body sensor network · Sensor fusion · Adaptive factor · Robust filter

1 Introduction

It is well known that magnetic and inertial measurement units (MIMUs), comprised of a tri-axial accelerometer, gyroscope, and magnetometer, are used to track the displacement and orientation of a rigid body in real-time. Because of their lightweight, small size, and low cost properties, MIMUs have been used extensively as an ideal tool in target tracking, unmanned vehicle navigation, robotics, and human motion capture [1,2]. However, each of these sensors have limitations and may yield poor results when they are used alone [3]. In almost all applications, whether walking trajectory tracking or joint angle measurement, sensor fusion algorithms (SFA) [4–6] may be the first option.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 130–141, 2022.
https://doi.org/10.1007/978-3-031-19214-2_11

Among different sensor fusion algorithms for attitude estimation, the simplest and widely used one are the Kalman Filter(KF)-based attitude estimation methods [7]. For decades, some extensions of the KF are proposed since many practical problems do not satisfy the linear hypothesis. The extended Kalman filter (EKF) become an important tool to tackle most of the filtering problems [8–10]. However, the EKF may become unstable when the systems are strongly nonlinear, since the linearization causes a large truncation error. Unscented Kalman filter (UKF) performs better than EKF in terms of robustness and speed of convergence, but suffers from computational complexity (sometimes referred to as the "curse of dimensionality") [11,12]. To overcome these deficiencies, Arasaratnam and Haykin [13] proposed the so-called cubature Kalman filter (CKF), which offers a numerically stable solution with a low computational effort to the nonlinear state estimation problem. CKF has been used in many applications such as, spacecraft attitude estimation [14], underwater target tracking [15,16], and power system dynamic state estimation [17,18]. Stochastic stability and convergence of CKF were reported in [19] and [20], where it is shown that under mild assumptions, the estimation error is bounded.

Although the popular uses of inertial sensors in motion capture, technique challenges still exist in detecting dynamic motion of human limbs. As a matter of fact, the covariance matrix of process noise may be amplified due to the use of low-cost inertial sensors and errors of gyroscopes in the applications of magnetic and inertial measurement units. Meanwhile, the output errors of gyroscopes increases with the scale factor errors during the violent maneuvering, which will lead to a prior uncertainty, the filter may face the situation that the measurement noise model is unknown [21,22], and the system model cannot be described accurately. As a result, the accuracy of the sensor fusion system will decrease. To reduce the effect of unknown measurement noise on the filtering results, various adaptive robust filtering algorithms such as adaptive CKF (ACKF) [26,27], improved CKF (ICKF) [28], and robust CKF (RCKF) [29,30] have been proposed.

However, as far as we know, there is no literature on the application of adaptive cubature Kalman filter to human posture calibration and motion analysis. In this study, therefore, a human attitude data fusion algorithm with the adaptive CKF (ACKF) is developed. In this way, the accurate and robust human attitude estimation in complex noise environment using simple wearable sensors are expected.

In order to deal with time-varying state-process noise, a adaptive factor method was utilized in this paper. Then, by using Huber-based robust filtering proposed in [28] as a starting point, we derive an enhanced version of the CKF for attitude estimation. Compared with the traditional algorithms EKF and UKF, the proposed ACKF generally has the following advantages:

1) Moderate computational complexity.
2) Higher stability, convergence, and reliability of filtering.
3) Strong robustness with respect to model parameters.
4) Lower sensitivity to random noise and initial state statistical properties.

5) Strong ability to track the abrupt state and maintain this ability when the filter reaches steady state.

The organization of this paper proceeds as follows. Section 2 introduces the conversion of quaternions to Euler angles, and the initial attitude calibration. Then, the adaptive robust CKF with quaternions including the combination of robust estimation methodology with the adaptive factor based on the estimated covariance matrix of the predicted residuals in CKF framework and the former combining with multiple fading factors CKF are developed in Sect. 3. The effectiveness of the proposed filtering algorithm for human attitude estimation is investigated in Sect. 4. Finally, some conclusions are drawn in Sect. 5.

2 Attitude Estimation System

The commonly used attitude matrix representation methods include directional cosine method, Euler angle transformation method and quaternion method. Among these, a more viable choice is often quaternion [31], which is a four-component object with three vectors and a scalar that captures the current attitude. The primary advantage of using the quaternion is that they are less computationally intensive, facilitate interpolation operations, and also avoid the gimbal lock problem, thus providing many flexible operations.

Define quaternion as:

$$q = q_0 + q_1 i + q_2 j + q_3 k \tag{1}$$

where q_0, q_1, q_2 and q_3 are real numbers, i, j and k are imaginary units, $i^2 = j^2 = k^2 = -1$. Given the unit-quaternion constraint $\|q\| = 1$.

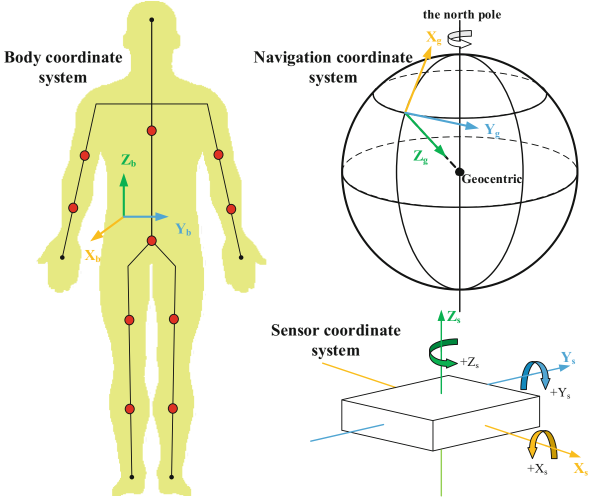


Fig. 1. The definitions of three coordinate systems in this study

For the purpose of equestrian motion capture, three coordinate systems which are navigation coordinate system, body coordinate system and sensor coordinate system are defines as Fig. 1.

2.1 Attitude Update Model

The quaternion differential equation is used to update the attitude of the carrier in the navigation coordinate system [32]:

$$\dot{q} = \frac{1}{2}\mathit{\Omega}\left(\omega_k^s\right) q, \;\; \mathit{\Omega}\left(\omega_K^s\right) = \begin{bmatrix} 0 & -\hat{\omega}_x^s & -\hat{\omega}_y^s & -\hat{\omega}_z^s \\ \hat{\omega}_x^s & 0 & \hat{\omega}_z^s & -\hat{\omega}_y^s \\ \hat{\omega}_y^s & -\hat{\omega}_z^s & 0 & \hat{\omega}_x^s \\ \hat{\omega}_z^s & \hat{\omega}_y^s & -\hat{\omega}_x^s & 0 \end{bmatrix} \tag{2}$$

where $\hat{\omega}_k^s = [\hat{\omega}_x^s, \hat{\omega}_y^s, \hat{\omega}_z^s]^T$ indicates the corrected measurement of the angular velocity projection of the gyroscope in sensor coordinate system. In view of (2), the quaternion can be solved by solving the differential equation, and then the posture of human body can be calculated.

Quaternions can be used to represent rotation relationships between coordinates, we need a more intuitive euler angle representation of these rotations. The relation between quaternion and Euler angle can be calculated by using the relation between direction cosine matrix and quaternion and Euler angle [28]:

$$\gamma = \arctan\left(\frac{2(q_2q_3 + q_0q_1)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}\right) \tag{3}$$

$$\theta = \arcsin\left(-2(q_1q_3 - q_0q_2)\right) \tag{4}$$

$$\varphi = \arctan\left(\frac{2(q_1q_2 + q_0q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}\right) \tag{5}$$

where γ, θ and φ indicate roll angle, pitch angle and taw angle respectively.

2.2 Attitude Estimation Model

Let the state-space model of the attitude estimation system be expressed as [28]

$$\begin{cases} x_{k+1} = f\left(x_k\right) + w_k \\ z_{k+1} = h\left(x_k\right) + v_k \end{cases} \tag{6}$$

where $x_k \in \mathcal{R}^n$ denotes the state at time k and $f(\cdot)$ is the nonlinear state function. $z_k \in \mathcal{R}^m$ represents the measurement vector and $h(\cdot)$ is the nonlinear measurement function. The process and measurement noise are represented as $w_k(0, Q_k)$ and $v_k(0, R_k)$, which are zero-mean Gaussian distributed with covariances $Q_k \geq 0$ and $R_k \geq 0$ respectively.

In this paper, we choose quaternions and as the bias of gyroscope the variables, one has,

$$x_k = \left[q_k, b_{\omega k}\right]^T \tag{7}$$

where q_k and $b_{\omega k}$ represent the quaternions and the bias of gyroscope at time k, respectively.

Gravity acceleration vector can observe the errors of pitch and roll of carrier, and the magnetic field vector can observe the errors of yaw. So we select the system measurement quantity based on gravity acceleration vector and magnetic field vector.

$$z(t) = \left(a_x^s(t), a_y^s(t), a_z^s(t), \varphi_m(t)\right)^T \tag{8}$$

where $a_x^s(t)$, $a_y^s(t)$ and $a_z^s(t)$ are the measurements of 3-axis accelerations, $\varphi_m(t)$ indicates the yaw angle by projection of the output value of magnetometer. In this paper, the system noise covariance matrix $Q(t)$ are regard as time-varying:

$$Q(0) = \begin{bmatrix} \sigma_q I_{4\times4} & 0 \\ 0 & \sigma_w I_{4\times4} \end{bmatrix} \tag{9}$$

where σ_q is the variance of the angle calculated using acceleration and magnetic field strength, σ_w is the angular rate variance of the three axes. In this paper, an adaptive random extinction factor method is employed to deal with time-varying covariance matrix $Q(t)$.

3 Design of Attitude Data Fusion Algorithm

This section presents the new adaptive robust attitude data fusion algorithm.

3.1 Data Fusion Scheme Design

The CKF uses the third-order volume rule and the numerical integration to approximate the Gauss weighted integration. The core problem of CKF is to solve the integral whose integral form is nonlinear function multiply Gauss density function. The weighted sum of a set of equal weight volume points is used instead of the weighted Gauss problem. CKF is similar to UKF, but it has more rigorous theoretical analysis and uses less sampling points than UKF. The EKF method was utilized in [24] to fuse the wearable sensons. To the best of our knowledge, the CKF has not used in human attitude analysis based on wearable inertial sensors.

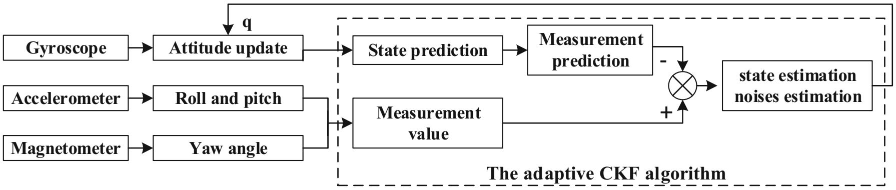


Fig. 2. The framework of wearable sensor fusion algorithm

In order to integrate the advantages of gyroscope, accelerometer and magnetometer in solving attitude and make up for their respective shortcomings, this paper designs an attitude data fusion scheme based on the adaptive cubature Kalman filter algorithm (ACKF) as shown in Fig. 2.

3.2 Robust Adaptive Cubature Kalman Filter Algorithm

The ACKF includes the standard steps of initialization, time update, and measurement update, but an extra step of robust correction is interposed between the time and measurement update steps, lastly, the adaptive factor will be updated. The underpinning mathematics of these steps are as follows. The entire robust cubature Kalman filter algorithm with adaptive factor based on attitude estimation is presented as follows:

Algorithm 1: Adaptive Robust Cubature Kalman Filter

```
Input  : $a_x, a_y, a_z, \omega_x, \omega_y, \omega_z, m_x, m_y, m_z, q_{init}$
Output: $q_0, q_1, q_2, q_3$
Parameter initialization: $x_0$, $P_0$, $Q_0$, $R_0$ ;
Time update: Calculate the cubature points $\mathcal{X}_{i,k-1|k-1}$ and bring the
  cubature points into $f(\cdot)$;
Measurement update: Recalculate the cubature points $\mathcal{X}_{i,k|k-1}$ bring the
  cubature points into $h(\cdot)$. Then, update the innovation-correlative
  covariance matrix $P_{zz,k|k-1}$ and cross-correlation covariance matrix
  $P_{xz,k|k-1}$.;
State update: Calculate the Kalman gain $K$. Update the estimate of the
  state vector $\hat{x}_k$ according to $\hat{x}_k = \hat{x}_{k|k-1} + K(z_k - \hat{z}_{k|k-1})$;
Adaptive factor update: Update the adaptive factor update according to
  $\hat{Q}_k = Q_{k-1}/\sqrt{\mu_k}$ ;
```

Initialzation

$$\hat{x}_0 = E(x_0),\ P_0 = E\left[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\right] \tag{10}$$

Time Update
Calculate the cubature points:

$$\mathcal{X}_{i,k-1|k-1} = S_{k-1|k-1}\xi_i + \hat{x}_{k-1|k-1},\ i = 1, 2, \cdots, 2n_x \tag{11}$$

where $P_{k-1|k-1} = S_{k-1|k-1}S^T_{k-1|k-1}$ and $\xi_i = \sqrt{n_x}[1]_i$ represents the ith cubature point.

Bring cubature points into the nonlinear function:

$$X_{i,k|k-1} = f(\mathcal{X}_{i,k-1|k-1}),\ i = 1, 2, \cdots, 2n_x \tag{12}$$

Prediction of the state:

$$\hat{x}_{k|k-1} = \frac{1}{2n_x}\sum_{i=1}^{2n_x} X_{i,k-1|k-1} \tag{13}$$

Evaluated the predicted error covariance:

$$P_{k|k-1} = \sum_{i=1}^{2n_x} X_{i,k|k-1}X_{i,k|k-1}^T - \hat{x}_{k|k-1}\hat{x}_{k|k-1}^T + \hat{Q}_{k-1} \tag{14}$$

Measurement Update

The Cholesky decomposition of $P_{k|k-1}$:

$$P_{k|k-1} = S_{k|k-1}S_{k|k-1}^T \tag{15}$$

Calculate the cubature points:

$$\mathcal{X}_{i,k|k-1} = S_{k|k-1}\xi_i + \hat{x}_{k|k-1},\ i = 1, 2, \cdots, 2n_x \tag{16}$$

Bring cubature points into the measurement nonlinear function:

$$Z_{i,k|k-1} = h(\mathcal{X}_{i,k|k-1}) \tag{17}$$

Estimate the predicted measurement at time k:

$$\hat{z}_{i,k|k-1} = \frac{1}{2n_x}\sum_{i=1}^{2n_x} Z_{i,k|k-1} \tag{18}$$

Innovation-correlative covariance matrix:

$$P_{zz,k|k-1} = \frac{1}{2n_x}\sum_{i=1}^{2n_x} Z_{i,k|k-1}Z_{i,k|k-1}^T - \hat{z}_{i,k|k-1}\hat{z}_{i,k|k-1}^T + R_k \tag{19}$$

Cross-correlation covariance matrix:

$$P_{xz,k|k-1} = \frac{1}{2n_x}\sum_{i=1}^{2n_x} X_{i,k|k-1}Z_{i,k|k-1}^T - \hat{x}_{k|k-1}\hat{z}_{k|k-1}^T \tag{20}$$

State Update

The Kalman gain at time k:

$$K = P_{xz,k|k-1}P_{zz,k|k-1}^{-1} \tag{21}$$

The estimated value at time k:

$$\hat{x}_k = \hat{x}_{k|k-1} + K(z_k - \hat{z}_{k|k-1}) \tag{22}$$

State error covariance estimate:

$$P_k = P_{k|k-1} - KP_{zz,k|k-1}K^T \tag{23}$$

Adaptive Factor Update

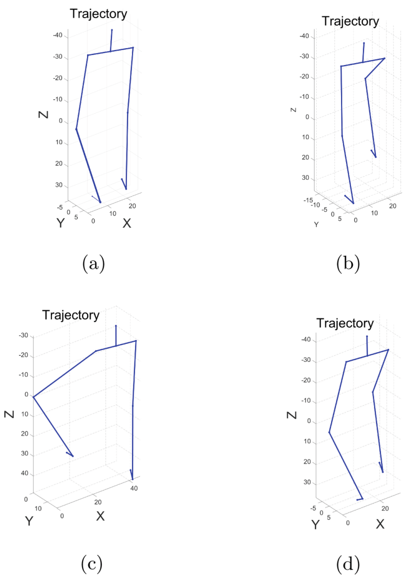
To improve the robustness of the adaptive filtering algorithm, a process noise scaling method is introduced here. The process noise covariance is adjusted by the adaptive factor and can be defined as [32]:

$$\hat{Q}_k = Q_{k-1}/\sqrt{\mu_k} \tag{24}$$

where $\mu_k = \frac{trace(H_k P_{k|k-1} H_k^T)}{trace(H_k \hat{P}_{k|k-1} H_k^T)}$ and H_k is the jacobian matrix of $h(\cdot)$ at time k.

4 Experiment Results and Algorithm Validation

In order to evaluate the accuracy of our method and the system performance, we utilize 10 sensors nodes on the chest, waist, upper arm, forearm, thigh and calf of human body respectively. Since the navigation coordinate system was set as the north-east-ground coordinate system, before the experiment began, participants stood facing due north with their hands hanging down naturally and kept perpendicular to the horizontal plane. Then, we will use those sensors to estimate the body posture when walking stairs, as shown in Fig. 3.



(a) (b)

(c) (d)

Fig. 3. Attitude estimation of inertial sensors

Table 1. The computation time of three methods

The method	EKF	CKF	ACKF
Computation time	1.432 ms	1.147 ms	1.238 ms

After the initial attitude calibration, we carried out the following experiment of attitude solution, and the measured value of quaternion changing with motion is shown in Fig. 4. During the whole attitude change period, there is no obvious phenomenon of leading or lagging, and the solution accuracy is ideal in time-varying dynamic environment, which meets the actual needs of real-time attitude tracking system.

We also compare our proposed method with existing mature algorithms (EKF and CKF). Table 1 compares the execution time of different algorithms, which is the average value of the algorithm after 1000 times of execution. It can be found that the running time of ACKF algorithm is about 1.238 ms, which is lower than that of EKF algorithm and slightly higher than that of CKF algorithm. Therefore, relatively speaking, the running time of the algorithm will be sacrificed to a certain extent when stable results are obtained. In addition, ACKF requires much less computation than EKF, which computes complex higher-order Jacobian matrices. Besides, the ACKF has the least computational complexity compared with the EKF and CKF algorithms.

In this paper, roll angle, pitch angle and yaw angle calculated by ACKF, EKF and CKF algorithms are compared, as shown in Fig. 5. It can be seen from the Fig. 5 that the yaw angle calculated by the ACKF algorithm is more stable than that calculated by EKF.

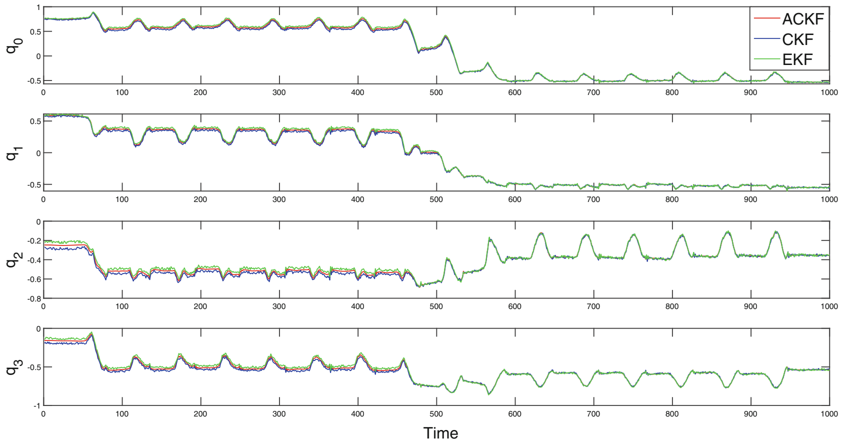


Fig. 4. The measured quaternion by adaptive cubature Kalman filter

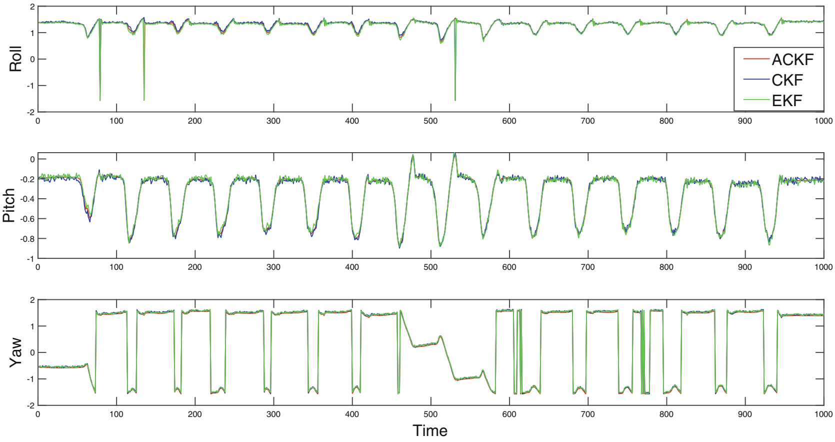


Fig. 5. The euler angle by adaptive cubature Kalman filter

5 Conclusion

In this paper, an adaptive quaternion-based ACKF estimator is developed for the human attitude fusion and estimation by applying the advantages of CKF. Above experimental results demonstrate that the proposed approach using the ACKF are accurate, reliable and robust for motion attitude solving. The proposed algorithm can effectively deal with misalignment errors, inherent nonlinearity in the measurement and model noises, noise-related, and measurement interference. In addition, the performance of the wearable fusion algorithm for attitude capture in the outdoor condition needs to be further validated. Thus, the following theoretical research needs to be further strengthened and the practical application of the algorithm needs to be improved.

Acknowledgements. This work was supported by National Natural Science Foundation of China under Grant No. 61873044 and 61803072, Fundamental Research Funds for the Central Universities under Grants No. DUT21YG125 and DUT20JC03, and Science and Technology Program of Liaoning Province (Grant no. 2021JH2/10300049).

References

1. Ahmad, N., Ghazilla, R.A.R., Khairi, N.M., Kasi, V.: Reviews on various inertial measurement unit (IMU) sensor applications. Int. J. Sig. Process. Syst. **1**(2), 256–262 (2013)
2. Qiu, S., et al.: Multi-sensor information fusion based on machine learning for real applications in human activity recognition: state-of-the-art and research challenges. Inf. Fusion **80**, 241–265 (2022)
3. Bhardwaj, R., Kumar, N., Kumar, V.: Errors in micro-electro-mechanical systems inertial measurement and a review on present practices of error modelling. Trans. Inst. Meas. Contr. **40**(9), 2843–2854 (2018)

4. Nazarahari, M., Rouhani, H.: Sensor fusion algorithms for orientation tracking via magnetic and inertial measurement units: an experimental comparison survey. Inform. Fusion **76**, 8–23 (2021)
5. Li, J., et al.: Real-time hand gesture tracking for human-computer interface based on multi-sensor data fusion. IEEE Sens. J. **21**(23), 26642–26654 (2021)
6. Liu, C., Zhu, H., Yu, L., Yin, H., Tang, X.: Performance evaluation of research laboratories with ecological theory and network data envelopment analysis. J. Clean. Prod. **327**, 129452 (2021)
7. Lefferts, E.J., Markley, F.L., Shuster, M.D.: Kalman filtering for spacecraft attitude estimation. J. Guidance, Control, Dyn. **5**(5), 417–429 (1982)
8. Simanek, J., Reinstein, M., Kubelka, V.: Evaluation of the EKF-based estimation architectures for data fusion in mobile robots. IEEE/ASME Trans. Mechatron. **20**(2), 985–990 (2014)
9. Li, J., et al.: Study on horse-rider interaction based on body sensor network in competitive equitation. IEEE Trans. Affect, Comput. (2019)
10. Dai, Z., Jing, L.: Lightweight extended Kalman filter for MARG sensors attitude estimation. IEEE Sens. J. **21**(13), 14749–14758 (2021)
11. Li, L., Xia, Y.: UKF-based nonlinear filtering over sensor networks with wireless fading channel. Inf. Sci. **316**, 132–147 (2015)
12. Li, L., Yu, D., Yang, H., Yan, C.: UKF for nonlinear systems with event-triggered data transmission and packet dropout. In: 2016 3rd International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS), pp. 37–42 (2016)
13. Arasaratnam, I., Haykin, S.: Cubature Kalman filters. IEEE Trans. Autom. Control **54**(6), 1254–1269 (2009)
14. Huang, W., Xie, H., Shen, C., Li, J.: A robust strong tracking cubature Kalman filter for spacecraft attitude estimation with quaternion constraint. Acta Astronaut. **121**, 153–163 (2016)
15. Sabet, M.T., Daniali, H.M., Fathi, A., Alizadeh, E.: Identification of an autonomous underwater vehicle hydrodynamic model using the extended, cubature, and transformed unscented Kalman filter. IEEE J. Oceanic Eng. **43**(2), 457–467 (2017)
16. Luo, J., Chen, Y., Wang, Z., Wu, M., Yang, Y.: Improved cubature kalman filter for target tracking in underwater wireless sensor networks. In: 2020 IEEE 23rd International Conference on Information Fusion (FUSION), pp. 1–8 (2020)
17. Sharma, A., Srivastava, S.C., Chakrabarti, S.: A cubature Kalman filter based power system dynamic state estimator. IEEE Trans. Instrum. Meas. **66**(8), 2036–2045 (2017)
18. Ling, L., Sun, D., Yu, X., Huang, R.: State of charge estimation of lithium-ion batteries based on the probabilistic fusion of two kinds of cubature Kalman filters. J. Energ. Storage **43**, 103070 (2021)
19. Wanasinghe, T.R., Mann, G.K., Gosine, R.G.: Stability analysis of the discrete-time cubature Kalman filter. In: 2015 54th IEEE Conference on Decision and Control (CDC), pp. 5031–5036 (2015)
20. Xu, B., Zhang, P., Wen, H., Wu, X.: Stochastic stability and performance analysis of cubature Kalman filter. Neurocomputing **186**, 218–227 (2016)
21. Yu, M.J.: INS/GPS integration system using adaptive filter for estimating measurement noise variance. IEEE Trans. Aerosp. Electron. Syst. **48**(2), 1786–1792 (2012)
22. Soken, H.E., Hajiyev, C., Sakai, S.I.: Robust Kalman filtering for small satellite attitude estimation in the presence of measurement faults. Eur. J. Control. **20**(2), 64–72 (2014)

23. Nazarahari, M., Rouhani, H.: A full-state robust extended Kalman filter for orientation tracking during long-duration dynamic tasks using magnetic and inertial measurement units. IEEE Trans. Neural Syst. Rehabil. Eng. **29**, 1280–1289 (2021)
24. Wang, Z., et al.: Motion analysis of deadlift for trainers with different levels based on body sensor network. IEEE Trans. Instrum. Meas. **70**, 1–12 (2021)
25. Qiu, S., et al.: Sensor network oriented human motion capture via wearable intelligent system. Int. J. Intell. Syst. **37**(2), 1646–1673 (2022)
26. Zhang, A., Bao, S., Bi, W., Yuan, Y.: Low-cost adaptive square-root cubature Kalman filter for systems with process model uncertainty. J. Syst. Eng. Electron. **27**(5), 945–953 (2016)
27. Lv, Y.W., Yang, G.H.: An adaptive cubature Kalman filter for nonlinear systems against randomly occurring injection attacks. Appl. Math. Comput. **418**, 126834 (2022)
28. Qiu, Z., Guo, L.: Improved cubature Kalman filter for spacecraft attitude estimation. IEEE Trans. Instrum. Meas. **70**, 1–13 (2020)
29. Qiu, Z., Qian, H., Wang, G.: Adaptive robust cubature Kalman filtering for satellite attitude estimation. Chin. J. Aeronaut. **31**(4), 806–819 (2018)
30. Guo, S., Chang, L., Li, Y., Sun, Y.: Robust fading cubature Kalman filter and its application in initial alignment of SINS. Optik **202**, 163593 (2020)
31. Lovren, N., Pieper, J.K.: Error analysis of direction cosines and quaternion parameters techniques for aircraft attitude determination. IEEE Trans. Aerosp. Electron. Syst. **34**(3), 983–989 (1998)
32. Ding, W., Wang, J., Rizos, C., Kinlyside, D.: Improving adaptive Kalman estimation in GPS/INS integration. J. Navig. **60**(3), 517–529 (2007)

Research on the Effect of BBR Delay Detection Interval in TCP Transmission Competition on Heterogeneous Wireless Networks

Weifeng Sun(✉), Kelong Meng, and Ailian Wang

School of Software Technology, Dalian University of Technology,
Dalian 116620, China
wfsun@dlut.edu.cn, klmeng@mail.dlut.edu.cn, alwang2021@163.com

Abstract. The bottleneck bandwidth and round-trip propagation time (BBR) algorithm effectively improves the network bandwidth utilization by its unique minimum delay and maximum bandwidth detection mechanism. However, with the development of 5G communication technology, whether the 10 s delay detection interval of BBR can meet the new high throughput and low latency heterogeneous network requirements needs to be studied. Therefore, based on the ns-3, this paper builds some scenarios to simulate the performance of BBR in wired, WiFi, and 5G networks. A spindle-shaped network topology is constructed to simulate the BBR competition. By modifying the delay detection interval of BBR to 5 s and 1 s, the competition among BBR streams with the same round-trip time (RTT), the competition among BBR streams with different RTTs, and the competition among BBR and other TCP congestion control algorithms (CCA) are simulated respectively. Then, a formula for calculating the delay detection interval is proposed. According to this formula, we propose a method to dynamically modify the delay detection interval. The method estimates the network state according to the change of RTT, and then calculates and updates the delay detection interval. Simulation results demonstrate that appropriately modifying the delay detection interval of BBR can alleviate the competition among BBR and other algorithms in heterogeneous wireless network.

Keywords: TCP-BBR · Heterogeneous network · Delay detection interval · Competition

1 Introduction

In recent years, with the rapid development of the Industrial Internet and 5G, a large number of terminal devices and equipment have been added to the network. At the same time, there are demands for new types of services and guarantees [1]. These changes make the network more complex and diversified, and make network services develop towards high throughput and low latency [2].

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 142–154, 2022.
https://doi.org/10.1007/978-3-031-19214-2_12

In order to meet the requirements of high throughput and low latency, BBR is proposed [3]. The BBR works through four stages, namely STARTUP, DRAIN, ProbeBW and ProbeRTT. BBR adjusts the end-to-end sending rate and congestion window (CWND) by periodically detecting the minimum RTT and maximum available bandwidth of the network to improve the network bandwidth utilization. But with the development of communication technology, the network status changes very rapidly. Therefore, the 10 s delay detection interval of BBR may affect its performance.

In the ultra-reliable and low latency communications scenarios of 5G network, the latency requirements are particularly strict. This requires CCA to be aware of the network status, and to take corresponding measures timely according to network status changes [4]. However, BBR with 10 s delay detection interval may not grasp the network status in time. For example, when the number of packets in the network increases, there will be buffer queuing delays. If the BBR fails to detect the increase of the delay in time because the delay detection interval of 10 s has not expired, the BBR will continue to send data packets to the network at a higher sending rate, which will cause packet loss and affect the network performance. However, most of the current studies improve the performance of BBR by modifying the detection bandwidth stage, and do not consider the impact of the delay detection interval mechanism on the network performance.

For the purpose of fully simulating the impact of different delay detection intervals on the performance of BBR, we use ns-3 to build spindle network topology, and simulate BBR in wired, WiFi, and 5G networks. The performance of BBR can be verified by simulation [5]. First, we modify the delay detection interval of BBR to 5 s and 1 s, and simulate the competition effect among BBR streams with the same RTT, the competition among BBR streams with different RTTs, and the competition among BBR, NewReno, Vegas, Westwood and Veno algorithms in wired and WiFi networks. Then, based on the analysis and simulation results, we propose a method to dynamically modify the delay detection interval according to the change of RTT. The simulation results show that BBR can quickly fill up the network bandwidth and improve the network bandwidth utilization. At the same time, the simulation results also demonstrate that dynamically modifying the delay detection interval can improve the performance of BBR and alleviate TCP transmission competition.

2 Related Work

Many scholars have conducted a lot of research on the performance of the BBR and proposed some improvement schemes. Zhang *et al.* [6] established different network communication models using ns-3. By analyzing the simulation data, they found that the actual throughput of BBR is better than Bic algorithm on high-latency, high-bandwidth networks, and proved that there is a fairness problem among BBR and other TCP CCAs. Furthermore, they confirmed that BBR streams with long RTT occupy more bandwidth than BBR streams with short RTT. Sun *et al.* [7] proposed the RFBBR algorithm. The RFBBR can guarantee the fairness of BBR streams with different RTTs. The simulation results show that RFBBR can significantly improve fairness compared with BBR and BBQ in the wireless network spindle topology scenarios. G. Kim *et al.* [8] proposed a BBR adversary congestion control identification model (OI-BBR) based on a decision tree classifier. In the evaluation experiment using Mininet, OI-BBR correctly identifies Cubic and performs different operations according to the congestion control behavior of competing algorithms in the network, thus improving the fairness between protocols by 1.31 times. Sun *et al.* [9] proposed a MFBBR, which has moderate fairness. The simulation results on the Mininet show that MFBBR can improve the fairness of BBR when it coexists with Westwood, and it also has better fairness compared with delay-based CCAs.

However, existing studies do not consider the delay detection interval of BBR, nor do they study the effect of different delay detection intervals on the TCP transmission competition. Therefore, in this paper, we simulate the effect of different BBR delay detection intervals on TCP transmission competition in heterogeneous wireless networks, and also propose a method to dynamically modify delay detection intervals to improve BBR performance.

3 BBR Delay Detection Interval

3.1 Mechanism of Delay Detection Interval

The state transition of BBR is shown in Fig. 1. In the STARTUP stage, BBR increases the sending rate with a larger coefficient (2.89). When it is detected that the sending rate does not increase for three consecutive times, BBR enters the DRAIN stage. During the DRAIN stage, BBR reduces the sending rate and drains excess packets. In the ProbeBW stage, BBR continuously adjusts the sending rate to detect available bandwidth of the network. Specifically, BBR cyclically adjusts the pacing rate coefficient with an array [1.25, 0.75, 1, 1, 1, 1, 1, 1]. Through this array, BBR can dynamically detect network bandwidth and improve bandwidth resource utilization. BBR enters the ProbeRTT stage every 10 s. In the ProbeRTT stage, BBR will detect the minimum delay of the network. After the ProbeRTT stage ends, the BBR decides to enter the STARTUP stage or the ProbeBW stage according to whether the network is fully loaded [3].

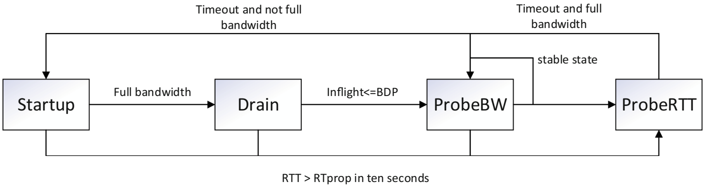

Timeout and not full bandwidth
Timeout and full bandwidth
stable state
Startup
Full bandwidth
Drain
Inflight<=BDP
ProbeBW
ProbeRTT
RTT > RTprop in ten seconds

Fig. 1. BBR state transition.

BBR enters the ProbeRTT stage every 10 s, which lasts for 200 ms. BBR will take the minimum RTT in this stage as RT_{prop}. In other stages, if BBR detects that the current RTT is less than RT_{prop}, it updates RT_{prop} to the current RTT, and updates the 10 s delay detection cycle. When the data packets are reduced, the queuing delay is reduced, and BBR can update RT_{prop} in time. However, when the number of packets increases, the queuing delay increases. At this time, the RT_{prop} is less than the current RTT, resulting in BBR mistakenly believing that the network state is good. Therefore, BBR will send a lot of packets to the network. However, a too short delay detection interval may reduce bandwidth utilization. This is because the CWND is only 4 packets in size during the ProbeRTT stage. Frequently entering the ProbeRTT stage will cause BBR's sending rate to drop and the network bandwidth cannot be effectively used.

When multiple BBR streams with the same RTT compete in the network, modifying the delay detection interval may not change the origin competition. This is because when the RTTs of the BBR streams are equal, each stream's minimum delay is same, and the BDPs calculated in the ProbeBW stage are equal, so that the dynamic fairness can be maintained. When there are multiple BBR streams with different RTTs competing in the network, reducing the delay detection interval may improve fairness. This is because when the delay detection interval is reduced, the BBR stream with long RTT passes through fewer rounds of ProbeBW stages in the same time, thereby reducing the amount of data packets sent. On the other hand, when BBR competes with other CCAs, such as Westwood, BBR always occupies a large amount of network bandwidth. BBR detects the available bandwidth due to failure to update RT_{prop} in time, and sends a large number of data packets to the network. That creates a queue at the bottleneck, which will cause additional queuing delay, and generate packet retransmissions. Westwood believes that it is in a congested state, and responds by reducing its Cwnd. Therefore, BBR will occupy a lot of network bandwidth.

3.2 Dynamic Delay Detection Interval

Through the analysis, we think that moderate delay detection interval will improve the network bandwidth utilization and increase the fairness of TCP transmission. Since the network state will change, keeping the delay detection interval constant may affect the performance of BBR. When the network changes from a congested state to a non-congested state, keeping the delay detection interval short will waste network bandwidth. Therefore, we design a method to dynamically modify the delay detection intervals according to the TCP transmission state. The delay detection interval can be calculated as Eq. (1)

$$I_{new} = I_{old} * F(Rtt, Cwnd, SendRate), \tag{1}$$

where I_{new} represents the new delay detection interval, I_{old} represents the previous delay detection interval. Rtt represents the round-trip delay of TCP connection, $Cwnd$ represents the congestion window of TCP connection, and $SendRate$ represents the sending rate of TCP sender. $F(Rtt, Cwnd, SendRate)$ is the function to calculate the delay detection interval coefficient. The function evaluates the current network congestion trend according to the changes of the three parameters Rtt, $Cwnd$ and $SendRate$ over a period of time, and then calculates a coefficient to adjust the delay detection interval. This function is only an open function and does not include a specific calculation process. According to this function, we propose a specific algorithm to calculate the delay detection interval. *Cwnd* and *SendRate* are directly related to the stage of the BBR. In the ProbeBW stage of BBR, the *Cwnd* and the *SendRate* are cyclically changed in synchronization with the sending rate coefficient array. Therefore their changing relationship cannot accurately display the network status. However, *Rtt* can accurately reflect the network state. RTT includes propagation delay, queuing delay and processing delay. The propagation delay and processing delay are determined by the current network transmission medium and the processing capability of the nodes, and cannot be changed with the network state. The queuing delay is caused by the queuing of data packets at intermediate nodes in the network, and its value directly reflects the current network state. When the queuing delay is large, the network may be congested. When the queuing delay is small, it means that the network is not congested.

Algorithm 1

```
Input: $RTT_{cur}$, $RTT_{last}$, $Count$
Output: $I_{new}$
// $RTT_{cur}$: The current RTT
// $RTT_{last}$: The last RTT
// $Count$: The number of times the RTT increases
When TCP sender receives ACK, Update $RTT_{last}$ and $RTT_{cur}$
while ProbeBW do
    if $RTT_{cur} > RTT_{last}$ then
        $Count = Count + 1$
        if $Count \geq 3$ then
            Reduce the interval when network is congested
            $I_{new} = 5$
            $Count = 0$
        else
            $I_{new} = 10$
    else
        Increase the interval when network is not congested
        $I_{new} = 10$
        $Count = 0$
Update $RTT_{last}$ and return $I_{new}$
$RTT_{last} = RTT_{cur}$
return $I_{new}$
```

Since the *Cwnd* and the *SendRate* fluctuate little during the ProbeBW stage of BBR, we have not considered them as the influencing factors for modifying the delay detection interval. We choose RTT, which is directly related to network congestion, as a factor affecting the delay detection interval. We also simplify the effect of RTT on the delay detection interval as the conversion between the delay detection interval of 10 s and 5 s. The pseudocode of dynamically modifying the delay detection interval method is shown in Algorithm 1.

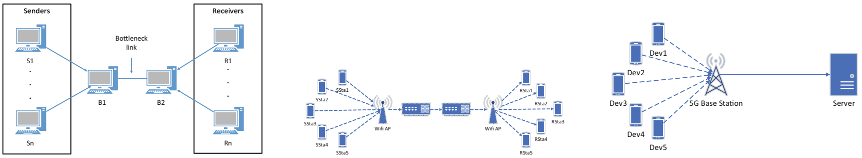
Lines 1–3 introduce the meaning of the relevant parameters, and lines 4–20 introduce the specific method of dynamically modifying the delay detection interval. Lines 4–11 modify the delay detection interval to 5 s. In the ProbeBW, when the sender detects that the RTT increases three times in a row, the sender considers that the network is congested and needs to update the minimum delay in time, so the delay detection interval is set to 5 s. Lines 12–20 reset the time delay detection interval to 10 s. When the sender detects that the current RTT is shorter than the previous RTT, the sender considers that the network congestion is relieved, and sets the delay detection interval to 10 s. The algorithm only increases the calculation of some parameters in the ProbeBW stage, so its time complexity is the same as that of BBR. This method can reduce the delay detection interval when the network is congested, so that the BBR can detect the minimum delay more frequently. It can also increase the delay detection interval of BBR, make full use of the characteristics of BBR, and improve the utilization rate of network bandwidth when the network status is good.

4 Simulation Setup and Result Analysis

In this section, by modifying the delay detection interval of BBR to 5 s and 1 s, the competition among BBR streams with the same RTT, the competition among BBR streams with different RTTs, and the competition among BBR and other TCP CCAs are simulated respectively. Then, the method of dynamically modifying the delay detection interval is simulated in the WiFi network. The ns-3 version is 3.33, and the 5G network is based on the NYU mmWave module [10]. The operating system is ubuntu 20.04.

4.1 Scenarios and Parameters

In the wired network, a spindle-shaped network topology with 5 leaf nodes on the left and right is built to simulate the competitive characteristics of BBR. For WiFi network, we build a wireless multi-hop spindle network topology. The characteristics and competition characteristics of BBR are simulated by allowing the access devices of the leaf nodes at both ends to communicate. The parameter settings are shown in Table 1. The network topologys are shown in Fig. 2.



(a) Wired Network. (b) WiFi Network. (c) 5G Network.

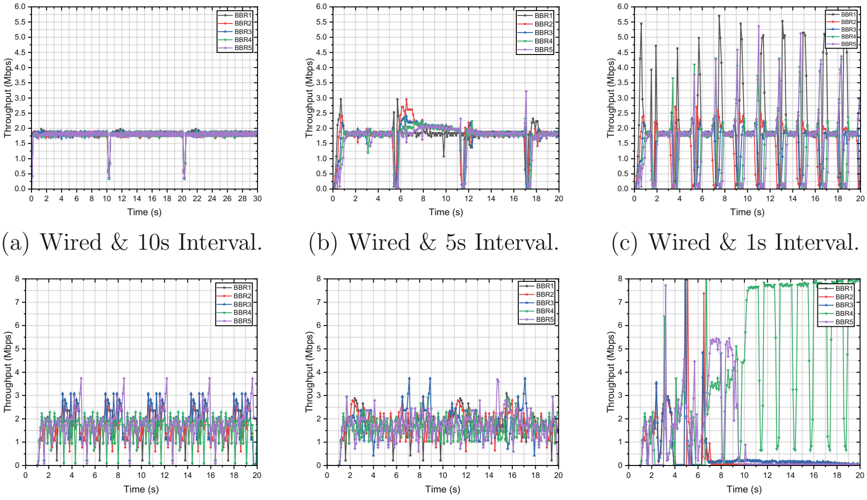
Fig. 2. Network topology

Table 1. Network parameter setting

Parameter	Value
Bottleneck bandwidth	10 Mbps
Bottleneck delay	30 ms
Access bandwidth	40 Mbps
Access delay	5/10/15/20/25 ms
Buffer size	128 KB

4.2 Simulation Results

In order to study the effect of different delay detection intervals on BBR, we simulate the competitive characteristic of BBR streams with the same RTT and different delay detection intervals in wired network and WiFi network respectively. The simulation time is set to 20 s, and the sender sends data traffic at 0 s. The simulation results are shown in Fig. 3.



(a) Wired & 10s Interval.

(b) Wired & 5s Interval.

(c) Wired & 1s Interval.

(d) WiFi & 10s Interval.

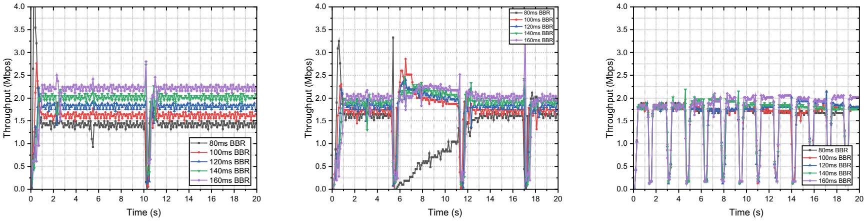
(e) WiFi & 5s Interval.

(f) WiFi & 1s Interval.

Fig. 3. Network bandwidth.

As can be seen from Fig. 3, when the BBR streams with the same RTTs compete for network bandwidth in wired network, the bandwidth of each stream is 1.8 Mbps regardless of the delay detection interval. However, as the delay detection interval becomes shorter, the stability and utilization of the bandwidth are gradually decreasing. In WiFi network, there is still a correlation between the bandwidth utilization and the delay detection interval. When the interval is 5 s, the bandwidth gap among each stream is reduced compared with when the interval is 10 s. But when the interval is 1 s, the bandwidth among each stream varies greatly. The results show that in WiFi network, the delay detection interval can affect the fairness and bandwidth utilization among BBR streams of the same RTT.

Then, we compare the fairness among BBR streams of different RTTs in the wired network. By setting different delay detection intervals, the influence of delay detection interval on fairness among BBR streams of different RTTs is simulated. The simulation results are shown in Fig. 4.



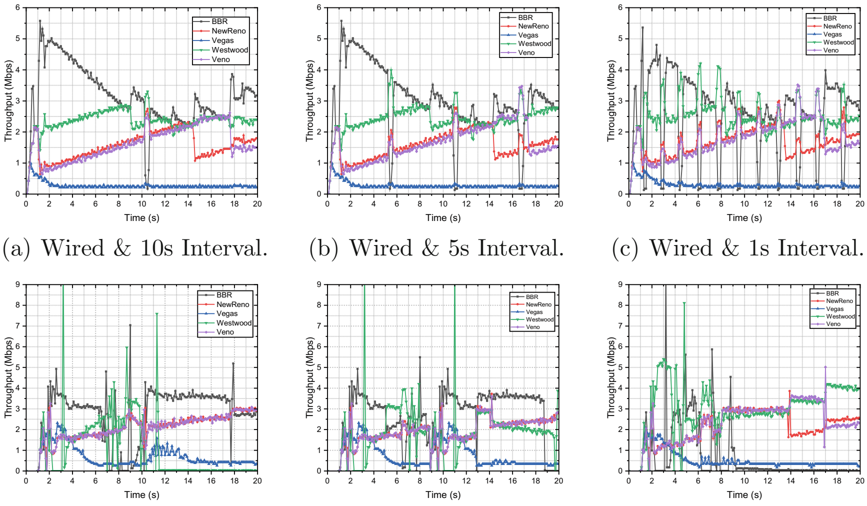
(a) Wired & 10s Interval. (b) Wired & 5s Interval. (c) Wired & 1s Interval.

Fig. 4. Network bandwidth.

In Fig. 4(a), When BBR streams with different RTTs compete for network bandwidth, the bandwidths of BBR streams are 1.4 Mbps, 1.6 Mbps, 1.8 Mbps, 2 Mbps, and 2.2 Mbps. The simulation results show that the fairness among BBR streams with different RTTs is poor. The specific performance is that the BBR stream with a long RTT occupies a large bandwidth, and the BBR stream with a short RTT occupies a small bandwidth. Meanwhile, with the change of the delay detection interval, the bandwidth gap among BBR streams of different RTTs is not improved. To make matters worse, as the delay detection interval decreases, the bandwidth utilization of each stream decreases, and the bandwidth fluctuation becomes larger.

For fairness among BBR and other CCAs, we simulate BBR, NewReno, Vegas, Westwood and Veno algorithms in wired and WiFi networks, respectively. The impact on the network bandwidth utilization and fairness of different CCAs is simulated by modifying the delay detection interval of BBR. The simulation results are shown in Fig. 5.

Figure 5 shows the network bandwidth of BBR with different delay detection intervals, NewReno, Vegas, Westwood and Veno algorithms in wired network and WiFi network. Among them, the BBR stream has the advantage in competition (the largest bandwidth), the competitiveness of Vegas is the weakest, and the fairness between NewReno and Veno is good. The simulation results show that BBR is more competitive than other TCP CCAs in wired network. Meanwhile, as the delay detection interval decreases, the bandwidth gap among BBR and other algorithms is alleviated in wired network. However, in WiFi network, when the delay detection interval of BBR is 1 s, BBR will gradually lose bandwidth until it reaches 0, which indicates that excessively reducing the delay detection interval will affect the bandwidth utilization of BBR.

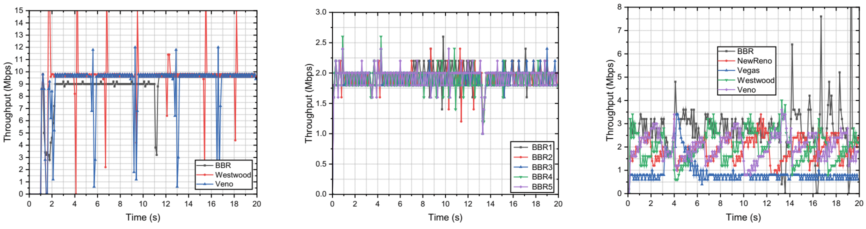


(a) Wired & 10s Interval. (b) Wired & 5s Interval. (c) Wired & 1s Interval.

(d) WiFi & 10s Interval. (e) WiFi & 5s Interval. (f) WiFi & 1s Interval.

Fig. 5. Network bandwidth.

In addition to these, we also simulate the competition of BBR in 5G network. Five access devices and 5G base station are set in a LAN, and the 5G base station is connected to the remote host through the wired network. Let access devices communicate with remote server host through 5G base station. The network topology is shown as Fig. 2(c) The simulation results are shown in Fig. 6.



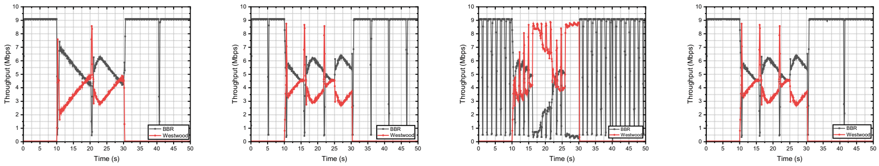
(a) 5G Network. (b) 5G Network. (c) 5G Network.

Fig. 6. Network bandwidth

Figure 6(a) shows the network bandwidth in 5G network. The bandwidth of BBR is 9 Mbps in the first 10 s and 9.6 Mbps in the last 10 s. The bandwidth of

Veno is maintained at 9.6 Mbps, and fluctuates up and down periodically. The bandwidth of Westwood fluctuates greatly. Although the bandwidth utilization of Veno and Westwood is slightly higher than that of BBR, the bandwidth stability is worse. Figure 6(b) shows that the bandwidth of BBR is relatively stable, staying between 1.75 Mbps and 2 Mbps. The results show that BBR can maintain good fairness in 5G network. Figure 6(c) shows the bandwidth of BBR, NewReno, Vegas, Westwood and Veno in 5G network. As can be seen from the figure, BBR has an advantage with a bandwidth of 3 Mbps. Vegas is the least competitive, with no bandwidth exceeding 1 Mbps. The fairness among NewReno, Veno and Westwood is better, and the bandwidth is basically equal. But BBR in 5G networks exhibits different competitive characteristics than in wired and WiFi networks. We will further study the 5G mmWave module to verify the correctness of BBR simulation in 5G network.

In order to evaluate the performance of the dynamically modifying the delay detection interval method, we set up the following scenario. In the WiFi network, two nodes communicate directly, and two TCP connections are set up. Among them, the BBR connection communicates for 50 s. Westwood connection starts sending data traffic at 10 s, and closes the connection at 30 s. When only the BBR stream is running, the network is not congested. When Westwood is running, the simulated network is congested. Westwood acts as background traffic in the simulation. By counting the bandwidth of two streams, the performance of dynamic delay detection interval method can be evaluated when the network state changes. The simulation results are shown in Fig. 7.



(a) 10s. (b) 5s. (c) 1s. (d) Dynamic.

Fig. 7. Network bandwidth

The simulation results show that when the network is not congested, the shorter the delay detection interval, the lower the network bandwidth utilization of BBR. When the network is congested (BBR and Westwood compete for network bandwidth), the 5 s delay detection interval enables BBR to update the network delay in a more timely manner, reducing the bandwidth gap with Westwood. When the BBR adopts the method of dynamically modifying the delay detection, the BBR can set a longer delay detection interval when the network is not congested, and reduce the delay detection interval when the network is congested. Compared with the results at 5 s, the simulation result of dynamic modification delay detection reduces the number of times of entering the ProbeRTT stage and improves the network bandwidth utilization.

4.3 Result Analysis

Through the above simulations, we summarize the fairness of BBR under different networks and different delay detection intervals. Fairness is divided into 4 levels, namely: excellent, good, medium and poor. The specific summary is shown in Table 2.

Table 2. Fairness comparison of different delay detection intervals

	sameRTT			diffRTT			diffCCA		
	10 s	5 s	1 s	10 s	5 s	1 s	10 s	5 s	1 s
Wired	Excellent	Excellent	Good	Medium	Medium	Medium	Medium	Good	Good
WiFi	Medium	Good	Bad	–	–	–	Good	Good	Good

In wired network, the delay detection interval of 5 s does not change the RTT fairness of BBR, but improves the fairness among BBR and other algorithms. This is because in the ProbeRTT stage, the CWND of BBR is the size of 4 data packets, and correspondingly, BBR will reduce the sending rate. With frequent detection of the minimum delay, BBR frequently reduces the sending rate, resulting in a decrease in bandwidth utilization. The delay detection interval of 1 s will break the RTT fairness among BBR streams with same RTT. In WiFi network, the inter-protocol fairness of BBR is not improved by different delay detection intervals. This may be because in the wireless network, due to the complex network channel, the BBR frequently changes between the ProbeRTT and the ProbeBW stage, which will cause network fluctuations, so the advantages of the BBR cannot be effectively utilized.

By dynamically modifying the delay detection interval, BBR can shorten the delay detection interval when the network is congested, and increase the delay detection interval when the network is good, thereby improving network bandwidth utilization and fairness. However, the results of this method are not significantly improved from the results at 5 s. This is mainly due to the hysteresis of the ProbeRTT stage. When the delay detection interval is modified, the next ProbeRTT stage must be passed through this interval. Since the delay cannot be detected immediately, the BBR cannot well grasp the network status. Meanwhile, we only dynamically set two parameters, 10 s and 5 s, in the simulation. Next, we will continue to adjust the method of dynamic detection, and dynamically calculate the delay detection interval according to some parameters such as RTT and CWND, instead of just switching it between 10 s and 5 s.

5 Conclusion

In this paper, we first analyze the effect of different delay detection intervals on BBR performance. Then a method to dynamically modify the delay detection interval is proposed. To validate the theoretical analysis, the competition among BBR streams with the same RTT, the competition among BBR streams with different RTTs, and the competition among BBR and other TCP CCAs are simulated respectively with BBR delay detection intervals of 10 s, 5 s and 1 s. We also simulate the performance of dynamically modifying the delay detection interval method in WiFi network. In addition, we also simulate the RTT competition and inter-protocol competition of BBR in 5G network. The simulation results show that properly reducing the delay detection interval of BBR will not affect the fairness among BBRs with the same RTT, and can enhance the inter protocol fairness of BBR, but too short delay detection interval will reduce the network bandwidth utilization. The simulation results of dynamically modifying the delay detection interval show that BBR can dynamically adjust the delay detection interval according to the network state, thereby improving the network bandwidth utilization and fairness of BBR.

In the future, we will set more complex network scenarios to simulate the performance of BBR. We will also improve the method of dynamically modifying the delay detection interval, and dynamically calculate the delay detection interval through deep learning or prediction method.

Acknowledgements. This work was supported by the National Key R&D Program of China (2018YFB1700100) and the CERNET Innovation Project (NGII20190801) and Fundamental Research Funds for the Central Universities under Grants (DUT21LAB115).

References

1. Yue, S., Ren, J., et al.: Efficient federated meta-learning over multi-access wireless networks. IEEE J. Sel. Areas Commun. **40**(5), 1556–1570 (2021)
2. Rene, S., Ascigil, O., et al.: A congestion control framework based on in-network resource pooling. IEEE/ACM Trans. Netw. **30**(2), 683–697 (2022)
3. Cardwell, N., Cheng, Y., et al.: BBR: congestion-based congestion control. Commun. ACM **60**(2), 58–66 (2017)
4. Harutyunyan, D., Shahriar, N., et al.: Latency and mobility-aware service function chain placement in 5G networks. IEEE Trans. Mob. Comput. **21**(5), 1697–1709 (2022)
5. Morato, D., Pérez-Gómara, C., et al.: Network simulation in a TCP-enabled industrial internet of things environment-reproducibility issues for performance evaluation. IEEE Trans. Ind. Inf. **18**(2), 807–815 (2022)
6. Zhang, H., Zhu, H., Xia, Y., et al.: Performance analysis of BBR congestion control protocol based on NS3. In: 2019 Seventh International Conference on Advanced Cloud and Big Data (CBD), pp. 363–368 (2019)
7. Sun, W., Jia, M., Zhang, G., et al.: RFBBR: a RTT faireness awared algorithm based on BBR. In: 2020 IEEE International Conference on Smart Internet of Things (SmartIoT), pp. 124–131 (2020)
8. Kim, G.H., Song, Y.J., Cho, Y.Z.: Improvement of inter-protocol fairness for BBR congestion control using machine learning. In: 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), pp. 501–504 (2020)

9. Sun, W., Jia, M., Wang, Z., et al.: MFBBR: an optimized fairness-aware TCP-BBR algorithm in wired-cum-wireless network. In: 29th IEEE Conference on Computer Communications(INFOCOM), pp. 171–176 (2020)
10. Mezzavilla, M., Zhang, M., et al.: End-to-end simulation of 5G mmWave networks. IEEE Commun. Surv. Tutorials **20**(3), 2237–2263 (2018)

BatMapper-Plus: Smartphone-Based Multi-level Indoor Floor Plan Construction via Acoustic Ranging and Inertial Sensing

Chuize Meng, Shan Jiang, Mengning Wu, Xuan Xiao, Dan Tao, and Ruipeng Gao(✉)

Beijing Jiaotong University, Beijing 100044, China
{mengchuize,hilljiang,meningwu,xiaoxuan,dtao,rpgao}@bjtu.edu.cn

Abstract. The lack of floor plans is one of the major obstacles to ubiquitous location-based services indoors. Dedicated mobile robots with high-precision sensors can scan and produce indoor maps, but the deployment remains low. Existing smartphone-based approaches usually adopt computer vision techniques to build the 3D point cloud, at the cost of extensive image collection efforts and the risk of privacy issues. In this paper, we propose BatMapper-Plus which constructs accurate and complete indoor floor plans by acoustic ranging and inertial sensing on smartphones. It employs acoustic signals to measure the distance to a nearby wall segment, and produces the accessible area by surrounding the building during walking. It also refines the constructed floor plan to eliminate scattered segments, and identifies connection areas including stairs and elevators among different floors. Extensive experiments in a teaching building and a residential building have shown our effectiveness compared with the state-of-the-art, without any privacy concerns and environmental limitations.

Keywords: Floor plan construction · Acoustic ranging · Inertial sensing · Smartphone

1 Introduction

Indoor location-based services (LBS) brings great convenience to our modern life, especially at large-scale hospitals, multi-level shopping malls, and underground parking lots. However, its deployment is still not yet pervasive, and one of its major obstacles is the lack of floor plans for indoor localization [1] and navigation [2].

At present, existing dedicated mapping systems [3] rely on mobile robots with cameras and other high-precision sensors to construct accurate indoor maps. However, such systems always cost expansively and are not wide-spread at large scale. With commodity smartphones, some AR/VR applications adopt computer

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 155–167, 2022.
https://doi.org/10.1007/978-3-031-19214-2_13

vision techniques [4] to build 3D point clouds for indoor objects, but images are affected by ambient light condition and risk privacy disclosure. Therefore, it is necessary to construct indoor floor plans without environmental supports and privacy concerns.

In a recent work [5], we have proposed BatMapper which uses acoustic signals to measure the distance for indoor mapping. Specially, it adopts a bilateral acoustic ranging mechanism that is effective to construct narrow corridors. SAMS [6] follows this work with higher accuracy by FMCW-based distance measurement. However, indoor environments are not limited to corridors, but also include wide areas such as rooms and lobbies. In such places, BatMapper can not produce satisfactory floor plans due to data association mistakes between wall segments and distance measurements. In addition, the constructed floor plan should be further adjusted to eliminate scattered points/segments, and augmented with connection areas to other floors.

In this paper, we propose BatMapper-Plus which is a smartphone-based indoor floor plan construction system by acoustic ranging and inertial sensing. It employs an unilateral ranging mechanism which measures the precise distance to a nearby wall segment, and produces the accessible area by walking around the room. It also refines scattered wall segments, and identifies connection areas to produce a multi-level floor plan. The system can build the indoor floor plan through a smart phone, which makes it have a low cost. Furthermore, the system does not need indoor images, so it is not affected by lighting conditions, and has high privacy.

Specially, we make the following contributions in this work:

- We explore a novel unilateral acoustic ranging method on smartphones. It emits acoustic signals for distance measurement to a side wall, thus users can simply surround the building to construct its floor plan with lightweight human efforts.
- We propose a map refinement algorithm to produce accurate and complete multi-level floor plans. It automatically adjusts and merges the scattered wall segments. It also detects and marks connection areas (e.g., stairs and elevators) on the map.
- We build a prototype and conduct extensive experiments in a teaching building and a residential building. Results have shown our improvements with around 2.8% in three experimental scenarios on F-score compared to BatMapper.

2 Overview

In this section, we present how the BatMapper works, explain its limitations during deployment in reality, and depict the overview of our BatMapper-Plus.

Background on BatMapper. BatMapper designs a two-pulse signals with linear frequency increasing sine waves and Hanning window reshaping for bilateral ranging on smartphones. It further explores a probabilistic evidence accumulation (PEA) method to associate the distance measurements to corresponding wall segments along the long corridor.

Limitation in BatMapper. 1) BatMapper measures the distance from echo signals which are reflected by two-side walls. This bilateral ranging mechanism is suitable in narrow areas such as a long corridor, but not designed for spacious spaces such as large rooms and lobbies. 2) BatMapper produces coarse floor plans which are composed of scattered points, and they are not consistent with the actual maps made of line segments. 3) BatMapper only generates the floor plan for one level, while modern buildings are always comprised of multiple levels with various connection areas.

BatMapper-Plus Overview. As Fig. 1 shows, BatMapper-Plus employs acoustic signals and inertial data as inputs. We adopt the Frequency Modulated Continuous Wave (FMCW) as the speaker's output signal, use two microphones (top/bottom) to receive the echo signal reflected from the side-wall, and calculate the distance between the wall and the smartphone. In addition, we collect the inertial data from smartphone to track the user and identify connection areas (e.g., stairs and elevators). Finally, we fuse distance measurements, walking trajectories and connection areas to construct and refine the multi-level indoor floor plan.

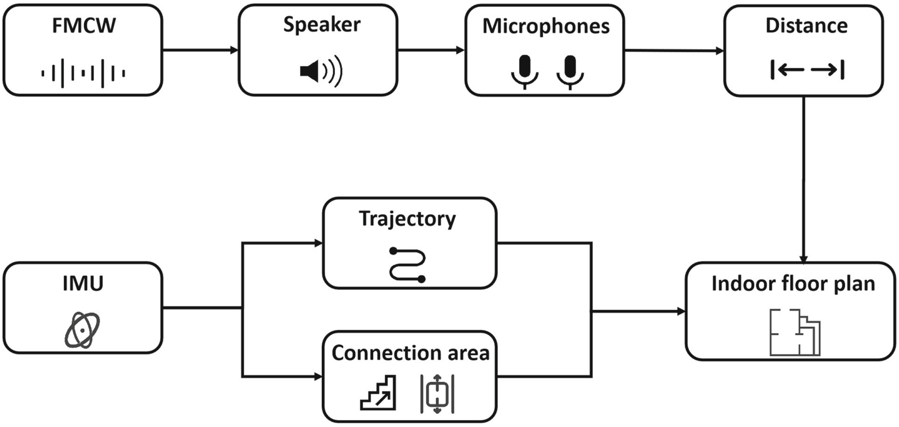


Fig. 1. Overview of BatMapper-Plus, which constructs multi-level indoor floor plan by acoustic ranging and inertial tracking.

3 Methods

In this section, we present the detailed design of BatMapper-Plus with three modules: unilateral acoustic ranging, map construction and refinement, and connection area detection.

3.1 Unilateral Acoustic Ranging

Acoustic Signal Design. The frequency of acoustic signal should balance both physical ability of smartphone and background noises. The sound frequency of smartphone is usually between $110\,Hz \sim 20\,KHz$, while the frequency of human voice is always less than $1\,KHz$. Through experiments, we found that a wider frequency range will make the echo peak more obvious. Furthermore, with the same emission energy, low-frequency sound spreads farther than high-frequency sound. Therefore, we generate the duration of acoustic signal as 3 ms and its frequency range is $8\,KHz \sim 16\,KHz$. Specially, We employ the Frequency Modulated Continuous Wave (FMCW) to produce the signal (the blue line in Fig. 2).

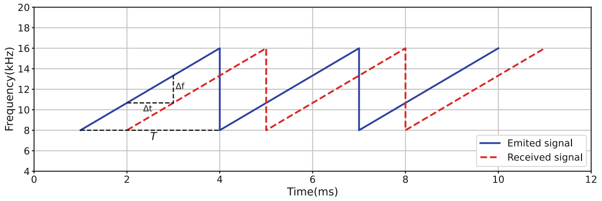


Fig. 2. Calculating distance by emited and received signals. (Color figure online)

Delay of Echo Signal. When the emitted acoustic signal meets the wall, its echo signal is reflected back and received by smartphone's microphone. At this time, the waveform of the received signal and the emitted signal are consistent, with a time delay Δt which is shown in Fig. 2. Specially, the time delay Δt is computed as:

$$\Delta t = \frac{\Delta f \cdot T}{f_{max} - f_{min}} \tag{1}$$

where f_{min} is the minimum frequency of the emitted signal, f_{max} is the maximum frequency of the emitted signal, and T is the duration of the emitted signal.

Distance to Wall. As shown in Fig. 3(a), when a user holds the smartphone horizontally during walking, Path 1 indicates the propagation path of the signal received by the top microphone, and Path 2 indicates the propagation path of the signal received by its bottom microphone. The propagation distance difference of received signals between by two microphones is the length l of the smartphone. Therefore, when the distance difference to the side-wall measured by two microphones is close to $\frac{1}{2}l$, the received signal is likely to be the echo reflected from the wall.

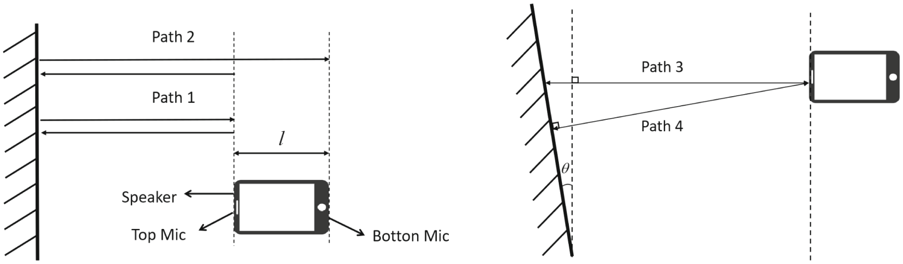
In this case, the distance d between the wall and the smartphone (by its top microphone in Fig. 3(a)) is expressed as:

$$d = \frac{1}{2}\Delta t \cdot v_{sound} \tag{2}$$

where v_{sound} is the sound propagation speed.

Through experiments, we found that when there is an angle θ between the mobile phone and the wall, the detected distance is the vertical distance from the top microphone to the wall (Path 4 in Fig. 3(b)), rather than the distance towards the top of the mobile phone (Path 3 in Fig. 3(b)). Therefore, when the mobile phone is not completely parallel to the wall, the accurate distance can still be obtained.

In addition, we use one-dimensional sound signal, which can only represent distance information and cannot distinguish indoor conditions. So this method has high privacy.



(a) Parallel to the wall

(b) Not parallel to the wall

Fig. 3. The position of the speaker and microphone, and the sound signal propagation path received by the microphone.

3.2 Map Construction and Refinement

Inertial Tracking. We use the dead-reckoning to track the walking user by his/her stride length, step count and heading orientation. 1) The normal stride length for an adult is about 60 cm, we use such value as default, and it can be customized by the outdoor trajectory with GPS [7]. 2) The step count is calculated by detecting the peak and valley values of vertical accelerations. Based on our experiments, we set the threshold of their difference as $3\,\text{m/s}^2$, and set the duration threshold between two steps as 400 ms in order to avoid errors caused by hand shaking. 3) In order to eliminate drifts from gyroscope and noises from magnetometer, we calculate the orientation by *gamerv* API of smartphone, which fuses accelerometer, gyroscope and magnetometer for robustness.

Door and Window Detection. We judge the existence of doors and windows by detecting the distance variation between the smartphone and the side-wall. When the change exceeds the threshold (20 cm in our system), we regard the point as a door/window. When the length of door/window is too short, we remove these points as outliers.

Floor Plan Refinement. The wall segment are positioned as scattered points on the map by distance measurements oriented to user's trajectory. Not only it contains outliers with extreme errors, but also such floor plan is not consistent with the actual map.

Intuitively, most walls are made of line segments, and their intersections could be identified by turning events of the walking user. After removing the detected doors and windows, we fit the rest points as line segments, i.e.,

$$f(x) = kx + b \tag{3}$$

where k is the gradient and b is the offset. In order to minimize the errors between a wall segment and the scattered points, we construct the objective function as:

$$e = \sum_{i}^{n}(kx_i + b - y_i) \tag{4}$$

where e represents the sum of errors, n represents the number of points on the wall, (x_i, y_i) is the ordinate of the i_{th} point by distance measurement. In order to minimize e, we calculate the partial derivatives of k and b respectively:

$$\frac{\partial e}{\partial k} = 2(\sum_{i}^{n}(kx_i + b - y_i)x_i) = 0 \tag{5}$$

$$\frac{\partial e}{\partial b} = 2(\sum_{i}^{n}(kx_i + b - y_i)) = 0 \tag{6}$$

thus

$$k = \frac{\sum_i^n (x_i y_i) - n\overline{xy}}{\sum_i^n (x_i)^2 - n\overline{x}^2} \tag{7}$$

$$b = \overline{y} - k\overline{x} \tag{8}$$

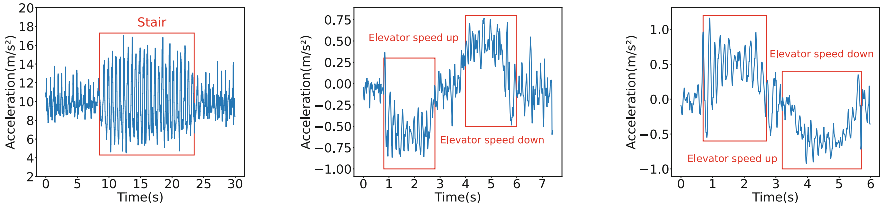
where $\overline{x}$ is the average value. Thus, we refine the reconstructed floor plan with wall segments, doors/windows, and corners.

3.3 Connection Area Detection

Since modern buildings are always multi-levels with different types of connection areas, we automatically identify stairs and elevators to associate each level of floor plan.

Stair Detection. Intuitively, there are always large variations in our accelerations when climbing stairs. In order to eliminate the influence of smartphone's attitude, we calculate the amplitude value of three-axis accelerations on smartphone (Fig. 4(a)). Next, we use a sliding window to dynamically detect the peaks and valleys along acceleration sequence. In order to avoid errors caused by hand shaking, the minimum time gap between peaks and valleys is set as 400 ms. Because when holding a mobile phone, it takes one second to take one step.

Elevator Detection. When an elevator starts or stops, the smartphone's acceleration along gravity direction varies accordingly, and it remains stable when the elevator moves at an uniform speed. As shown in Fig. 4(b) and Fig. 4(c), the vertical acceleration first decreases than increases when the elevator goes down, and verse versa. We adopt a sliding window of 3 s to detect its rising/sinking interval, and verify if such two periods are within a reasonable period (30 s in our system).



(a) Stair (b) An elevator sinks (c) An elevator sinks

Fig. 4. Three-axis acceleration variations at stairs and acceleration variations along gravity direction when an elevator rises and sinks.

4 Evaluation

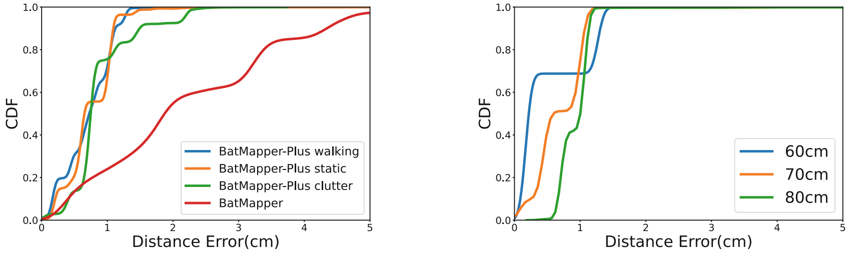
We have developed the prototype of BatMapper-Plus on Android Studio and installed it on MI 10S smartphone for data collection. Experiments are carried out in a residential building and a teaching building, both with multiple levels. The ground true distance and floor plan are measured by a laser rangefinder. Our evaluation includes three aspects, i.e., unilateral acoustic ranging, floor plan construction, and connection area detection.

4.1 Unilateral Acoustic Ranging

We carried out the acoustic ranging measurements indoors, with different user states and distances to wall.

User States. We test the accuracy with static users, walking users and a cluttered environment with many obstacles, and compare with the previous BatMapper. As shown in Fig. 5(a), our distance measurement error in the static state is close to that in walking state, with the median value around 0.7 cm and the maximum value less than 1.5 cm, both are obviously lower than the BatMapper. In a cluttered environment, the accuracy decreases slightly. This demonstrates our effectiveness of one-side ranging.

Distance to Wall. Three distances are tested at 60 cm, 70 cm and 80 cm to the same wall. As shown in Fig. 5(b), all distance measurement errors are less than 2 cm. In addition, such error increases with the farther distance.



(a) Different user states

(b) Different distances to wall

Fig. 5. Unilateral acoustic ranging with different user states and distances to wall.

4.2 Floor Plan Construction

Construction Effect. We illustrate the construction effect in an 8.7 m × 6.3 m classroom in teaching building, a 7 m × 4.5 m living room and a 6 m × 11 m corridor in residential building. The three experimental scenarios contain several doors, windows, and blackboards/closets. In order to construct the floor plan, a user hold the smartphone horizontally and walk along the experimental scenarios border, and we produce the position of wall segments by acoustic ranging. Such scattered points are drawn on the map based on the walking trajectory (Fig. 6(b), Fig. 7(b) and Fig. 8(b)). Next, our refinement algorithm improve the constructed floor plan with corners and line segments (Fig. 6(c), Fig. 7(c) and Fig. 8(c)).

Quantitative Results. In order to evaluate the reconstructed floor plan precisely, we overlay it onto the ground truth to achieve the maximum overlap (Fig. 9), and observe that the location errors of wall segments are all within 0.3 m. Next, we define the precision, recall and F-score as:

$$P = \frac{S_{re} \bigcap S_{gt}}{S_{re}}, R = \frac{S_{re} \bigcap S_{gt}}{S_{gt}}, F = \frac{2P \cdot R}{P + R} \tag{9}$$

where S_{re} denotes the shape of reconstructed map, S_{gt} denotes the shape of the ground truth, and $S_{re} \bigcap S_{gt}$ represents their overlap area.

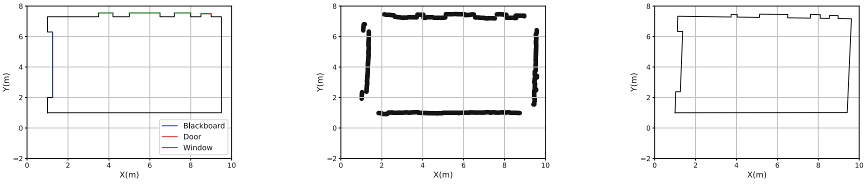
Table 1 shows the quantitative results for floor plan construction in classroom, living room and corridor. Compared with BatMapper and CrowdInside [8], the recall and F-score of our BatMapper-Plus are significantly higher than the other methods, which indicates that we produce more precise indoor maps. In addition, since the other methods generate floor plans with larger areas than the ground truth, their recall values are higher.

Table 1. Shape evaluation of floor plans.

Sense	Classroom			Living room			Corridor		
Criterion	R(%)	P(%)	F(%)	R(%)	P(%)	F(%)	R(%)	P(%)	F(%)
CrowInside	77.28	100	87.18	74.80	100	85.58	58.82	100	74.07
BatMapper	96.36	99.61	97.96	97.46	96.21	96.58	84.71	94.12	89.17
BatMapper-Plus	97.89	99.48	98.68	99.05	98.46	98.75	97.65	91.76	94.61

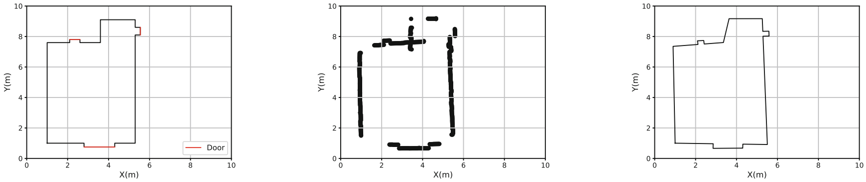
4.3 Connection Area Detection

Stairs. In order to evaluate the stair detection accuracy, users conduct a 2-minute walk either on the ground or climbing stairs, and we predict the location type for each walking step. In addition, we collect the inertial data with different postures during walking, i.e., holding the smartphone horizontally or with an arbitrary posture. As shown in Table 2, there are at most six incorrect steps on each walk for all postures, with an approximate accuracy of 97%. We look into such incorrect steps and find them are mainly located at the junction area on each floor, with slight impacts on stair detection.



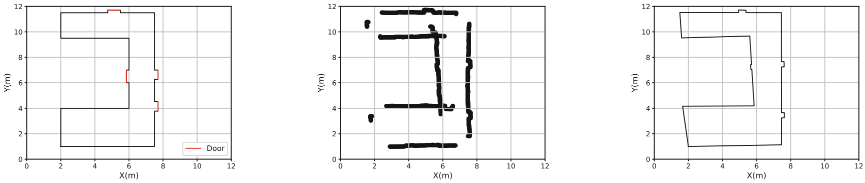
(a) Ground truth (b) Scattered points (c) Constructed map

Fig. 6. Construction process of a classroom in teaching building.



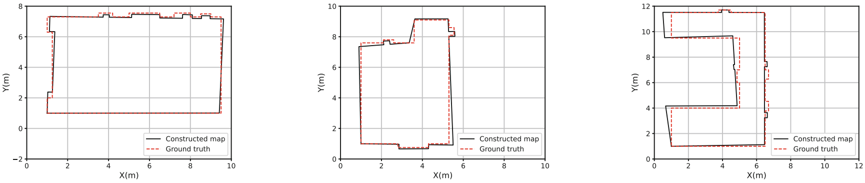
(a) Ground truth (b) Raw points (c) Constructed map

Fig. 7. Construction process of a living room in residential building.



(a) Ground truth (b) Raw points (c) Constructed map

Fig. 8. Construction process of a corridor in residential building.



(a) Classroom (b) Living room (c) Corridor

Fig. 9. Comparison between reconstructed floor plans and the ground truth.

Table 2. Accuracy of stair and elevator detection.

Area	Stair		Elevator		
State	Arbitrary posture	Horizontal posture	Rising	Stable	Sinking
Accuracy	96.84%	97.06%	100%	100%	100%

Elevator. As shown in Table 2, for a statically standing user, all test data are correctly detected (either on rising/sinking elevators or on the ground), thus the detection accuracy of elevators reaches 100%.

5 Related Work

Indoor Floor Plan Construction. At present, the construction of indoor floor plan is mainly realized by the combination of image and inertial sensor. Jigsaw [4] obtains the spatial relationship between adjacent landmark objects from the image taken by the user and the inertial sensor data, and combines the user's trajectory and the position of the captured image to generate a complete plan. Plansketcher [9] uses deep learning technology to extract new comprehensive features to identify different landmarks. Then the indoor floor plan is constructed based on sensing data, depth data and images. [10] combines the magnetic fingerprint map and user trajectory to build the indoor floor plan. IndoorCrowd2D [11] uses the image information and sensory data in crowdsourcing data to restore the building's structure. MapGENIE [12] uses syntax to represent the structural information of buildings, which has a better effect than simple trajectory mapping.

Acoustic Ranging. Acoustic ranging is a relatively new content in the field of mobile computing. DeepRange [13] uses a depth neural network to estimate the distance. [14] proposes an improved TOA estimation method to maintain high ranging accuracy and robustness in the closed environment of reverberation room. [15] and DopEnc [16] calculate the distance by measuring the time between the initial pulse of the smartphone and its reflection.

Inertial Tracking. Indoor tracking has a lot of related researches, mainly through image and inertial sensor. Walkie-Markie [17] uses WiFi tags and inertial sensor data to build user's trajectories. Zee [18] uses inertial sensors and WiFi signals for tracking. VeTrack [19] uses the inertial sensor of the mobile phone to track the position of the vehicle in real time. [20] tracks the user by gradually integrating WiFi interface and inertial sensors of smartphones. EasiTrack [21] uses the RF signal to accurately infer the moving distance of the target to achieve tracking. DeepIT [22] achieves higher precision tracking by evaluating the reliability of inertial data and synthesizing its data opportunitically. Imulet [23] adopts machine learning method to reduce the error of inertial data and improve the accuracy of tracking.

6 Conclusion

In this paper, we propose BatMapper-Plus to construct multi-level indoor floor plans without heavy human efforts and privacy/copyright concerns. Our unilateral ranging technique eliminates the limitation of existing bilateral ranging and achieves accurate distance measurements with less than 2 cm errors. We also refine the reconstructed map with line segments to replace the scattered points, and identify connection areas among different floors. We build a prototype and conduct experiments in two buildings, and the results have shown our effectiveness.

Acknowledgments. This work was supported in part by the Fundamental Research Funds for the Central Universities 2021JBM029, NSFC under Grant 62072029 and Grant 61872027, Beijing NSF Grant L192004, DiDi Research Collaboration Plan, and OPPO Research Fund.

References

1. Wang, X., Marcotte, R.J., Olson, E.: Glfp: Global localization from a floor plan. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1627–1632 (2019)
2. Anderson, R., Curro, J.: Indoor navigation using convolutional neural networks and floor plans. In: Proceedings of ION GNSS+, pp. 2133–2150 (2021)
3. Harithas, S.S., Pardia, B.: Gennav: a generic indoor navigation system for mobile robots. In: Proceedings of IEEE I-SMAC, pp. 182–187 (2020)
4. Gao, R., et al.: Multi-story indoor floor plan reconstruction via mobile crowdsensing. IEEE Trans. Mob. Comput. **15**(6), 1427–1442 (2016)
5. Zhou, B., Elbadry, M., Gao, R., Ye, F.: Batmapper: acoustic sensing based indoor floor plan construction using smartphones. In: Proceedings of ACM MobiSys, pp. 42–55 (2017)
6. Pradhan, S., Baig, G., Mao, W., Qiu, L., Chen, G., Yang, B.: Smartphone-based acoustic indoor space mapping. Proc. ACM on Interact. Mobile Wearable Ubiquit. Technol. **2**(2), 1–26 (2018)
7. Gao, R., et al.: Glow in the dark: smartphone inertial odometry for vehicle tracking in gps blocked environments. IEEE Internet Things J. **8**(16), 12955–12967 (2021)
8. Alzantot, M., Youssef, M.: Crowdinside: automatic construction of indoor floorplans. In: Proceedings of GIS, pp. 99–108 (2012)
9. Peng, Z., Gao, S., Xiao, B., Wei, G., Guo, S., Yang, Y.: Indoor floor plan construction through sensing data collected from smartphones. IEEE Internet Things J. **5**(6), 4351–4364 (2018)
10. Luo, H., Zhao, F., Jiang, M., Ma, H., Zhang, Y.: Constructing an indoor floor plan using crowdsourcing based on magnetic fingerprinting. Sensors **17**(11), 2678 (2017)
11. Chen, S., Li, M., Ren, K., Fu, X., Qiao, C.: Rise of the indoor crowd: reconstruction of building interior view via mobile crowdsourcing. In: Proceedings of ACM SenSys, pp. 59–71 (2015)
12. Philipp, D., et al.: Mapgenie: grammar-enhanced indoor map construction from crowd-sourced data. In: Proceedings of IEEE PerCom, pp. 139–147. IEEE (2014)
13. Mao, W., Sun, W., Wang, M., Qiu, L.: Deeprange: acoustic ranging via deep learning. Proceed. ACM Interact. Mobile Wearable Ubiquit. Technol. **4**(4), 1–23 (2020)
14. Liu, Z., Chen, R., Feng Ye, F., Guo, G., Li, Z., Qian, L.: Improved toa estimation method for acoustic ranging in a reverberant environment. IEEE Sens. J. **2**, 4844–4852 (2020)
15. Graham, D., Simmons, G., Nguyen, D.T., Zhou. G.: A software-based sonar ranging sensor for smart phones. IEEE Internet Things J. **2**(6), 479–489, 2015
16. Zhang, H., Du, W., Zhou, P., Li, M., Mohapatra. P.: Dopenc: acoustic-based encounter profiling using smartphones. In: Proceedings of ACM MobiCom, pp. 294–307 (2016)
17. Shen, G., Chen, Z., Zhang, P., Moscibroda, T., Zhang, Y.: {Walkie-Markie}: indoor pathway mapping made easy. In: Proceedings of USENIX NSDI, pp. 85–98 (2013)

18. Rai, A., Chintalapudi, K.K., Padmanabhan, V.N., Sen, R.: Zee: zero-effort crowdsourcing for indoor localization. In: Proceedings of ACM MobiCom, pp. 293–304 (2012)
19. Zhao, M., Ye, T., Gao, R., Ye, F., Wang, Y., Luo, G.: Vetrack: real time vehicle tracking in uninstrumented indoor environments. In: Proceedings of ACM SenSys, pp. 99–112 (2015)
20. Martínez del Horno, M., Orozco-Barbosa, L., García-Varea, I.: A smartphone-based multimodal indoor tracking system. Inf. Fusion **76**, 36–45 (2021)
21. Wu, C., Zhang, F., Wang, B., Liu, K.J.R.: Easitrack: decimeter-level indoor tracking with graph-based particle filtering. IEEE Internet Things J. **7**(3), 2397–2411, 2019
22. Gong, J., Zhang, X., Yuanjun Huang, J., Zhang, R.Y.: Robust inertial motion tracking through deep sensor fusion across smart earbuds and smartphone. Proc. ACM Interact. Mobile Wearable Ubiquit. Technol. **5**(2), 1–26 (2021)
23. Alloulah M., Tuominen, L.: Imulet: a cloudlet for inertial tracking. In: Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications, pp. 50–56 (2021)

Prediction of Cancellation Probability of Online Car-Hailing Orders Based on Multi-source Heterogeneous Data Fusion

Haokai Sun[1], Zhiqiang Lv[1,2], Jianbo Li[1](✉), Zhihao Xu[2], Zhaoyu Sheng[2], and Zhaobin Ma[2]

[1] College of Computer Science and Technology, Qingdao University, Qingdao 266071, China
{2021020687,lijianbo}@qdu.edu.cn

[2] Institute of Ubiquitous Networks and Urban Computing, Qingdao University, Qingdao 266701, China

Abstract. In recent years, the demand for urban travel is increasing and the travel modes are diverse. Online car Hailing has become an important way to meet the travel needs of residents. The online car-hailing platform receives tens of thousands of travel requests every day. However, a large portion of the thousands of orders are unfinished, that is, canceled by passengers. This not only reduces the income of drivers but also affects the order dispatching efficiency of the online car-hailing platform. To predict the cancellation probability of online car-hailing orders(OCP), the relationship between multi-source heterogeneous data and OCP is first introduced, in which the presence of idle taxis is the main factor for passengers to cancel their orders during the waiting period. Secondly, a deep learning model based on the Seq2Seq structure is designed to predict OCP in real-time. The model consists of an attribute fusion module, encoder layer, and decoder layer. Finally, a full experiment is carried out using the Didi Chengdu online car-hailing order data set to verify the effectiveness of the algorithm.

Keywords: Taxi order cancellation · Urban travel · Feature fusion · Deep learning

1 Introduction

With the development of the Internet and the popularity of smartphones, online car-hailing has become the most commonly used travel mode [1]. However, orders are not all fulfilled. When a user sends an online car-hailing order that is confirmed by the driver, the user may still cancel it while waiting for a ride [2]. Canceling the answered order not only increases the no-load cost of the driver, affects the driver's working mood but also increases the scheduling cost of the online car-hailing platform [3]. Moreover, it affects the riding experience of other users. Didi Chuxing proposes an online car-hailing dispatching algorithm based on the Markov decision process [4]. However, if the user cancels the order, optimization effect of the algorithm will be greatly reduced. The online car-hailing platform has formulated certain punishment measures to minimize the impact

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 168–180, 2022.
https://doi.org/10.1007/978-3-031-19214-2_14

of users' cancellation of orders on the order distribution system. However, in the face of fierce competition in the market, increasing the cost of user rides may lead to the loss of users [5]. Therefore, the research on OCP has important practical significance.

However, the existing literature rarely studies the cancellation behavior of users to the confirmed online car-hailing orders. The OCP varies with the distance of the online car-hailing from the user and the waiting time of the user. And OCP can be influenced by a variety of factors [6]. Through experiments, we find that OCP is strongly correlated with the waiting time of users and the distance of the online car-hailing from the user's pick-up point. And, the main reason for users to cancel their orders is the appearance of idle taxis and the arrival of public transportation. It means that most users cancel confirmed orders not by chance, but by a rational decision after weighing the cost of travel. In addition, OCP is also related to the total number of online car-hailing orders, time and other factors. Deep learning has made outstanding achievements in the field of intelligent transportation [7, 8]. This work designs a deep fusion network DF-OCP based on the Seq2Seq framework with the goal of real-time OCP prediction. The main contributions of this work are as follows:

- A variety of data affecting OCP are counted and modeled to assist in predicting OCP.
- A deep fusion network DF-OCP based on the Seq2Seq framework is proposed. The model consists of an attribute fusion module, an encoder layer, and a decoder layer. The attribute fusion module is used to fuse different OCP influencing factors. The encoder layer is used to encode the different input features and extract the implied features of online car-hailing data. The decoder layer decodes the output based on the above two to achieve predictive OCP.
- Based on the real data set in the real world, sufficient experiments are carried out to prove the effectiveness of the method. It is also compared with existing deep feature fusion models. The experimental results show that the DF-OCP model can achieve the smallest experimental error.

2 Literature Review

By analyzing the operation data of Didi Chuxing in Shanghai, Wang et al. [9] find that there is a certain regularity in the time of users' OCP: the OCP is lowest in the morning and evening peak period, while it is higher in the flat peak period. Moreover, the travel distance and pick-up distance of canceled orders are significantly higher than those of completed orders. When the average pick-up time is longer, the probability of online car-hailing orders being canceled is lower. Besides, Wang et al. believed that the main reason for most passengers to cancel their answered orders was the appearance of idle taxis. The study of He et al. [10], for the first time, considered the cancellation behavior of customers in the ride-hailing market. And they also solved the optimization design problem of car-hailing platform pricing and penalty/compensation strategies. Li et al. [11]. Study the punishment scheme after users cancel the confirmed orders in the car-hailing platform. They find that a fixed penalty fee is likely to generate more users, while a time-based penalty scheme can minimize social costs. Abid et al. [12] consider the user's behavior of canceling orders in the design of the taxi scheduling algorithm.

By reducing the number of user cancellations, improving the experience of taxi drivers and reducing complaints, the effectiveness of taxi scheduling is improved. Lv et al. [13] predicted taxi traffic using graph convolutional networks. Xu et al. [14] predict travel demand considering natural environment and socioeconomic factors.

3 Data Design

3.1 Problem Definition

Let the total number of online car-hailing orders be C. For the i-th online car-hailing order, we use $D_i = \{d_1, d_2, .., d_n\}$ to represent the distance traveled at each sampling time before reaching the passenger's boarding point after the online-hailing car receives the order. Similarly, $W_i = \{w_1, w_2, .., w_n\}$ represents the waiting time of passengers at each sampling time after the driver answers the order. The order cancellation probability at the j-th sampling moment of this order is represented by p_j, $P_i = \{p_1, p_2, \ldots p_N\}$. Other data that can affect the OCP are uniformly represented by A. The goal of OCP prediction is shown in Eq. (1), where f represents the function map, $D, W, P \in \mathbb{R}^{C \times N}$, α, β, γ are the parameters to be found.

$$P = f(\alpha D + \beta W + \gamma A) \tag{1}$$

3.2 Social Factors

To facilitate the statistics and calculation of information, we use regular hexagons to divide the main urban area of Chengdu into 8519 different areas. We separately count the idle taxi transfer rate, online car-hailing demand, taxi distribution, and public transportation resource information in each hexagonal area.

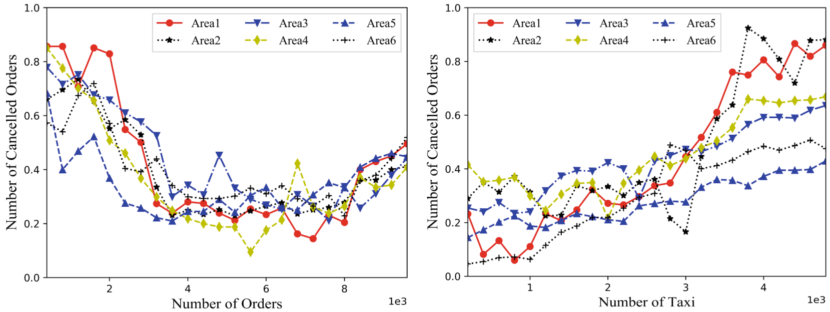


Fig. 1. OCP fluctuations caused by changes in travel demand supply and demand.

Supply and Demand of Travel Demand. For each car-hailing order, we add a car-hailing travel request for the area where the starting position of the order is located. For each ordinary taxi order, we add an idle taxi for the end location of the order. We use $O \in \mathbb{R}^{M \times N}$ to denote the total demand for online taxis and $E \in R^{M \times N}$ to denote the total number of idle taxis, where M represents the total number of areas.

We randomly selected six areas and counted the relationship between the total number of online car-hailing orders and the number of canceled orders (normalized to 0–1) in units of 30 min. The results are shown in Fig. 1 left. From the figure, we can see that with the continuous increase of the total amount of online car-hailing orders, the value of OCP gradually decreases as a whole. Besides, we counted the relationship between the number of idle cabs and the number of canceled orders (normalized to 0–1) in each area, and the results are shown on the right in Fig. 1. As can be seen from the figure, with the increase in the number of idle taxis, the OCP also increases. This shows that if the user encounters an idle taxi, there is a high probability of canceling the confirmed order and taking an idle taxi instead.

When a passenger encounters an idle taxi while waiting, the passenger will measure the current travel cost and further decide whether to cancel the order. We assume that it takes S time slices to reach the destination by taking a car-hailing or a taxi, and the car-hailing arrives at the user's pick-up point at time t. When the user encounters an idle taxi at t', the travel cost of choosing to continue waiting is shown in Eq. (2), where α is the time cost of taking the car-hailing, and θ is the time cost of waiting. The travel cost of users choosing to cancel this order and take an idle taxi is:

$$C_1 = \alpha S + \theta t \tag{2}$$

$$C_2 = \mu S + \theta t^{'} + F \tag{3}$$

where μ is the time cost of taking a idle taxi, and F is the penalty for canceling the current online car-hailing order. When the cost of canceling an order is lower than the cost of continuing to wait, users are more inclined to cancel their current order and take an idle taxi. Therefore, we use $I = C_2 - C_1$ to denote the effect of idle taxis on OCP. The larger the value, the more likely the user is to cancel the order.

Idle Taxi Transfer Rate. Idle taxi transfer rate refers to the probability of taxis transferring from one area to another. It reflects the spatial transfer of idle online car-hailing vehicles. Taking the area A at time t as an example, PI represents the sum of idle taxi transfer rate from other areas to A, and PO represents the sum of idle taxi transfer rate from A to other areas. Then the idle taxi transfer rate of area A is shown in Eq. (4).

$$P_{At}^{idle} = f(x) = \begin{cases} \frac{PI}{1-PO}, & PO < 1 \\ 0, & PO \geq 1 \end{cases} \tag{4}$$

The idle taxi transfer rate in area A reflects the possibility of idle car-hailing vehicles in the surrounding area entering area A. The larger the value, the greater the demand for car-hailing in the area A than in the surrounding area. At this time, OCP in area A is lower.

Time. Figure 2 shows the normalized hourly average of OCP (averaged over the same time slice on the same weekday in different weeks). From Fig. 2 we can see that the daily fluctuations in the OCP show a similar intraday pattern, which is a strong indication of the reproducibility and predictability of the OCP on an hourly average. The minimum occurs from 7:00 to 9:00 and from 18:00 to 20:00. Due to work and other reasons,

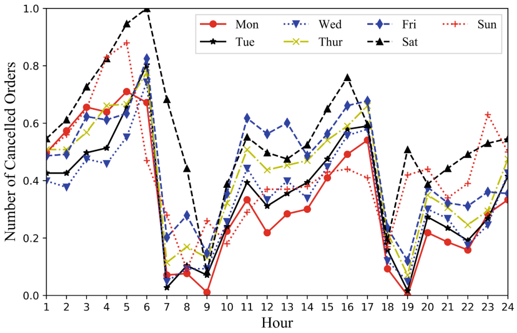
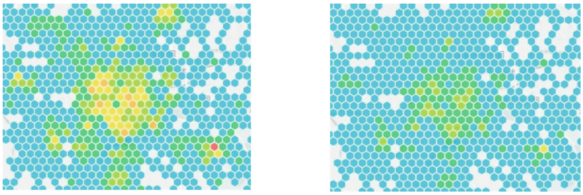


Fig. 2. The number of canceled orders in a day trended over time



(a) (b)

Fig. 3. Comparison of the relationship between public transportation resources and the total amount of online car-hailing order cancellations (normalized)

the overall demand for travel in the morning and evening peak hours is large and the timeliness requirements are high, so the demand for online car-hailing is high.

3.3 Public Transportation Resources

Public transportation resources are another important incentive for users to cancel online car-hailing orders [15]. Figure 3(a) is a heat map of Chengdu's public transportation resources, and Fig. 3(b) is a heat map of the number of canceled car-hailing orders. The public transport resource counts the number of bus stops and subway stations in different areas. It can be clearly seen that the hot spots for the number of canceled online car-hailing orders are concentrated in areas with more public transportation resources.

4 Model Design

In this session, we will explain to the proposed DF-OCP model as shown in Fig. 4. DF-OCP consists of three components: attribute fusion module, encoder layer and decoder layer. The attribute fusion module is used to deal with external factors. The encoder layer uses Gated Recurrent Unit(GRU) [16] to learn the impact on OCP from the data of ride-hailing distance and passenger waiting time. Finally, the decoder layer obtains the predicted value of the OCP.

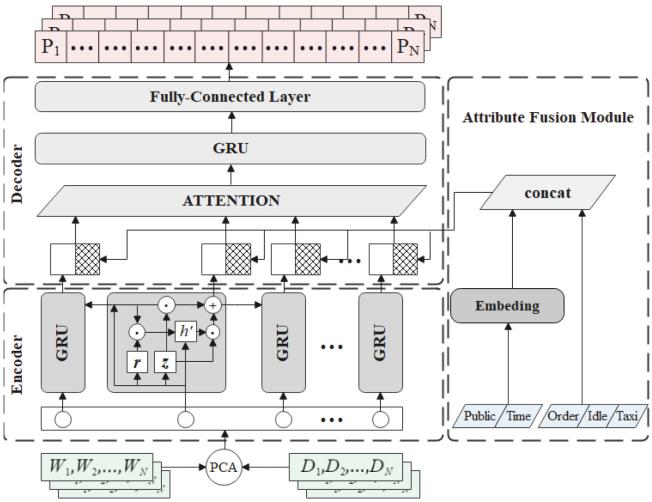


Fig. 4. Illustration of the proposed model for prediction of OCP

4.1 Attribute Fusion Module

We designed a simple and effective module to integrate the external factors into our model. Time data and distribution data of bus and subway are not in vector form, so we use the Embedding method [17] to encode attribute values as low-dimensional dense vectors. The specific method is to use a parameter matrix W to encode the feature attributes as vectors of specified dimensions. Equation 5 represents converting an E-dimensional feature into an F-dimensional vector where $W \in \mathbb{R}^{E\times F}$.

$$R^F = R^E * W \tag{5}$$

We encode both the time data and the distribution data of bus and subway as N-dimensional vectors. Embedding method can reduce the dimension of input features and reduce the computational pressure of deep learning models. The encoded temporal features and public transport features will be combined with other factors.

4.2 Encoder Layer

Algorithm1: Principal Components Analysis

```
**Input**: Sample set $D = \{x_1, x_2, \dots, x_m\}$, Dimensionality of output features $k$;
**Output**: Features of dimension $k$
$x_i^{(j)} \leftarrow x_i^{(j)} - \frac{1}{m}\sum_{i=1}^{m} x_i^{(j)}$ // Centralization
Calculate the covariance matrix of the sample $XX^T$
Eigenvalue decomposition of covariance matrix $XX^T$
Take the eigenvector corresponding to the largest $k$ eigenvalues $W = w_1, w_2, \dots, w_k$
$X' \leftarrow X \cdot W$
**return** $X'$
```

The encoder layer is mainly composed of a Principal Component Analysis (PCA) [18] module and a series-connected GRU module. After the driver confirms the order, there is a linear relationship between the distance traveled by the car-hailing and the waiting time of the user. To eliminate this collinearity problem, we combined two features into one feature using PCA. The calculation process is shown in Algorithm 1.Concating D and W to get the input I of PCA, $I \in \mathbb{R}^{2T \times N}$. The output of PCA is O_p, $O_p \in \mathbb{R}^{T \times N}$. O_p is input into the GRU module in series. Compared to Long Short-Term Memory(LSTM) [19], it combines the forget gate and input gate into a single update gate. Taking the input x_t at time t as an example, suppose the hidden state at time $t-1$ is h_{t-1}, then the reset gate r_t and the update gate z_t are:

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{6}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{7}$$

where W and U are trainable parameters, and σ(·) is the sigmoid activation function. Then we need to calculate the candidate hidden layer h', which can be regarded as the new information at the current moment. Finally, the update gate z_t controls how much information needs to be forgotten from the hidden layer h_{t-1} at the previous moment, how much information of the hidden layer at the current moment needs to be added, and the final output h_t is obtained.

$$h^{'} = tanh(Wx_t + r_t U h_{t-1}) \tag{8}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h' \tag{9}$$

The hidden state of each GRU unit will be fed into the next parallel GRU with the same structure to learn more abstract and deeper features in the ride-hailing trip data. There is no complex convolution structure and fewer training parameters, which makes GRU computation fast.

4.3 Decoder Layer

In the decoder, we use the attention mechanism [20] to reflect the importance of the current input features to the OCP to improve the learning effect of the model. Let the output of the i-th time slice encoder be h_i, and the output of the attribute fusion module be c_i. h_i has captured the driving data characteristics of the online car-hailing, while c_i has captured the impact of external factors on OCP at the same time. We can get the online car driving data with the attention weight added:

$$h^{at} = \sum_{i=1}^{N} \alpha_i \cdot h_i \tag{10}$$

where $\alpha_i = \frac{e^{\eta_i}}{\sum_j e^{\eta_j}}$, $\eta_i =< \sigma_{at}(c_i), h_i >$, $<>$ represents the inner product operator, and $\sigma_{at}(c_i)$ represents the nonlinear mapping to map c_i into a vector of the same length as h_i.

α_i is the weight of the i-th time slice data, and the sum of all data weights is 1. After the attention mechanism adds the weights, the data is fed to GRU for decoding. The output of the GRU in the decoder will be passed to several equal-sized fully connected layers. The output of the fully connected layer is the real-time prediction result of OCP.

5 Experiments

5.1 Dataset Introduction

The car-hailing order dataset used in this experiment is provided by the Didi Gaia Project. The dataset includes trajectory data, order cancellation probability data, hexagonal grid data, and idle car transfer rate of Didi Express orders in the urban area of Chengdu, with a total of 210,000 records. The sampling interval of the trajectory points is 2–4 s. The public transportation resources dataset counts the number of bus stops and subway stations in different hexagonal areas of Chengdu. The idle taxi dataset includes vehicle ID, latitude, longitude, occupancy, and time. We counted the number of unloaded taxis in different hexagonal areas at different times.

5.2 Quantitative Analysis

To verify the performance of our method in predicting OCP, we compare it with several state-of-the-art baselines.

- ARIMA [21]: ARIMA is an autoregressive model based on mathematical statistics that combines moving average and autoregression to model time series.
- RNN [22]: RNN can obtain “memory” information from previously input data to influence current input and output.
- TCN [23]: TCN is a CNN-based convolutional neural network for processing sequence data.
- VMD-LSTM [24]: The model first uses Variational Mode Decomposition (VMD) to decompose the time series data into Intrinsic Mode Functions (IMFs) at different time scales. Then combined with long short-term memory neural network (LSTM) to predict time-series data.
- MLRNN [25]: This model designs a taxi area clustering algorithm. Local and global-level prediction modules are developed to extract intra-cluster and inter-cluster features, respectively.The OCP is predicted by combining these two modules.
- DF-TAR [26]: The DF-TAR model consists of convolutional blocks, recursive blocks, fusion blocks, and fully connected blocks. Convolutional blocks are used to learn hidden representations of spatial features of the data.

The prediction errors of the different models mentioned above are shown in Table 1. The underlined data represents the best baseline, and the bolded data represents the minimum error. It is found that the ARIMA prediction results based on statistical methods have the highest error. Among the deep learning models, RNN has the largest prediction

Table 1. Comparison of prediction errors of different models

Model	MAE	MAPE (%)	RMSE
ARIMA	19.2	21	26
RNN	8.4	8.9	12.3
TCN	6.1	7.9	8.6
VMD-LSTM	4.3	5	6.8
MLRNN	3.8	4.8	5.9
DF-TAR	3.5	4.5	5.5
DF-OCP	**2.7**	**3.1**	**4.3**
Improvements	22.8%	31.1%	21.8%

error, followed by TCN. Compared with the above two models, the error of the VMD-LSTM model has been greatly reduced. Compared with other baselines, the DF-TAR model achieves the smallest error under all three indicators and is the optimal baseline.

The main reason for the poor performance of ARIMA is that OCP has no significant autocorrelation in time. RNN can use the historical information in the car-hailing data to assist in predicting the OCP at the current moment. However, when the online car-hailing data is too long, the RNN can only memorize local historical information and cannot perceive the global situation, so the prediction accuracy is limited. TCN expands the receptive field by dilated causal convolution, enabling it to extract more time-series feature information. Therefore, TCN has improved prediction accuracy compared to RNN. Through variational modal decomposition, the noise generated by the fusion of multi-source data is avoided, and the prediction accuracy is improved, but it is still not ideal. The OCP in different areas will be greatly affected by other factors, but the OCP in different areas will not affect each other. Moreover, when clustering areas, the setting of the number of clusters lacks theoretical guidance. The convolutional block of the DF-TAR model can learn the hidden features of the environment of the car-hailing trip data. In the fusion module, the above- learned feature data is rescaled, and only the features that have an important impact on OCP are retained. This feature extraction approach not only fully extracts the deep representation of the online car-hailing driving data, but also enables the dimensionality reduction operation of the learned features. Therefore, the model achieves the optimal baseline. However, since different online car-hailing orders have no obvious interdependence in space, its recursive module does not perform as expected. The DF-OCP model first performs feature dimension reduction on the input data through PCA. At the encoder layer, the input features are encoded with GRU and fused with external attributes. After obtaining the deep representation of the driving data of the online car-hailing vehicle, the GRU is used to decode the information and perform real-time prediction of OCP. This encoding-decoding data processing method is more suitable for feature extraction and representation of online car-hailing data. Therefore, DF-OCP achieves the smallest error.

5.3 Ablation Experiment

To verify the contribution of different factors to OCP prediction, we selected different factors each time to conduct multiple experiments and compared the experimental results. The effects of different features on OCP prediction results are shown through multiple experiments. The experimental results are shown in Table 2.

As shown in Table 2, among all external factors, the most influential factor on the prediction accuracy is the idle taxi data, followed by public transportation resources. The total number of online car-hailing orders and the idle car transfer rate have a similar impact on the forecast results. The time factor improves the accuracy of prediction results better than the above two factors. In the absence of external factors to assist the prediction, the prediction error of the DF-OCP model increases significantly.

Table 2. The effect of using different influencing factors on prediction accuracy

Model	MAE	MAPE (%)	RMSE
DF-OCP-without extra factors	4.8	6.4	7.2
+Online car-hailing order	3.6	5.4	5.8
+Time information	3.4	4.7	5.7
+Idle taxi transfer rate	3.6	4.6	5.8
+Public transport resources	3.2	4	5.3
+Idle taxi	3	3.8	5.2
+ALL	**2.7**	**3.1**	**4.3**
Improvements	10%	15.7%	5.7%

From the experimental results, the factors that have the greatest impact on OCP are the distribution of idle taxis and the distribution of public transportation resources. Due to similar prices and service content, there is such strong substitutability between online car-hailing and taxis in real life. Online car-hailing passengers often encounter idle taxis while waiting to be picked up. If a passenger calls and waits for an online car-hailing near public transportation facilities, the online car-hailing order may be canceled due to the arrival of public transportation during the waiting, and the public transportation is used instead. The ride-hailing order data and car-hailing idle transfer rate together reflect the demand for car-hailing in each area. The idle car transfer rate intuitively expresses the spatial flow of idle car-hailing vehicles. According to the origination location of each online car-hailing order, the DF-OCP model can use the idle car transfer rate to judge the impact of the online car-hailing travel demand on the cancellation probability of each order.

As shown in Fig. 5, we compare the prediction results of different models using different external factors. Here we only compare two state-of-the-art deep learning models.

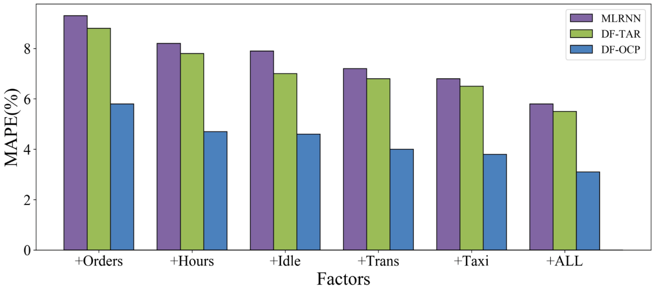


Fig. 5. Comparison of prediction results under different models and factors.

It can be seen from the figure that when each model uses external factors to assist in predicting OCP, the MAPE will decrease, and the most important improvement in accuracy is the idle taxi data. Additionally, the prediction accuracy of our proposed DF-OCP model consistently outperforms the other two models.

Combining the prediction results of the other two models, the factor that has the greatest impact on OCP is always the emergence of idle taxis. Therefore, we can conclude that the emergence of idle taxis is the most important factor causing users to cancel their current orders. In addition, the prediction accuracy of both the DF-TAR model and the MLRNN model improves when using other different factors. This also verifies the effectiveness of our proposed multi-source heterogeneous data fusion.

6 Conclusion

Aiming at the practical problem of users' cancellation probability of answered online car-hailing orders, this work first designs a variety of OCP-related data. Second, a deep fusion network DF-OCP based on the Seq2Seq structure is designed to predict OCP. The model consists of three parts: attribute fusion module, encoder layer, and decoder layer. Finally, we conduct sufficient experiments with real-world real datasets to demonstrate the effectiveness of the DF-OCP model. The experimental results show that the proposed method can effectively predict OCP. Also, the factors that have the most influence on the probability of cancellation of confirmed orders are the presence of idle taxis and the availability of public transport.

Acknowledgement. This research was supported in part by National Key Research and Development Plan Key Special Projects under Grant No. 2018YFB2100303, Shandong Province colleges and universities youth innovation technology plan innovation team project under Grant No. 2020KJN011, Shandong Provincial Natural Science Foundation under Grant No. ZR2020MF060, Program for Innovative Postdoctoral Talents in Shandong Province under Grant No. 40618030001, National Natural Science Foundation of China under Grant No. 61802216, and Postdoctoral Science Foundation of China under Grant No. 2018M642613.

References

1. Lyu, T., Wang, P.S., Gao, Y., Wang, Y.: Research on the big data of traditional taxi and online car-hailing: a systematic review. J. Traffic Transp. Eng. (English Edition) **8**(1), 1–34 (2021)
2. Sun, Z., Xu, Q., Zhang, G., Liu, J.: Pricing and matching for on-demand platform considering customer queuing and order cancellation. In: INFOR: Information Systems and Operational Research, pp. 1–39 (2022)
3. Xu, K., Saberi, M., Liu, W.: Dynamic pricing and penalty strategies in a coupled market with ridesourcing service and taxi considering time-dependent order cancellation behaviour. Transp. Res. Part C: Emerg. Technol. **138**, 103621 (2022)
4. Xu, Z., et al.: Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 905–913 (2018)
5. Lv, T.: Factors affecting users' stickiness in online car-hailing platforms: an empirical study. Int. J. Internet Manuf. Serv. **7**(1), 176–189 (2020)
6. Liu, M.: Factors influencing online car-hailing demand: A perspective of data analysis. In: 2020 Chinese Control and Decision Conference (CCDC), pp. 3437–3442. IEEE (2020)
7. Wang, Y., Lv, Z., Sheng, Z., Sun, H., Zhao, A.: A deep spatio-temporal meta-learning model for urban traffic revitalization index prediction in the COVID-19 pandemic. Adv. Eng. Inf., 101678 (2022)
8. Lv, Z., Li, J., Dong, C.: Deep learning in the COVID-19 epidemic: a deep model for urban traffic revitalization index. Data Knowl. Eng. **135**, 101912 (2021)
9. Wang, X., Liu, W., Yang, H., Wang, D., Ye, J.: Customer behavioural modelling of order cancellation in coupled ride-sourcing and taxi markets. Transp. Res. Part B: Methodol. **132**(1), 358–378 (2020)
10. He, F., Wang, X., Lin, X.: Pricing and penalty/compensation strategies of a taxi-hailing platform. Transp. Res. Part C: Emerg. Technol. **86**(1), 263–279 (2018)
11. Li, X., Li, Q.: Time-based or fixed-fee? how to penalize cancellation of orders of car-hailing applications. Int. J. Prod. Econ. **232**(1), 107960 (2021)
12. Abid, A., Nawaz, N.A., Farooq, M. S., Farooq, U., Abid, I.: Taxi dispatch optimization in smart cities using TOPSIS. Secur. Commun. Netw. **2022** (2022)
13. Lv, Z, Li, J, Dong, C.: DeepSTF: a deep spatial–temporal forecast model of taxi flow. Comput. J. (2021)
14. Xu Z, Lv Z, Li J, Sun H, Sheng, Z.: A novel perspective on travel demand prediction considering natural environmental and socioeconomic factors. In: IEEE Intelligent Transportation Systems Magazine, pp. 2–25 (2022)
15. Bi, H., Ye, Z., Hu, L., Zhu, H.: Why they don't choose bus service? understanding special online car-hailing behavior near bus stops. Transp. Policy **114**(1), 280–297 (2021)
16. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
17. Gal, Y., Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. Adv. Neural Inf. Process. Syst. **29** (2016)
18. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemom. Intell. Lab. Syst. **2**(1), 37–52 (1987)
19. Hochreiter, S.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
20. Vaswani, A., Shazeer, N., Parmar, N.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)
21. Liu, S., Liu, S., Tian, Y., Sun, Q.: Research on forecast of rail traffic flow based on ARIMA model. In: Journal of Physics: Conference Series. IOP Publishing (2021)

22. Sherstinsky, A.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D **404**, 132306 (2020)
23. Torres, J.F., Jiménez-Navarro, M.J., Martínez-Álvarez, F., Troncoso, A.: Electricity consumption time series forecasting using temporal convolutional networks. In: Alba, E., et al. (eds.) CAEPIA 2021. LNCS (LNAI), vol. 12882, pp. 216–225. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85713-4_21
24. Liang, D., Xu, J., Li, S., Sun, C.: Short-term passenger flow prediction of rail transit based on VMD-LSTM neural network combination model. In: 2020 Chinese Control and Decision Conference (CCDC), pp. 5131–5136. IEEE (2020)
25. Zhang, C., Zhu, F., Lv, Y., Ye, P., Wang, F.: MLRNN: taxi demand prediction based on multi-level deep learning and regional heterogeneity analysis. IEEE Trans. Intell. Transp. Syst., 1–11 (2021)
26. Trirat, P., Lee, J. G. Df-tar: a deep fusion network for citywide traffic accident risk prediction with dangerous driving behavior. In: Proceedings of the Web Conference 2021, New York, NY, United States, pp. 1146–1156 (2021)

FedGAN: A Federated Semi-supervised Learning from Non-IID Data

Chen Zhao[1], Zhipeng Gao[1(✉)], Qian Wang[2], Zijia Mo[1], and Xinlei Yu[1]

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China
gaozhipeng@bupt.edu.cn

[2] Beijing University of Technology, Beijing, China

Abstract. Federated Learning (FL) lately has shown much promise in improving sharing model and preserving data privacy. However, these existing methods are only of limited utility in the Internet of Things (IoT) scenarios, as they either heavily depend on high-quality labeled data or only perform well under idealized distribution conditions, which typically cannot be found in practical applications. In this work, we propose FedGAN, a Generative Adversarial Network (GAN) based federated learning method for semi-supervised image classification. In IoT scenarios, a big challenge is that decentralized data among multiple clients are normally non-independent and identically distributed (non-IID), leading to performance degradation. To address this issue, we further propose a dynamic aggregation mechanism that can adaptively adjust client weights in aggregation. Extensive experiments on three benchmarks demonstrate that FedGAN outperforms related federated semi-supervised learning methods, including a 55.36% accuracy on CIFAR-10 with 2k labels and 70.65% accuracy on SVHN with 1k labels - just 100 labels per class. Moreover, we carry out an extensive ablation and robust study to tease apart the experimental factors that are important to FedGAN's improvement.

Keywords: Federated learning · Internet of Things · Self-supervised learning · Unsupervised learning

1 Introduction

Federated Learning (FL) [1] are ubiquitously employed to protect data privacy on IoT clients (e.g., mobile devices, laptops and wearable devices) for collaborate optimize model, e.g., user habits prediction [2], wireless network optimization [3], personalized recommendation [4]. However, a fundamental weakness of deep neural network is that it typically requires a lot of labeled data to work well, as shown in Fig. 1 (a), while data on devices always come with few accompanying labels, which limited FL applications. Existing methods usually leverages transfer learning to classify unlabeled data, which lacks robustness in case of non-IID data distribution. A universal FL method should work in both supervised and semi-supervised settings, which inspired the recent work to integrate semi-supervised

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 181–192, 2022.
https://doi.org/10.1007/978-3-031-19214-2_15

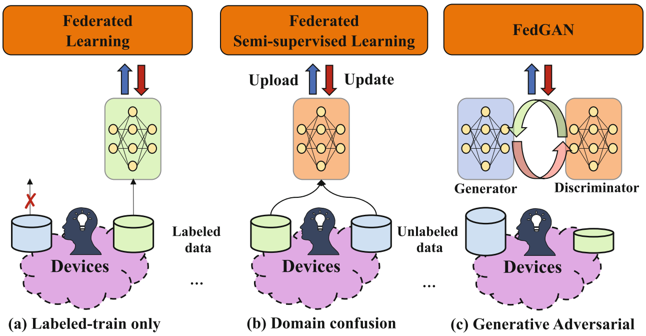


Fig. 1. (a) Federated Learning, which can only train labeled data. (b) Federated Semi-supervised Learning, which is insufficient robust in data non-IID scenarios. (c) FedGAN, which is an efficient method that optimizes sharing model when clients come with few labeled data and is robust to data non-IID.

learning techniques into the FL framework [5,6] (i.e. employing domain confusion between clients to train unlabeled data), as shown in Fig. 1 (b). However, for different devices, data are commonly non-IID since users always have different habits and usage frequencies. These independent are almost unable to be learned and optimized by knowledge transfer or domain confusion techniques.

The GAN [7] based semi-supervised learning have achieved great success in various applications, which learns the data classifier via an adversarial discriminator. Specifically, the generator G attempts to learn the data from real data pairs distribution to make the fake data distinguishable to the adversarial discriminator D. In this work, we propose FedGAN, a collaborative federated learning method for semi-supervised image classification. The framework of our FedGAN is shown in Fig. 1 (c). In FedGAN, inspired by Triple GAN [8] and Bad GAN [9] methods, we propose a GAN-based Training Mechanism (GTM) and a Dynamic Aggregation Mechanism (DAM). Specifically, GTM consists of three generators and a discriminator network to learn the correlation between labeled and unlabeled data, respectively. Then, all clients' model parameters are updated to the cloud server, and each client's network parameters weights are calculated by DAM adaptively. We validate our method empirically through a range of experiments on commonly used semi-supervised tasks. We show that FedGAN can reach similar accuracy levels to full-label tasks on real-world datasets such as Mini-ImageNet and COVID-19.

To summarize, we make the following contributions:

- We introduce a new method, FedGAN, to address an important but overlooked problem: leveraging unlabeled data from multiple parties for semi-supervised learning while preserving data privacy.

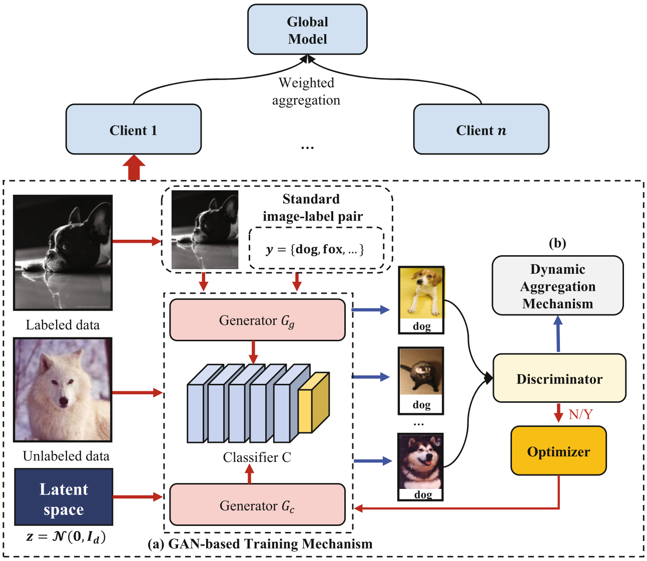


Fig. 2. The overall pipeline of FedGAN. Given a set of IoT clients C with labeled and unlabeled data $\{\mathbf{x_L}, \mathbf{x_U}\}$, our goal is to learn a sharing model N_G. FedGAN consists of two components: (a) GAN-based training mechanism; (b) dynamic aggregation mechanism.

- We propose a GAN-based training mechanism (GTM) for disjoint learning on labeled and unlabeled data by analyzing the impact of non-IID data in IoT scenarios.
- We propose a new module, dynamic aggregation mechanism (DAM), to dynamically adjust local model aggregation weight based on optimization difficulty caused by data non-IID.
- To the best of our knowledge, we are the first to use GAN-based federated learning to address the Semi-Supervised Learning (SSL) problem in IoT scenarios.

2 Overview

To outline our method, we consider the same SSL problem as in IoT scenarios. As shown in Fig. 2, given a set of IoT devices $\{c_1, c_2, ..., c_n\}$ and a cloud server, each client possesses a local dataset $\{\mathbf{x_L}, \mathbf{x_U}\}$ including a relatively small labeled dataset $(x_l, y_l) \sim p_l(x, y)$ and a large unlabeled dataset $(x_u) \sim p_u(x)$, let $\{1, 2, ..., K\}$ be the label space for classification task. Suppose the real data distribution as $p(x, y)$, we aim to training a sharing classifier C that can approximate the conditional distribution $p_C(y|x) \sim p(y|x)$. To achieve this, we will utilize the adversarial process that enables the classifier to learn from both labeled and unlabeled data.

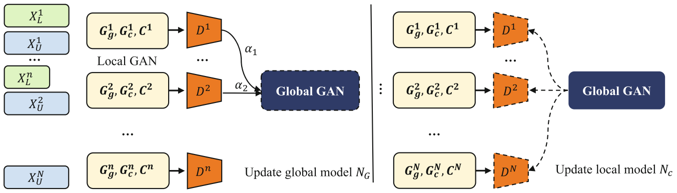


Fig. 3. The training process of FedGAN. The solid arrows represent the gradient flow from the local training procedure, and the dotted arrows represent the model update from the server. The dotted block means model parameters are fixed during this step and the solid block indicates that the model is being updated.

2.1 Objective of FedGAN

The FedGAN is based on the UGAN [10], for semi-supervised setting, each local model N_c has the same structure and are trained in an asynchronized fashion, the optimize objective of each local model is

$$\begin{aligned}\min_{C,G}\max_{D} V(C,D,G) = &-\mathbb{E}_{x,y\sim p(x,y)}[\log D(x,y)]\\ &+\mathbb{E}_{y\sim p(y),y\sim p_G(z)}[\log(1-D(G(y,z),y))]\\ &+\mathbb{E}_{x\sim p(x),y\sim p_C(y|x)}[\log(1-D(x,y))],\end{aligned} \tag{1}$$

where z is latent space (e.g., uniform or standard normal). G consists of two generator networks, G_g and G_c, that generate pseudo examples given real labels, and generate complement examples, respectively. The conditional classifier C and discriminator D are used to generate pseudo labels given data and distinguish the generated data-label pairs from the real data-label pairs, respectively.

In the FedGAN, local GANs are distributed over n clients, these local GAN individually optimized by a subnet of local datasets. Thus the loss function of our FedGAN can be represented as

$$\begin{aligned}\min_{C,G}\max_{D} V(C_{1:n},D_{1:n},G_{1:n}) = &\sum_{i\subseteq n}\alpha_i\{-\mathbb{E}_{x,y\sim p(x,y)}[\log D_i(y|x)]\\ &+\mathbb{E}_{y\sim p(y),y\sim p_G(y)}[\log(1-D_i(G_i(y,z),y))]\\ &+\mathbb{E}_{x\sim p(x),y\sim p_C(y|x)}[\log(1-D_i(x,y))]\},\end{aligned} \tag{2}$$

where α_i are used to balance the weight of each client. The loss function of each client is obtained by *GTM*, and α_i is computed by the proposed *DAM*. We will detail the optimization process, the GTM and the DAM in Sects. 2.2, 2.3, and 2.4 respectively.

2.2 Optimization Process

The optimization process of the FedGAN is shown in Fig. 3. In each communication round, the server randomly selects partial clients $\mathcal{B}$ to participate in training, then the global model N_G is updated iteratively in the following order:

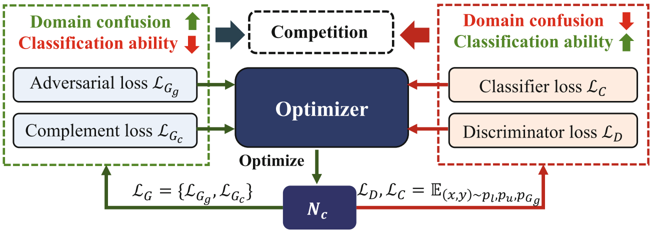

Domain confusion ⬆
Classification ability ⬇
Competition
Domain confusion ⬇
Classification ability ⬆
Adversarial loss $\mathcal{L}_{G_g}$
Complement loss $\mathcal{L}_{G_c}$
Optimizer
Classifier loss $\mathcal{L}_C$
Discriminator loss $\mathcal{L}_D$
Optimize
$\mathcal{L}_G = \{\mathcal{L}_{G_g}, \mathcal{L}_{G_c}\}$
N_c
$\mathcal{L}_D, \mathcal{L}_C = \mathbb{E}_{(x,y)\sim p_l, p_u, p_{G_g}}$

Fig. 4. Pipeline of the GTM. GTM consists of four components: (1) generator G_g, generate pseudo data given real labels; (2) conditional generator G_c, generate complement data to obtain class boundaries in low-density areas; (3) conditional classifier C, generate pseudo labels given data; (4) discriminator D distinguishes the generated data-label pairs from the real data-label pairs.

1) Calculating the participate clients loss $\mathcal{L}_c$ and weight α_c, where $i \subset \mathcal{B}$, then update global model N_G using local losses $\mathcal{L}_{Total} = \sum_{c \in \mathcal{B}} \alpha_c \mathcal{L}_c$.

2) After the global model is updated, all client models will be updated by $N_c \leftarrow N_G$, completing a communication round.

We apply the cross-entropy loss in local model training and further analyze the FedGAN framework in semi-supervised settings. Note that the framework can be applied to various neural network models including VGG-Net and ResNet.

2.3 GAN-Based Training Mechanism

A powerful GAN can memorize the empirical distribution from a real labeled dataset on each client. As shown in Fig. 4, the main idea of the competition mechanism is that through controlling generator G to make domain confusion and compromises the classification ability, meanwhile, $\mathcal{L}_C$ and $\mathcal{L}_D$ try to improve the classification ability and compromise the domain confusion.

Given this setup, there are two questions we need to address. The first is how the classifier benefits from joint training with generators. For this, FedGAN contains two generator $G = \{G_g, G_c\}$, as the weighted average between two objective functions: the good generator G_g are used to generate distribution p_g exactly matches the true data distribution p. The loss function of generator G_g follows the base GAN, represented as

$$\mathcal{L}_{G_g} = -\mathbb{E}_{x,y \sim p_{G_g}(x,y)}[1 - p_D(x,y)], \tag{3}$$

Then, the complement generator G_c are used to encourage the discriminator to obtain class boundaries in low-density areas to improve sharing model generalization performance, thus the objective function of $\mathcal{L}_{G_c}$ can be represented as

$$\mathcal{L}_{G_c} = -\mathcal{H}[p_{G_c}] + \left\|\mathbb{E}_{x,y \sim p_u(x,y)} f(x) - \mathbb{E}_{x,y \sim p_{G_c}(x,y)} f(x)\right\|_2^2, \tag{4}$$

where $-\mathcal{H}[p_{G_c}(x)]$ is the negative entropy of samples generated by G_c, which is used to avoid collapsing issues during training.

The second question is how to employ the classifier in the context of the semi-supervised setting. For this, we classified input samples by a classifier C which approximately the conditional discriminative distribution $p_C(y|x) \approx p(y|x)$. The classifier C takes four types of samples (labeled samples, samples from G_g and G_c, and unlabeled samples) and produces pseudos labels for them according to conditional distribution $p_C(y|x)$. For the labeled samples x_l and generated by G_g, we hope C put them to right class, while for the samples from G_c and unlabeled samples, we hope C put them to the $K+1$ class as negative samples. We refer to the loss function of C as

$$\begin{aligned}\mathcal{L}_C = \mathcal{L}_{C_1} + \mathcal{L}_{C_2} + \mathcal{L}_{C_3} + \mathcal{L}_{C_4} = &- \mathbb{E}_{x,y\sim p_l(x,y)}[\log(p_C(y|x, y \leq K))] \\ &- \mathbb{E}_{x,y\sim p_{G_g}(x,y)}[\log(p_C(y|x, y \leq K))] \\ &- \mathbb{E}_{x,y\sim p_u(x)}[\log(1 - p_C(y = K+1|x))] \\ &- \mathbb{E}_{x,y\sim p_{G_c}(x,y)}[\log(p_C(y = K+1|x))], \quad (5)\end{aligned}$$

where $\mathcal{L}_{C_1}$ and $\mathcal{L}_{C_2}$ represent cross-entropy loss for labeled data and G_g generated samples, respectively. The $\mathcal{L}_{C_3}$ forces the C generate labels that conform to the real data for the unlabeled data $p(x_u)p_C(y|x_u)$, and the $\mathcal{L}_{C_4}$ are used to make the data generated by the completion generator be recognized as negative data.

The discriminators are not the key points of our work, thus we follow the definition in [7,10] expressed as

$$\begin{aligned}\mathcal{L}_D = &- \mathbb{E}_{x,y\sim p_l(x,y)}[\log(p_D(x,y))] \\ &- \frac{1}{2}\mathbb{E}_{x,y\sim p_{G_g}(x,y)}[1 - \log(p_D(x,y))] \\ &- \frac{1}{2}\mathbb{E}_{x,y\sim p_C(x,y\leq K)}[1 - \log(p_C(x, y \leq K))], \quad (6)\end{aligned}$$

where discriminator D only treats the labeled data paris as positive samples, while the pseudo paris from both G and C as negative samples.

2.4 Dynamic Aggregation Mechanism

In this subsection, we define the clients' weight in parameters aggregation and detail how we dynamically adjust it, based on labeled data size and optimization difficulty.

While the local loss is continuous, there are indirect effects when data non-IID on each client, this can have a drastic effect on the aggregation process. As an extreme example, the global model could degeneracies, where all local model parameters get the same weights, making convergence impossible. To address this problem, we utilize the Dynamic Task Prioritization (DTP) [11] to adjust the model aggregation weights, for each client c, we select a performance indicator denoted by $\kappa_t \in [0,1]$, the κ_t is a average precision list. In FedGAN, we compute κ_t as an exponential changing average

$$\overline{\kappa}^{(t)} = \theta\kappa^t + (1-\theta)\overline{\kappa}^{(t-1)}, \quad (7)$$

where t is the communication rounds, $\theta \in [0, 1]$ is a discount factor to measure priorities of historical recent performance. We adopt the Focal Loss [12] instead of cross-entropy loss as our way to down-weight easier task, task weight defined as

$$\mathrm{FL}(\overline{\kappa}_t; \gamma) = -(1 - \overline{\kappa}_t)^{\gamma} \log(\overline{\kappa}_t), \tag{8}$$

where γ is the focusing factor, setting $\gamma > 0$ will reduces the weight for a well-performance clients. Then we scale each client-specific loss $\mathcal{L}_c$ by computing the optimize difficulty $\mathrm{FL}(\overline{\kappa}_t; \gamma_t)$, our total aggregation loss can be represent as

$$\mathcal{L}_{Total} = \sum_{c=1}^{|n|} \mu \cdot \mathrm{FL}(\overline{\kappa}_t; \gamma_t) \mathcal{L}_c, \tag{9}$$

where $\mathcal{L}_c$ is the losses of local GANs, $\mu = \exp(\left|\frac{\mathbf{x_L^c}}{\mathbf{x_L}}\right|)$ is the proportion of labeled data in total data for client c, the weight α_c in Eq. 2 is calculated by $\mu \cdot \mathrm{FL}(\overline{\kappa}_t; \gamma_t)$.

3 Experiment

In the following experiments, we mainly evaluate the performance of FedGAN on three aspects, average accuracy, robustness, and scalability.

Datasets. We evaluate the efficacy of FedGAN on several commonly used SSL benchmarks. Specifically, we perform experiments with varying labeled data size on MNIST, CIFAR-10, and SVHN, following standard semi-supervised learning evaluation protocols [13]. The previous works were almost performed with a balanced split of data in which every client was assigned the same size of data points. In realistic IoT scenarios, however, the data sets on different clients will typically vary heavily in and labeled size. To simulate different degrees of non-IID, we split the data according to [14], the data size of each client defined as

$$\varphi_c(\lambda, \sigma) = \frac{\lambda}{n} + (1 - \lambda) \frac{\sigma^c}{\sum_{c=1}^{n} \sigma^c}, \tag{10}$$

where λ controls the minimum data size on each client, and σ controls the data concentration.

Experimental Setting. Our FedGAN is implemented in PyTorch, and all experiments are performed on a server with four NVIDIA Geforce RTX 3090 GPUs. For all experiments, by default, we set $n = 20$ in Eq. (2), $\theta = 0.75$ in Eq. (7), $\gamma = 0.6$ in Eq. (8), and we fixed $\lambda = 0.1$ and $\sigma = 0.9$ in Eq. (10). Each local training epoch set 50 iterations, and batch number set 32. The local GAN architecture adopt is based on TripleGAN [8] and BadGAN [9], and the optimization is Adam, learning rate is 0.0002 and momentum parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$.

Table 1. Ablation study on our improvements

Inprovement	Average accuracy(%)		
Component	MNIST	CIFAR-10	SVHN
W/O G_c	75.86	53.85	69.74
W/O C	73.66	50.61	66.85
W/O DAM	72.38	54.22	66.29
FedGAN	**77.43**	**55.36**	**70.65**

3.1 Ablation Study

Since FedGAN is a combination and improvement of existing techniques and achieves better performance, we report the results of the ablation study in Table 1. In the following, we analyze the effects of several improvements in our method.

Influence of GAN-based Training Mechanism. We train on the CIFAR-10 dataset ten times and reach between 78.56% and 75.27% test accuracy rate with a median of 77.43%. Then we removed the G_c in FedGAN and compared it with FedGAN with default hyperparameters, experimental results in Table 1 show that the model performance has slightly dropped around 1.57%, which means the complement generator can effectively preserve classification knowledge from previous training.

To further analyze the influence of GTM, we removed the classifier C and extend the discriminator D as classifier [7], we observe that the average model performance on three datasets has dropped (3.77%–4.70%). This gap tells us that our GTM improves model optimization while keeping reliable classification knowledge.

Influence of Dynamic Aggregation Mechanism. We removed the DAM in FedGAN evaluation and the results are shown in Table 1, our method without DTM shows a decline compared with FedGAN (1.14%–5.05%), which indicates that the DTM component is essential to adjust the loss weight between clients, and this implies that our improvement effectively utilizes inter-client knowledge in this imbalanced setting.

Influences of Discount and Focusing Factor. We study the interactions between the discount factor θ and focusing factor γ. We fix γ, the experimental results in Fig. 5 left show that a too higher priority will ignore historical training performance and make the partial local model overfit. Therefore, we set $\theta = 0.5$ as our default value. As shown in Fig. 5 right, we vary γ between 1.0 and 2.0 in our experiments. We observe that a larger γ forces the network to focus on the more difficult classification task, although at the cost of easier task performance. Here, we set $\gamma = 1.6$ as our default value.

3.2 Comparison with Related Methods

We compare FedGAN with related Federated Semi-Supervised Learning (FSSL) methods which have the potential to optimize sharing model in semi-supervised settings. For a fair comparison, we utilize the same labeled dataset setting to optimize the model, their results are generated by running author-released codes with default settings.

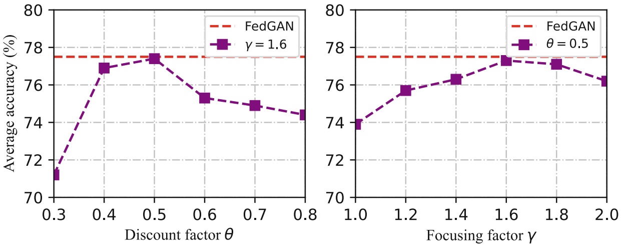


Fig. 5. Analysis of (a) impact of discount factor and (b) impact of focusing factor on the performance of FedGAN. The average accuracy of FedGAN with default hyperparameters is in the red dotted line. (Color figure online)

Table 2. Compared with SSL-based and GAN-based methods

Method	MNIST	CIFAR-10	SVHN
SSL-based methods			
FL UDA	68.64	52.69	64.31
FL Pseudo Label	70.75	44.63	52.79
DS-FL	75.32	53.63	66.83
GAN-based methods			
FL FM-GAN	73.85	53.28	68.25
FL Triple-GAN	73.83	51.26	68.56
FL BadGAN	75.47	53.64	65.84
FedGAN(Ours)	**77.43**	**55.36**	**70.65**

The first class is SSL-based methods (including UDA [15], Pseudo Label [16], DS-FL [5]), Table 2 shows averaged local model accuracy on non-IID tasks. We observe that our FedGAN outperforms other FSSL methods with better performance (1.73%-17.86%). The second class we compared is GAN-based methods (including FM-GAN [7], Triple-GAN [8], BadGAN [9]), the average accuracy is shown in Table 2, our proposed method outperforms GAN-based methods with higher accuracy (1.72%-4.81%). Particularly, for the prior work of semi-supervised GAN [8–10], we observe that the classification performance of their methods heavily relies on the adversarial learning between generator and classifier which lead to the decline of model performance in a distributed environment. Contrarily, our method is more robust with the non-IID settings.

3.3 Robustness

Below we conduct a series of experiments to evaluate the model robustness of our FedGAN in a real-world application.

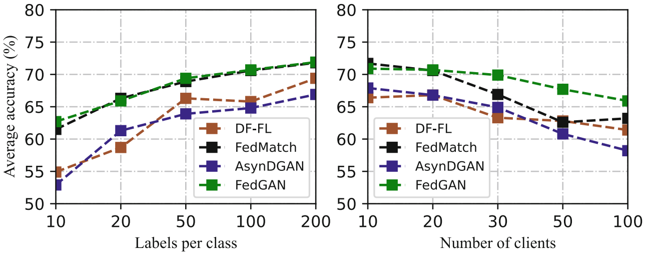


Fig. 6. Robustness evaluation on the different number of client and labels. (a) results of differently labeled data per class. (b) results under different number of clients.

Influence of the Number of Labels per Class. To evaluate the method robustness, we conduct a comparative experiment on related works of FSSL and our method (including DS-FL [5], FedMatch [6], AsynDGAN [17]). Experimental results in Fig. 6 left show that FedGAN achieves consistent performance improvement (0.91%–11.80%) as the number of labeled data increases. Interestingly, we observe those baseline methods significantly affected by labeled data numbers compared with FedGAN, which means our method effectively addresses the issues of lack labeled caused by requirements of various application scenarios.

Influence of the Number of Clients. The comparison results in Fig. 6 right show that with the increase of clients, our method achieves better performance compared with the baselines (0.12%–8.73%). Moreover, although the model performance of our method is slightly lower than baseline methods when there are fewer clients, due to the huge amount of IoT devices in the actual application scenario, our method still has strong advantages.

3.4 Scalability

Then, to evaluate whether FedGAN can scale to problems with a large scale and higher difficulty, we now turn to the Mini-ImageNet dataset [18] and COVID-19 Radiography dataset [19] with supervised and semi-supervised settings.

Specifically, we consider two experiment settings with different natures:

- We use all images and training ResNet-50 and VGG-19 with a centralized method.
- We use 1 K labeled data training with a semi-supervised method.

The results are summarized in Table 3. In the first setting, even with only 1K labeled data, FedGAN can offer decent or even competitive performances compared to the centralized methods trained with full supervised data. In the second setting, FedGAN consistently bing significant gains compared to the unsupervised baseline. This shows that FedGAN is not only able to scale but also able to utilize out-of-domain unlabeled data to improve model performance.

Table 3. Top-1/Top-5 accuracy on two real-world datasets

Method	SSL	Mini-ImageNet	COVID-19
ResNet-50	✗	72.17/84.54	90.26/95.27
VGG-19	✗	70.36/85.47	91.56/95.61
AsynDGAN	✓	47.32/72.51	81.38/88.56
FedMatch	✓	66.48/77.68	83.03/89.68
FedGAN(Ours)	✓	71.55/81.14	88.25/93.47

4 Conclusion

In this paper, we propose FedGAN, a federated learning method for semi-supervised image classification where each IoT clients learn with partially labeled data. To guarantee such sharing methods are efficient and robust, we proposed a GAN-based training mechanism and a dynamic aggregation mechanism.

Our experimental results suggest that FedGAN can effectively optimize sharing model in semi-supervised and non-IID settings while preserving accuracy. At the same time, we note that FedGAN also provides good scalability, our analysis in Sect. 3.4 suggests that this may be since it preserves the original knowledge for each client, and optimized the weight between clients.

Acknowledgemnt. This work is supported by the National Natural Science Foundation of China (62072049).

References

1. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. Proc. Mach. Learn. Res. **54**, 1273–1282 (2017)
2. Hard, A., et al.: Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604 (2018)
3. Tran, N.H., Bao, W., Zomaya, A., Nguyen, M.N.H., Hong, C.S.: Federated learning over wireless networks: Optimization model design and analysis. In: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, pp. 1387–1395 (2019)
4. Zhu, Y., Liu, Y., Yu, J.J.Q., Yuan, X.: Semi-supervised federated learning for travel mode identification from GPS trajectories. IEEE Trans. Intell. Transp. Syst. **23**3, 1–12 (2021)

5. Itahara, S., Nishio, T., Koda, Y., Morikura, M., Yamamoto, K.: Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. arXiv preprint arXiv:2008.06180 (2020)
6. Jeong, W., Yoon, J., Yang, E., Hwang, S.J.: Federated semi-supervised learning with inter-client consistency and disjoint learning. arXiv preprint arXiv:2006.12097 (2020)
7. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. Adv. Neural. Inf. Process. Syst. **29**, 2234–2242 (2016)
8. Li, C., Xu, K., Zhu, J., Zhang, B.: Triple generative adversarial nets. CoRR abs/1703.02291 (2017)
9. Dai, Z., Yang, Z., Yang, F., Cohen, W.W., Salakhutdinov, R.: Good semi-supervised learning that requires a bad GAN. CoRR abs/1705.09783 (2017)
10. Li, W., et al.: Semi-supervised learning using adversarial training with good and bad samples. Mach. Vis. Appl. **31**(6), 1–11 (2020). https://doi.org/10.1007/s00138-020-01096-z
11. Guo, M., Haque, A., Huang, D.-A., Yeung, S., Fei-Fei, L.: Dynamic task prioritization for multitask learning. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11220, pp. 282–299. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01270-0_17
12. Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007 (2017)
13. Oliver, A., Odena, A., Raffel, C., Cubuk, E.D., Goodfellow, I.J.: Realistic evaluation of deep semi-supervised learning algorithms. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 3239–3250. NIPS (2018)
14. Sattler, F., Wiedemann, S., Müller, K.R., Samek, W.: Robust and communication-efficient federated learning from non-iid data. IEEE Trans. Neural Netw. Learn. Syst. **31**(9), 3400–3413 (2019)
15. Xie, Q., Dai, Z., Hovy, E., Luong, M.T., Le, Q.V.: Unsupervised data augmentation for consistency training. arXiv preprint arXiv:1904.12848 (2019)
16. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on Challenges in Representation Learning, ICML, vol. 3, pp. 896 (2013)
17. Chang, Q., et al.: Synthetic learning: Learn from distributed asynchronized discriminator gan without sharing medical image data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13856–13866 (2020)
18. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei, L.F.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
19. Rahman, T., et al.: Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images. Comput. Biol. Med. **132**, 104319 (2021)

DEANet: A Real-Time Image Semantic Segmentation Method Based on Dual Efficient Attention Mechanism

Xu Liu[1], Rui Liu[1](✉), Jing Dong[1], Pengfei Yi[1], and Dongsheng Zhou[1,2](✉)

[1] National and Local Joint Engineering Laboratory of Computer Aided Design, School of Software Engineering, Dalian University, Dalian, China
{liurui,zhouds}@dlu.edu.cn

[2] School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China

Abstract. Image semantic segmentation is the basis of performing various tasks in computer vision. It has been widely used in medical imaging, robotics and many other fields. However, the existing image semantic segmentation technology cannot improve the segmentation speed while ensuring the segmentation accuracy, and cannot meet the requirements of real-time applications. Therefore, this paper proposes a real-time image semantic segmentation method based on dual efficient attention mechanism (DEANet). Pyramid sampling is introduced into the channel dimension to extract multi-scale information, and higher resolution aggregation features are adopted as the input of the spatial dimension. It can achieve high efficiency and accuracy of image semantic segmentation. The proposed DEANet was tested on two classic datasets. On the Cityscapes dataset, when the input size is 512 × 1024, the segmentation accuracy reaches 74.90% mIoU, and the segmentation speed reaches 99.91FPS. On the CamVid dataset, when the input size is 360 × 480, the segmentation accuracy reaches 70.07% mIoU and the segmentation speed reaches 142.72 FPS.

Keywords: Real-time semantic segmentation · Channel · Attention spatial attention

1 Introduction

As an indispensable part of computer vision, image semantic segmentation has been applied in many fields. It has been used in satellite remote sensing, medical imaging, and robotics. However, with the promotion and in-depth application, it also brings new challenges to image semantic segmentation. For example, in some fields the requirement of segmentation speed is very high. For the reason that improve the segmentation speed, it is often at the expense of the segmentation accuracy. Thus, how to strike a balance between segmentation speed and accuracy has become an urgent problem to be solved in image semantic segmentation. With the continuous development and progress of deep learning, deep neural networks with high accuracy have emerged one after another.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 193–205, 2022.
https://doi.org/10.1007/978-3-031-19214-2_16

And some of them have achieved good segmentation accuracy on classic datasets, for instance FCN [1], ResNet [2], PSPNet [3], etc. In recent years, many real-time semantic segmentation methods have also been proposed, such as ENet [4], ERFNet [5] and so on. Some of these methods improve the segmentation speed, but the segmentation effect is unsatisfactory. Or some methods maintain high segmentation effect, but the improvement of segmentation speed is very limited.

With the emergence of the attention mechanism, researchers have found that the attention mechanism is beneficial for feature aggregation with only a small increment in parameters and computation. Inspired by this, we consider adding an attention mechanism in a light-weight network to improve the accuracy and maintain a high segmentation speed at the same time. Therefore, this paper proposes a real-time image semantic segmentation network based on dual efficient attention mechanism (DEANet). Among them, CPA (Channel Pyramid Attention Module) and SEA (Spatial Efficient Attention Module) are used to refine the results of the backbone network. Experimental results show that this method not only achieves high segmentation accuracy, but also has high segmentation speed.

The main contributions of this paper are summarized as follow:

This paper proposes a dual efficient attention module. In the channel dimension, the pyramid sampling is introduced to collect multiscale information. In the spatial dimension, using high-resolution feature maps as input, so that features can be aggregated together efficiently and the refine segmentation results can be got.

Based on the dual efficient attention module, a real-time image semantic segmentation network (DEANet) is proposed. The results on the Cityscapes and CamVid datasets show that DEANet has significantly improved the segmentation speed and segmentation accuracy while maintaining similar performance in accuracy and speed.

2 Related Work

After the emergence of convolutional neural network, it has been highly applied in the field of image processing. In 2015, Shelhamer et al. proposed FCN to classify images at the pixel level. Compared with traditional methods, the segmentation results are significantly improved. From then on, the semantic segmentation technology has entered the era of deep learning.

2.1 Real-Time Semantic Segmentation

Recently, real-time semantic segmentation task has attracted a lot of attention. The real-time semantic segmentation method applied to images starts with SegNet [6] proposed by Kendall et al. It encodes the effective information through the encoder, and then decodes it through the decoder to achieve image segmentation. Compared with AlexNet [7], VGG [8], ResNet [2] and other networks, in terms of saving expenses and increasing the nonlinear fitting ability of the neural network, the neural network has been deeply designed. However, the more complex the network is, the more computational operations and network parameters are required, so the image processing speed is greatly affected. To address the speed issue, Adam et al. proposed an efficient neural network ENet [4].

Different with SegNet, ENet adopts a completely symmetric encoder-decoder structure. Adam et al. considered that decoding only up-samples the outcome of the encoder, so the design of the decoder is relatively simple. Therefor the overall network structure parameters are reduced accordingly, and the speed of processing images is improved. Yu et al. proposed BiSeNet [9]. The Spatial Path is used to extract feature maps to obtain accurate information, and the Context Path is used for fast down-sampling. Through the fusion module, the features output by the two are combined. The BiSeNet not only improves the segmentation speed, but also improves the segmentation effect.

With the development of computer vision, many application scenarios require more accurate segmentation results in a shorter response time. However, segmentation speed and segmentation accuracy are a pair of opposing indicators. So, it is difficult to make both of them achieve satisfactory results at the same time. Although many works have been done by researchers, there is still much room for improvement.

2.2 Attention Mechanism

Attention mechanisms are highly applied in computer vision tasks. And this is helpful for semantic segmentation. Hu et al. proposed a channel attention module in SENet [11]. Zhu et al. proposed ANNNet [13]. The network adopts two modules, APNB and AFNB. The APNB module embeds SPP nonlocality to enhance the global representation and reduce computational overhead. The AFNB module is used to integrate different levels of functions. Tian et al. proposed DANet [14] for scene segmentation. Two attention modules are added to the FCN with dilated convolution. PAM is used to learn the dependencies on spatial features, and CAM is used to learn the dependencies on the channel dimension.

The fundamental of the attention module is to give the same weights to the same areas in the feature map, so that it could collect the useful information in the feature map, while suppressing the useless information. On the other hand, attention module could obtain global context information and this is available to improve the segmentation accuracy. In this paper, we proposed a dual efficient attention module. In the channel dimension, we propose to combine the attention module and pyramid sampling to collect multi-scale information. In the spatial dimension, high-resolution features are efficiently aggregated for inferring segmentation results.

3 Methodology

In this chapter, we will recommend the ensemble framework of DEANet, the attention modules, and how they are aggregated together for fine-grained semantic segmentation.

3.1 Network Structure of DEANet

Most of the real-time semantic segmentation neural networks use the structure of encoder-decoder, such as ERFNet [5]. The encoder extracts feature and the decoder decodes feature to realize segmentation. As the attention mechanism is highly applied in computer vision tasks, researchers have found that the attention module can extract features more conveniently. Moreover, as the ERFNet is a lightweight network and the attention module adds only a small amount of computation, so the DEANet could maintain a high image processing speed. The ensemble framework of DEANet is shown in Fig. 1.

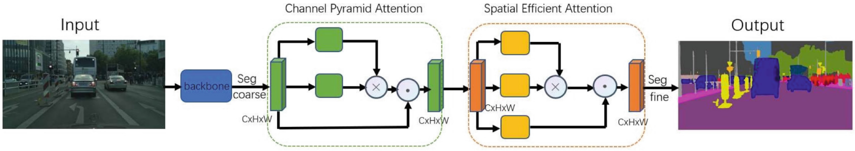


Fig. 1. The overall framework of DEANet.

In this paper, the encoder of ERFNet is used as the backbone of DEANet, and the dual efficient attention module is proposed to replace the decoder based on encoder-decoder. As shown in Fig. 1 above, the image input is used to extract the feature map through the encoder, and the feature map gets the Seg_{coarse} through the classification layer. Then the Seg_{coarse} are input into the dual efficient attention module for feature extraction. Finally, the Seg_{fine} is obtained. As shown in Fig. 1 above, the attention module consists of two parts, CPA module and SEA module. Pass through CPA and SEA module in turn to perform feature aggregation and finally achieves fine segmentation result. The process can be formulated as follows:

$$x = Cla(X) \tag{1}$$

$$y = Channel\,Attention(x) \tag{2}$$

$$Seg_{fine} = Spatial\,Attention(y) \tag{3}$$

Among them, x represents the feature maps obtained by the backbone network, $Cla()$ represents the classification layer in the backbone, x is the coarse segmentation result, y is the aggregated feature obtained by the CPA module, and Seg_{fine} indicates the final fine segmentation result obtained by the SEA module. According to the theoretical analysis, by removing the decoder part instead of using the attention module, DEANet can achieve finer segmentation results and improve the efficiency. And in the framework, the encoder in any encoder-decoder network can be used as the backbone. Because ERFNet is a lightweight semantic segmentation network, and it achieves a fine balance between segmentation accuracy and speed. In this paper, we choose the ERFNet as the baseline model. The encoder part of DEANet comes from ERFNet, and the decoder part is replaced by the CPA module and the SEA module proposed in this paper.

3.2 Attention Module

In this semantic segmentation framework, pyramid sampling is introduced into the channel dimension to extract multi-scale information. The channel pyramid attention module can overcome the shortcomings of the traditional convolution layer and extract global information. Using down-sampling in SEA can consume lesser computing resources without reducing the attention resolution. In the design of attention module, attention module is used to replace the up-sampling module in the original ERFNet to improve the accuracy without adding additional parameters. A better balance between speed and accuracy can be found by adjusting the pyramid pooling in CPA and the super parameters in down-sampling in SEA. In this paper, we refer to the channel spatial attention mechanism proposed by Tian et al. [14]. They proposed two modules, which introduced global information from the spatial attention mechanism and channel attention mechanism respectively. And the local features and global dependencies can be integrated adaptively.

Channel Pyramid Attention Module

We believe that in the current deep neural network, a series of convolutional layers constitute the main architecture of the neural network. The convolution kernel of each layer of the network performs convolution operations on the input of this layer. And this makes the output feature map can only sense the information of the surrounding points corresponding to the input feature map point. Only when the number of network layers is deep enough, the receptive field of the feature map will be large enough. This makes the neural network have obvious shortcomings when integrating nonlocality information. In deep neural networks, such as VGGNet, compared with AlexNet, they are composed of convolution layer and fully connected layers. The difference is that VGGNet explores the relationship between depth and performance in convolutional neural networks. For a given receptive field, the small convolution kernel with stacking is better than the large convolution kernel. And while not increasing the number of parameters, we can improve the performance by deepening the network structure.

In [10], Wang et al. proposed the non-local block, in which the features of all relevant positions are weighted while calculating a certain position. And this helps the deep neural network to integrate non-local information effectively. In [12], Li et al. proposed (PAN), in which the attention module and the spatial pyramid were combined together to construct FPA. The FPA can be used to obtain the dense feature and semantic information. Inspired by these works, in the DEANet proposed in this paper, we add pyramid pooling to the non-local module to make the most of the global context information and provide sufficient feature information for global semantic scene.

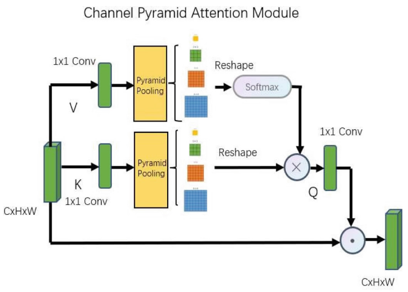


Fig. 2. Channel pyramid attention module.

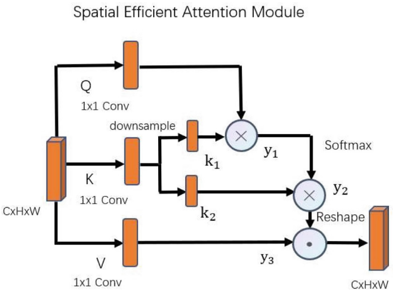


Fig. 3. Spatial efficient attention module.

The coarse segmentation result is treated as the input of the CPA module (as shown in Fig. 2) which is denoted as $x \in R^{C \times H \times W}$. C represents the number of channels, H represents the height of the image, and W represents the width of the image. Two new features are generated by two convolution operations, where the features generated by K and V are $x_1 \in R^{C/2 \times H \times W}$, $x_2 \in R^{1 \times H \times W}$, respectively. And then they are input into the pyramid pooling module. The pyramid pooling module combines the features of 4 different pyramid scales. In the network of this paper, the compression ratio is set to 1, 3, 6, and 8. The outputs of different levels have different scales. And the feature maps of different scales provide multi-scale information. Then the 4 pooling results are concatenated as the input of the next layer. After processing of the CPA module, the output is shown in Eq. 4:

$$\begin{aligned} \mathrm{y} &= \mathit{Channel\ Attention}(x) \\ &= \{\mathit{softmax}(\alpha(x_1) \otimes \alpha(x_2)\} \odot x \end{aligned} \tag{4}$$

where α represents the *reshape* operation and *softmax* represents the calculation probability operation, $\otimes$ represents the matrix multiplication, $\odot$ represents the Channel-Based Multiplication. The matrix multiplication is performed between the transformed features x_1, x_2 to obtain a new feature $Q \in R^{C/2 \times 1 \times 1}$. Finally, a channel-based multiplication operation is performed with the initial feature input $x \in R^{C \times H \times W}$ to obtain the channel attention result denoted as y.

Spatial Efficient Attention Module

In [20], Fan et al. proposed the STDC module, in which is used to collect deep features with retractable receptive fields and multi-scale information. Inspired by STDC [20], we use down-sampling in the SEA module to reduce the number of feature channels while maintaining high attention resolution in the feature input dimension. Improve the accuracy and efficiency of segmentation with less computational resource consumption.

As shown in Fig. 3, the SEA module takes the output of the CPA module $y \in R^{C \times H \times W}$ as the input. New featuresq, k, v are obtained through three 1x1 convolution operations. Among them, the feature k is down-sampled to obtain $k_1 \in R^{C/2 \times H \times W}$, $k_2 \in R^{C/2 \times H \times W}$. In this process, the feature dimension is reduced from $C \times H \times W$ to $C/2 \times H \times W$. The purpose of this is to deepen the network depth with less computational resources while maintaining a high attention resolution. Then perform matrix multiplication on the features q and k_1 to obtain the aggregated feature y_1 as shown in Eq. 5:

$$y_1 = q \otimes k_1 \tag{5}$$

Then the aggregated feature y_1 is processed by the *softmax* operation and perfume the matrix multiplication with k_2 to obtain a new aggregated feature y_2, as shown in Eq. 6:

$$y_2 = softmax(y_1) \otimes k_2 \tag{6}$$

After the reshape operation, y_2 performs a space-based multiplication operation with the feature $v \in R^{C \times H \times W}$ and finally obtains the fine segmentation result Seg_{fine} as shown in Eq. 7:

$$Seg_{fine} = \alpha(y_2) \odot y_3 \tag{7}$$

4 Experiments

We have carried out many experiments on the public data sets Cityscapes and CamVid to verify the validity of the method proposed in this paper. The final results show that DEANet proposed in this paper not only maintains a high speed, but also makes a significant improvement in accuracy. Next, the dataset, experimental details, ablation experiments and comparative experiments will be introduced in detail.

4.1 Dataset

Cityscapes [16]: The Cityscapes has two evaluation criteria datasets: fine dataset and coarse dataset. The former contains 5000 finely labeled images, and the latter contains 5000 fine labeled images and 20000 coarse labeled images. For equitable comparison, we apply fine dataset as evaluation criteria in our experiments.

CamVid [17]: CamVid is video collection with target semantic tags, from which more than 700 images can be specified for pixel level semantic segmentation.

4.2 Evaluation Indicators and Experimental Details

All evaluation indicators are based on a RTX2080Ti graphics card and Ubuntu operating system. We implement our approach based on Pytorch. IOU (Intersection over Union), the ratio of intersection and union. In semantic segmentation, the intersection ratio is the ratio of the intersection and union of ground truth labels and predicted values. mIoU (Mean Intersection over Union) is the average of the intersection ratio of each class in the dataset. In the training process, the data will be enhanced by random horizontal flip and random scale transformation. To verify the validity of the module, we use the same hyperparameter settings as the original network. For the Cityscapes dataset we employ SGD to optimize our network. Num_works is set to 4, batch_size is set to 8, the number of training epochs is set to 2000, and 512 × 1024 high-resolution original images are randomly cropped as training input. For the CamVid dataset the Adam optimizer is applied and the network has been trained for 400 epochs. The size of input image is 360 × 480.

4.3 Ablation Experiment

The Validity Verification of the Redesigned Attention Module

To verify the validity of the dual efficient attention module in this paper, we use ERFNet as the backbone and the PSA module [15] proposed by Liu et al. as the reference. The performances of some different combinations of channel-space attention modules are compared. The polarized self-attention module PSA proposed by Liu et al. includes two branches. One branch is channel-dimensional self-attention (CSA for short), and the other branch is spatial-dimensional self-attention (SSA for short). The above two self-attention mechanisms can form four different combinations with the channel-dimension pyramid attention (CPA for short) and the spatial-dimension efficient attention (SEA for short) proposed in this paper.

Table 1. mIoU on the Cityscapes test set, √ means adopting this module, × means abandoning this module.

Method	CSA	SSA	CPA (ours)	SEA (ours)	mIoU
PSANet	√	√	×	×	70.68
CSENet	√	×	×	√	72.65
SCPNet	×	√	√	×	73.47
DEANet	×	×	√	√	74.90

Compared with self-attention module, the pyramid attention module in channel dimension and the efficient attention module in spatial dimension both improve the experimental results to a certain extent (1.97% and 2.79% respectively). This is because adding pyramid pooling in the channel modules can fully utilize the global context information. For the semantic segmentation task, global context information and multi- scale information provides rich semantic information and preserves detailed semantic features. Adding down-sampling to the spatial attention module while maintaining a higher resolution could reduce computational resource consumption and improve the segmentation accuracy. Moreover, the experimental results achieved the greatest improvement when the CPA module and the SEA module were used meanwhile (compared to the self-attention module by 4.22%). This shows that the two attention modules proposed in this paper have complementary functions (Table 1).

Network Performance Verification Under Different Structures

To verify the effects of CPA module and SEA module on network performance under different structures, we validate dual attention modules with 3 different structures on the same experimental setup. Structure 1: The two attention mechanisms use a parallel structure; Structure 2: The two attention mechanisms use a serial structure, with spatial attention in the front and channel attention in the back; Structure 3: The two attention mechanisms use a serial structure, the CPA is in the front, and the SEA is in the back.

Table 2. mIoU on the Cityscapes test set,1 means before, 2 means after.

Structure	CPA	SEA	mIoU
Parallel	√	√	72.32
Serial	2	1	73.48
Serial	1	2	74.90

After experimental comparison, we found that the two attention mechanisms adopt a serial structure, and the CPA module is in the front and the SEA module is behind, the optimal results are obtained. This is because when the channel attention is in the front, the weight of each channel will be calculated first, and the key information channels will be aggregated. Thereby the ability of feature representation is improved. Based on the channel attention, the spatial attention module weights and integrates the features based on the channel direction. And the feature aggregation is further improved (Table 2).

Validation of the Reconstruction of the Network Structure

To verify the validity of the lightweight segmentation network constructed based on dual efficient attention module (DEA) proposed in this paper, we compare the representation of 3 neural networks with diverse structures on the Cityscapes validation set. Accuracy-speed trials including the complete body ERFNet, ERFNet encoder with classification layer, and DEANet using ERFNet encoder as backbone.

Table 3. Comparison of accuracy and speed on Cityscapes.

Model	Parameters	Speed(ms)	FPS	mIoU
ERFNet-base	2.067 M	11.76	85	72.46
ERFNet-enc	1.876 M	8.20	121.95	70.60
DEANet-ERF	1.877 M	10.01	99.91	75.78

Table 3 shows that the ERFNet encoder with a classification layer is faster than the complete body ERFNet, but the lack of a decoder causes a large drop in segmentation accuracy. Compared with the baseline, DEANet has improved the accuracy by 3.32 percentage points, and the speed has also improved. The visualization results of the Cityscapes dataset are shown in Fig. 4. It can be seen from Fig. 4 that the network in this paper can better deal with details.

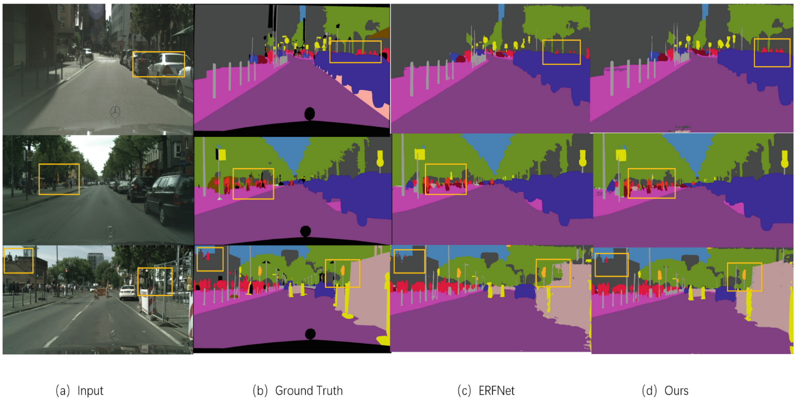


Fig. 4. Visualizing the results on the Cityscapes dataset.

4.4 Comparison with Other Methods

Our previous series of experiments have demonstrated that DEANet can improve the current image real-time semantic segmentation methods. Both in speed and precision. Next in this subsection, we compare DEANet with the current state-of-the-art models on the Cityscapes dataset. All results are obtained from the experimental results of the official website of the Cityscapes dataset or the author's paper.

Table 4. Comparison with state-of-the-art results on Cityscapes dataset.

Model	Pretrain	InputSize	Parameters	FPS	mIoU
ENet [4]	ImageNet	512 × 1024	0.37 M	76.9	58.3
ERFNet [5]	No	512 × 1024	2.07 M	41.7	68.0
DABNet [18]	No	512 × 1024	0.76 M	104.2	70.1
BiseNet [9]	No	1563 × 768	55.3 M-	45.7	73.6
EDANet [19]	No	512 × 1024	0.68 M	108.7	67.3
FRFNet [23]	No	512 × 1024	4.02 M	225	68.2
BiseNetv2 [22]	No	512 × 1024	–	156	72.6
STDC2-Seg50 [20]	No	512 × 1024	12.5 M	188.6	73.4
CSRNet [21]	No	512 × 1024	–	56	74.0
PP-LiteSeg-T1 [24]	No	512 × 1024	–	273.6	72.0
DEANet	No	512 × 1024	1.88 M	99.91	74.90

As shown in Table 4, we show the three metrics including parameter volume, inference speed and segmentation accuracy. When the input size of DEANet is 512 × 1024, the processing speed can reach 99.91FPS while the mIoU reaches 74.90%. Compared with other networks with similar performance in segmentation speed, the accuracy has been significantly improved. For example, compared with DABNet [18], the speed is 104.2FPS compared to 99.91 FPS, and the accuracy is 70.1% mIoU compared to 74.90% mIoU. In the newly proposed STDC2-Seg50 [20] network, an STDC module is designed to remove structural redundancy and improve network performance at a certain computational cost. Compared to the STDC2-Seg50 [20] network, our network parameters are reduced to only 1.88 M and the segmentation accuracy is also improved by 1.5 percentage points. Compared with CSRNet [21], the segmentation accuracy of our network is improved by 0.9 percentage points and the speed is enhanced by nearly 50 FPS.

Table 5. Comparison with state-of-the-art results on CamVid dataset.

Model	InputSize	Parameters	FPS	mIoU
ENet [4]	360 × 480	0.37 M	111	51.3
ERFNet [5]	360 × 480	2.07 M	133	65.0
DABNet [18]	360 × 480	0.76 M	104	66.4
EDANet [19]	360 × 480	0.68 M	–	66.4
FRFNet [23]	360 × 480	4.02 M	225	68.2
DEANet	360 × 480	1.88 M	142.72	70.07

We also evaluate DEANet on the CamVid dataset. We show the three indicators of parameter volume, inference speed and segmentation accuracy in Table 5. As can be seen from Table 5, the DEANet in this paper has improved in speed and accuracy. For example, compared with ERFNet [5], the speed improves 9.72 FPS and the accuracy improves 5.07%. Compared with FRFNet [23], although our speed is reduced, our parameter volume is reduced and the accuracy is improved.

5 Summary

In this paper, we have proposed a real-time image semantic segmentation network based on a dual efficient attention module named as DEANet. The proposed dual efficient attention module is more suitable for image semantic segmentation and can perform feature aggregation more efficiently. In addition, a lightweight backbone network is used to improve the image processing speed while improving the segmentation accuracy. We have carried out many experiments on the Cityscapes and CamVid datasets and better segmentation results were obtained under a high segmentation speed.

In the next, we will continue to explore real-time semantic segmentation tasks, but pay more attention to segmentation speed. Try a lighter backbone network to increase segmentation speed without losing accuracy.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (Grant No. U1908214), Special Project of Central Government Guiding Local Science and Technology Development (Grant No. 2021JH6/10500140), Program for the Liaoning Distinguished Professor, Program for Innovative Research Team in University of Liaoning Province (Grant No. LT2020015), the Support Plan for Key Field Innovation Team of Dalian (2021RT06), the Science and Technology Innovation Fund of Dalian (Grant No. 2020JJ25CY001), the Support Plan for Leading Innovation Team of Dalian University (XLJ202010), Dalian University Scientific Research Platform Project (No. 202101YB03).

References

1. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (2015)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
3. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI: IEEE, pp. 6230–6239 (2017)
4. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: ENet: a deep neural network architecture for real-time semantic segmentation. arXiv:1606.02147 [cs] (2016)
5. Romera, E., Álvarez, J.M., Bergasa, L.M., Arroyo, R.: ERFNet: efficient residual factorized convnet for real-time semantic segmentation. IEEE Trans. Intell. Transp. Syst. **19**(1), 263–272 (2018)
6. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
8. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
9. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: BiSeNet: bilateral segmentation network for real-time semantic segmentation, pp. 325–341 (2018)
10. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks, pp. 7794–7803 (2018)
11. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks, pp. 7132–7141 (2018)
12. Lih, C., Xiong, P.F., An, J., et al.: Pyramid attention network for semantic segmentation. arXiv:1805.10180 (2018)
13. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 593–602 (2019)
14. Fu, J., et al.: Dual attention network for scene segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3141–3149 (2019)
15. Liu, H., Liu, F., Fan, X., Huang, D.: Polarized self-attention: towards high-quality pixel-wise regression. Arxiv Pre-Print arXiv:2107.00782 (2021)
16. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
17. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88682-2_5

18. Li, G., Yun, I., Kim, J., et al.: Dabnet: depth-wise asymmetric bottleneck for real-time semantic segmentation. arXiv preprint arXiv:1907.11357 (2019)
19. Lo, S.Y., Hang, H.M., Chan, S.W., et al.: Efficient dense modules of asymmetric convolution for real-time semantic segmentation. In: Proceedings of the ACM Multimedia Asia, pp. 1–6 (2019)
20. Fan, M., Lai, S., Huang, J., et al.: Rethinking bisenet for real-time semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9716–9725 (2021)
21. Xiong, J., Po, L.M., Yu, W.Y., et al.: CSRNet: cascaded selective resolution network for real-time semantic segmentation. arXiv preprint arXiv:2106.04400 (2021)
22. Yu, C., Gao, C., Wang, J., et al.: Bisenet v2: bilateral network with guided aggregation for real- time semantic segmentation. Int. J. Comput. Vision **129**(11), 3051–3068 (2021)
23. Sixiang, T.: Feature Reuse and Fusion for Real-time Semantic segmentation. arXiv preprint arXiv:2105.12964 (2021)
24. Peng, J., Liu, Y., Tang, S., et al.: PP-LiteSeg: a superior real-time semantic segmentation model. arXiv preprint arXiv:2204.02681 (2022)

A Deep Learning Approach Based on Continuous Wavelet Transform Towards Fall Detection

Yingwen Chen[1], Yuting Wei[1], Deming Pang[1](✉), and Guangtao Xue[2]

[1] National University of Defense Technology, Changsha 410015, China
pang3724@nudt.edu.cn
[2] Shanghai Jiao Tong University, Shanghai 200030, China
xue-gt@cs.sjtu.edu.cn

Abstract. In this paper, we investigate device-free fall detection based on wireless channel state information (CSI). Here, we mainly propose a method that uses continuous wavelet transform (CWT) to generate images and then uses transform learning of convolutional networks for classification. In addition, we add a wavelet scattering network to automatically extract features and classify them using a long and short-term memory network (LSTM), which can increase the interpretability and reduce the computational complexity of the system. After applying these methods to wireless sensing technology, both methods have a higher accuracy rate. The first method can cope with the problem of degraded sensing performance when the environment is not exactly the same, and the second method has more stable sensing performance.

Keywords: Intelligent wireless sensing · Fall detection · Continuous wavelet transform · Deep learning

1 Introduction

Perception of the physical world has entered a new stage of ubiquitous intelligence. The Internet of Things and artificial intelligence technology together promote the human society from the interconnection of everything to the intelligent connection of everything [1]. While sensors get smaller and data collection is widely used, a problem emerges: due to the high deployment cost of sensing systems and the increased range and scale of sensing, it is difficult and costly to deploy and maintain large-scale sensing systems for long-term stable operation. So is it possible to sense various environmental information without deploying any dedicated sensors? Currently, Wireless networks are widely used to sense the environment. It is possible to use wireless sensing technology to sense human behavior and also to detect the health and behavioral understanding of people, etc. [2–4].

Y. Chen and Y. Wei—Contribute equally to this work.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 206–217, 2022.
https://doi.org/10.1007/978-3-031-19214-2_17

Wireless sensing technology is mainly used to perceive the scene by analyzing the changes of wireless signals during propagation and obtaining the characteristics of the signal propagation space [5]. RF signals, commonly used in wireless sensing technologies, are mainly generated by radio waves from a signal transmitter, and during the propagation of the RF signals, physical phenomena such as direct transmission, reflection as well as scattering often occur, resulting in multiple propagation paths. This makes the multipath superimposed signal formed at the signal receiver carry information reflecting the signal propagation space. Previously, received signal strength (RSSI) was widely used in scene perception. For example, when we use a cell phone, the signal is strong when we are close to the base station. This allowed us to infer the location of the transmitter and the environment it was in, but RSSI is a coarse-grained piece of information with limited accuracy. Channel state information (CSI) is now commonly used in the perceived environment [6]. CSI is a fine-grained information in WIFI communication, which describes how the signal propagates in the wireless channel by combining the effects of time delay, energy fading and phase shift. There are 30 subcarriers in each packet, containing amplitude and phase information. Different subcarriers correspond to the amplitude and phase of multipath propagation at different frequencies [7]. However, CSI phase information usually contains a lot of noise so it needs to be processed before using.

There has been abundant recent research using WiFi CSI for sensing human activities and has been applied to various fields such as healthcare and security. It has been reported that the aging population has been growing at a rapid rate, and people over 65 years old account for about one-third of the global population. One of the main causes of accidental death among the elderly is the lack of timely assistance after a fall, especially since more and more elderly people are living alone nowadays. Thus, it is necessary to design a reliable fall detection system. Currently, there are many studies on fall detection using cameras and sensors in addition to device-free sensing. However, cameras require light and have the disadvantage of invading people's privacy. In addition, sensors need to be carried around, and elderly people might forget to wear them. Using device-free sensing for falls can be a good solution to the above problems.

The current research on fall detection based on WiFi CSI has made great progress [8–10], but there are still some shortcomings. There are two main problems: one is that most of the research is still processing classification on the study of time-series data, which will make the classification less interpretable and the extracted features less comprehensive; the second is that the sensing performance will be significantly reduced when some changes occur in the environment where the target is located.

Aiming at overcoming the above challenges, in this paper, we design a method based on wavelet analysis and deep learning to detect falls. First, due to the large amount of noise in the original CSI, we have to preprocess the data. Then the processed data is used to generate images using continuous wavelet transform, and finally a convolutional neural network with transform learning [11] is used to classify these images. In addition, we add a wavelet scattering network approach that increases the interpretability and reduces the computational complexity of the system.

2 Related Work

2.1 Fall Detection

At present, WiFi-based CSI for fall detection systems are Wifall [8], RTFall [9], and FallDeFi [10]. The first two mainly extract time-domain features to detect falls while the latter one detect falls by extracting time-frequency features. Compared with pure time domain or pure frequency domain features, the time-frequency domain features contain both time and frequency domain information, bringing more significant advantages. FallDeFi proposes a power burst curve (PBC) to pre-screen fall behavior before using a support vector machine (SVM) to determine whether it is a fall behavior or not.

2.2 Image Classification

Classification of time-series data has now been widely studied, and one of them is called visibility algorithm, which converts time series into graphs [12]. The advantage of the algorithm is that both global and local features can be considered. We apply this idea to device-free fall detection by first converting to an image and then performing image classification. Image classification is the classification of different images into different categories to achieve the minimum classification error [13]. The models that are now widely used are GoogLeNet and ResNet. GoogLeNet adopts the structure of Inception, which is to put multiple convolution or pooling operations together to assemble a network module and to design neural networks to assemble the whole network structure as a module. GoogLeNet uses multiple convolutional kernels to extract information from different scales of the image and then fuse the image to obtain a better image representation [14]. ResNet mainly alleviates the problem of gradient dissipation in neural networks by connecting across layers, which enables the training of multi-layer networks [15].

3 Methods

3.1 System Overview

The system is divided into four parts: data collection, data pre-processing, feature extraction and classification, as shown in Fig. 1. The system deploys a pair of transceivers then uses an Intel 5300 NIC to collect raw CSI. In the data pre-processing part, we first perform linear interpolation and use CSI ratio [16] to eliminate carrier bias, and then perform denoising with discrete wavelet transform to obtain clear and rich information. Following that, we analyze the correlation between subcarriers. Next, we propose a method of image generation using continuous wavelet transform to generate images for extracting features. Finally, transform learning using convolutional neural networks is used to classify falls. In addition, a wavelet scattering network approach is added, which can automatically extract features to reduce computational complexity, but is more dependent on environment.

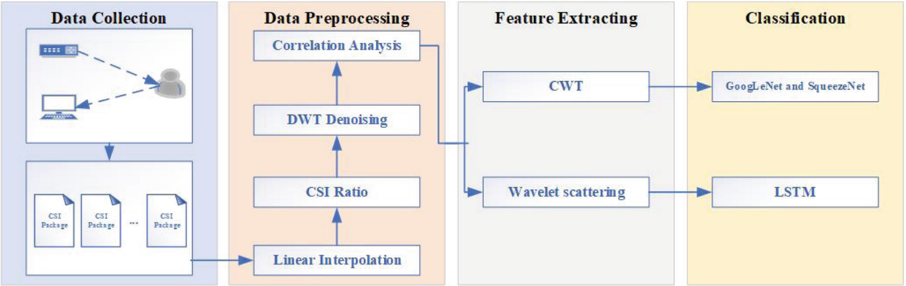


Fig. 1. Framework of system.

3.2 Data Preprocessing

Weak signals in some links due to non-line of sight connections usually lead to some packet loss when collecting data. To solve this problem, we usually use a linear interpolation method to make all recording channels have the same sampling rate. After linear interpolation, the CSI of the two antennas with the highest power is selected to do the ratio, which can eliminate the high impulse noise and burst noise that are difficult to eliminate in the original CSI amplitude. The obtained signal is then used with discrete wavelet transform (DWT) to eliminate the in-band noise. By discrete wavelet transform, the signal will be decomposed into several frequency levels and we can get the frequency level we need. The highest frequency stage contains mainly noise. DWT works mainly by first estimating the threshold of the level, adjusting the threshold to a lower frequency level and then removing the noise at all wavelet levels without significant distortion of the signal components. Finally, the denoised signal in the wavelet domain is transformed back to the time domain to obtain the information segment we need. Since not every subcarrier is representative, we next analyze the correlation between the subcarriers. There are 30 subcarriers in each packet and we observe the distribution of each subcarrier. The data distribution of each subcarrier is essentially the same and there are some subcarriers with higher correlations in both fall and non-fall actions. Based on the results of the analysis of the two correlations, it is found that the best results are obtained with both correlation thresholds set to 0.7.

3.3 Feature Extraction

After data preprocessing, we use continuous wavelet transform (CWT) to generate scale maps and then perform image classification. Some systems use Short Time Fourier Transform (STFT) for time-frequency feature extraction, which is a windowed Fourier transform that decomposes the entire time-domain process into an infinite number of small processes of equal length and each small process is approximately smooth in the Fourier transform. The definition of CWT is shown in Eq. (1). The CWT identifies the frequency components of the signal. The main reason for using CWT instead of STFT here is that STFT's window size is not easy to set. CWT can determine the signal frequency as well as its corresponding time interval and can tell the magnitude of the frequency from the thickness of the stripes. So here CWT is used to do the time-frequency transformation.

$$W(a,b) = \int_{-\infty}^{\infty} x(t)\frac{1}{\sqrt{a}}\psi(\frac{t-b}{a})dt \tag{1}$$

We firstly compute a filter bank of CWT, obtain the CWT of the signal and give the coefficients to obtain its scale map. Figure 2 show the spectrogram of a single subcarrier for a fall action and a unfall action. From the figure, we can see that the scale maps generated for the two different actions are clearly different. In the unfall action, the thin stripes at low frequencies indicate a higher frequency resolution and the stripes span a larger interval in accordance with the unfall action characteristics. In the fall action, the presence of high-frequency streak segments with shorter streak spanning intervals indicates that a short-duration high-frequency event, i.e., a fall, occurred at this time. After generating scale maps, RGB images are created for them and these images are used as input for classification.

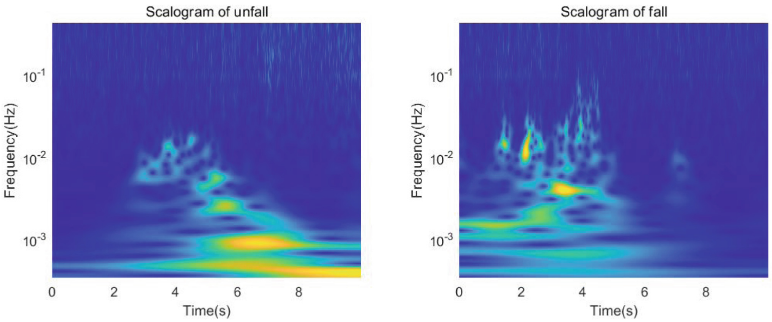


Fig. 2. Scale maps of a single subcarrier for a unfall (left) and a fall (right).

3.4 Classification

Neural network models of deep learning have also been widely used in the field of wireless sensing. When faced with a specific problem in a domain, it is difficult to find enough training data. Therefore, it has been investigated that if we can use models trained from other data sets and then modify and refine the models according to the specific problem, we can use them repeatedly to solve the problem of insufficient amount of data. The technique to solve this problem is transform learning, which is to transfer the model parameters trained to a new model to help the new model training. Because most of the data or tasks are correlated, the learning speed of the new model can be accelerated by transfer learning. For example, it is much easier to learn to use an electric bike when we have already learned a bicycle. The first few layers of the network of transfer learning can reveal the contour of the image, which can be used to identify the image generated after wavelet transform. Here we use transformation learning based GoogLeNet and SqueezeNet for classification.

When using GoogLeNet, we firstly need to modify the network parameters on top of the original network. Because we need to merge the features extracted form the network into different information types such as category probabilities, loss values and predictive labels, we need to replace some layers in the original network with new layers that fit our data. The main goal of GoogLeNet is to improve the accuracy of recognition, so it takes the approach of deepening the network structure and enhancing the functionality of the convolution module, which will lead to an increasingly complex network and memory. SqueezeNet can maximize the computational speed without reducing the accuracy of the model, and the number of parameters is reduced by a factor of tens. The innovation of SqueezeNet in both squeezing and expanding, which can reduce the dimensionality of the feature map. SqueezeNet is used here instead of ResNet, because SqueezeNet is optimized for ResNet and a better structural design reduces the size of the network and the associated parameters without the need for complex compression algorithms. Again, the use of SqueezeNet has to modify the network parameters according to our data.

3.5 Wavelet Scattering Network

We additionally add a wavelet scattering network approach that consumes less memory and reduces computational complexity. This method is identical to the continuous wavelet transform image generation method in terms of data preprocessing, and then feature extraction is performed using the wavelet scattering network method. Wavelet scattering is a technique that can be used to automatically extract low variance and compact features that minimize intra-class variation while preserving the distinguishability between individual classes. The main procedure is to first construct a wavelet time scattering network with a filter bank. After that, the scattering coefficients of the training data are represented in matrix form. These multi-signal scattering transforms are then reconstructed into a matrix as the input to the LSTM classifier. LSTM is a recurrent neural network, which is usually used to study time series data. It learns long-term correlations between time steps of time-series data and is suitable for analyzing time-series data.

4 Performance Evaluation

4.1 Experiment Setup

Dataset. The data collected by FallDeFi [10] is used for the experimental data in this paper. The experimental environment for its data collection was selected from five experimental areas: two bedrooms, a hallway, a kitchen, a bathroom and a laboratory. The data were collected by transmitters sending 100B packets for a duration of 10 s at a data rate of 800 kb/s. 10,000 packets were collected for each activity, corresponding to 30 * 10000 CSI values per antenna pair. All experiments were performed on the 5.2 GHz frequency band. The data from the corridor, bedroom, kitchen and bathroom were divided into two groups, A and B. There were some differences between groups A and B in terms of days or equipment settings. For example, the collection interval between group B and group A data in the corridor environment is 7 days and the transmitter moves 0.5 m from the original distance; the collection interval between group B and group A data in the bedroom environment is 33 days and there is one more person; the collection interval between group B and group A data in the kitchen environment is 33 days and there is another person, in addition to moving non-line of sight furniture; in the bathroom environment the laboratory was used as the experimental environment for the robustness study with only one set of data. The data used in our experiments are shown in Table 1.

Table 1. Data collection

Group	Measurements	Corridor	Bedroom	Kitchen	Bathroom	Lab	Total
A	*Falls*	*33*	*21*	*39*	*40*	*45*	*178*
	Others	*21*	*29*	*43*	*33*	*35*	*161*
B	*Falls*	*21*	*28*	*30*	*30*	–	*109*
	Others	*28*	*35*	*31*	*36*	–	*130*
Changes from A to B	*Diff. in days*	*7*	*33*	*33*	*10*		
	Diff. in environment	*Tx. moved by 0.5* m	*+1 person*	*+1 person, furniture moved nLoS*	*+1 person, Tx. moved by 0.5* m		

Evaluation Metrics. For the evaluation metrics of the results the accuracy is mainly used here. A test result that matches the true result is defined as a correct prediction (CP), and a test result that does not match the true result is defined as an incorrect prediction (IP). Then accuracy is defined as shown in Eq. (2).

$$Accuracy = \frac{CP}{CP + IP}, \tag{2}$$

4.2 Comparison of the Two Methods

Result of the First Method. The first method performs continuous wavelet transform after data acquisition and preprocessing to generate scale maps and then uses pre-trained convolutional networks for classification. Here mainly GoogLeNet and SqueezeNet are used. The experimental results of these two networks are shown in Fig. 3 and Fig. 4.

As can be seen from the figures, both networks perform well on a single data set, with results above 95% for almost all environments and above 90% for a few individual ones. When the environment changes, the detection results of fall to action are not as good as the single data set, but still slightly improve overall. The performance of the two networks of the first approach does not differ much in the same dataset. However, when the environment changes, the results of SqueezeNet network outperform GoogLeNet, indicating that SqueezeNet is less dependent on the environment. However, the detection results of GoogLeNet are more stable than those of SqueezeNet, and the results fluctuate within 10%, while the detection results of SqueezeNet sometimes vary widely and therefore it requires several experiments.

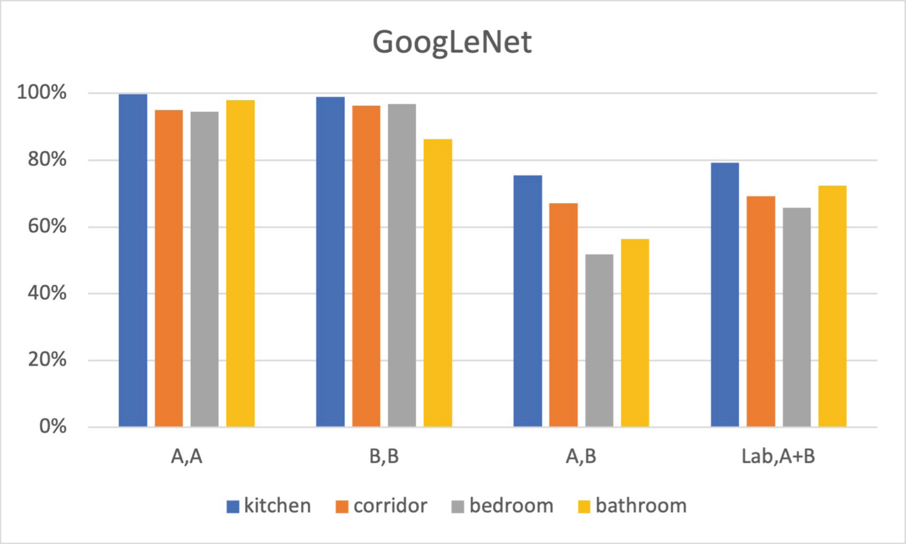


Fig. 3. Results of performance GoogLeNet across different environments.

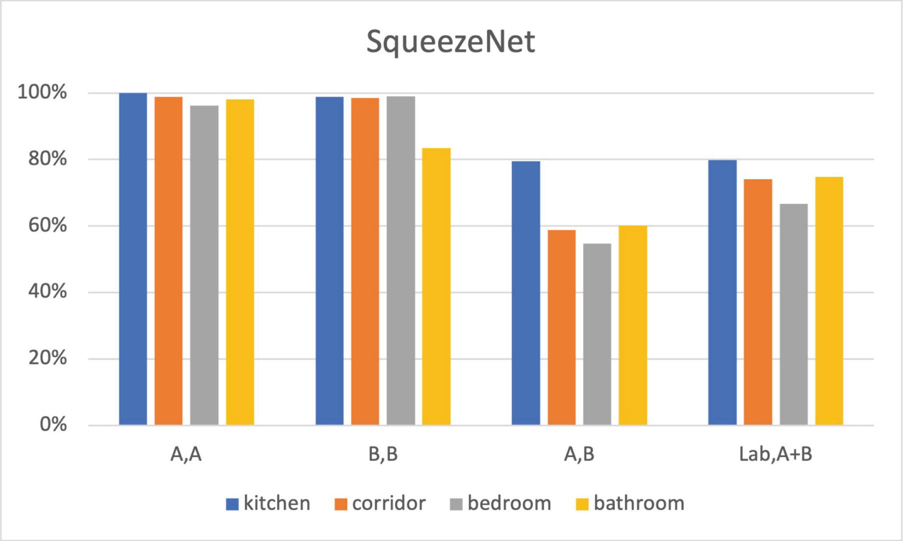


Fig. 4. Results of performance of SqueezeNet across different environments.

Result of the Second Method. The second method extracts features using wavelet scattering network after data acquisition and pre-processing and then classifies them using LSTM network. The experimental results of this method are shown in Fig. 5. From the figure, we can observe that the performance of the metrics for training and testing on the same dataset is better, almost always above 95%. However, when the environment changes, the method does not work very well, especially in the bedroom environment. The reason for this problem may be that the method is more demanding on the environment, and the extracted wavelet scattering features are still more environment-dependent. Another reason may be that there is a problem in the data acquisition process. However, there is still some improvement in the overall performance, especially in the same environment.

4.3 Comparison with Existing Fall Detection Efforts

Since the work in the FallDeFi paper has been compared with WiFall as well as RTFall [10], and outperformed these systems in terms of accuracy, here we only compare with FallDeFi. We use the data shown in Table 1 for comparison experiments, and then we average the accuracy experimental results of the four environments in four different training test sets to compare. The experimental results are shown in Fig. 6. Both methods improve the accuracy by 3%–10% over FallDeFi when trained and tested in the same environment. When the environment changes, SqueezeNet is more effective and slightly more accurate than FallDeFi.

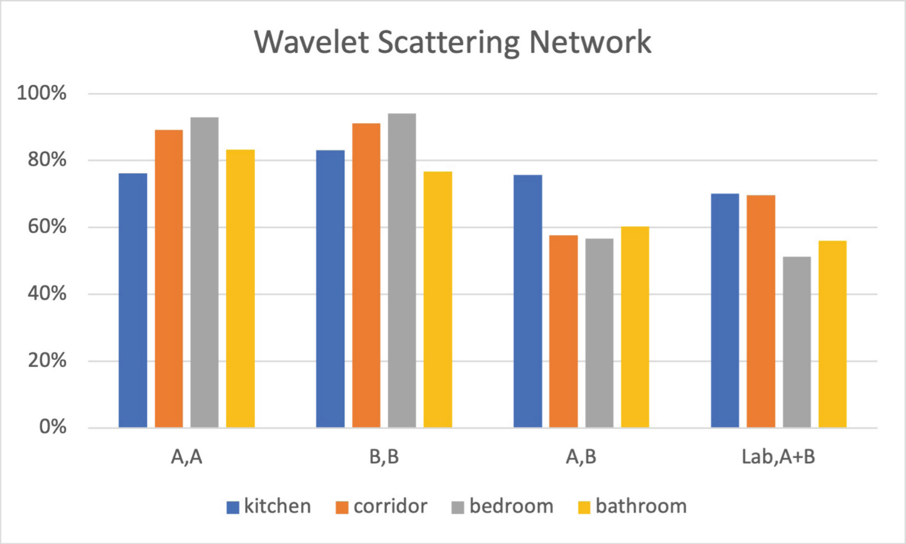


Fig. 5. Results of performance of wavelet scattering network across different environments.

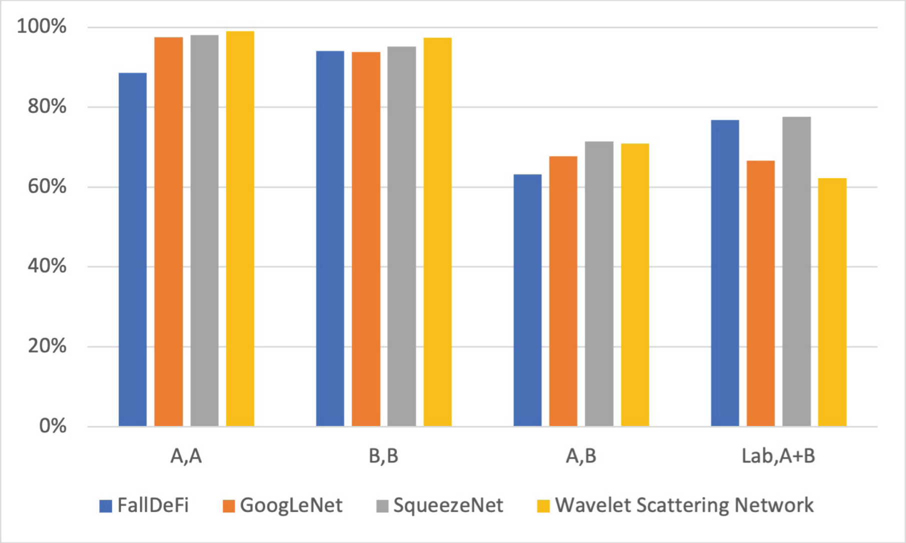


Fig. 6. Comparison with existing fall detection efforts.

5 Conclusion and Future Work

In this paper, we focus on the detection of fall behavior based on CSI of WiFi. We mainly adopt a method based on continuous wavelet transform for fall detection, which uses continuous wavelet transform to generate scale maps for feature extraction and then uses a pre-trained convolutional neural network for classification. However, convolutional neural networks still lack interpretability and are usually designed manually in a tedious trial-and-error process. Therefore, we also adopt the method of automatic feature extraction by wavelet scattering network and then classification by LSTM network. The average accuracy is above 95% in the pre-trained system and above 75% when the environment changed.

For fall detection the two methods proposed in this paper have improved in terms of detection accuracy, but there are still problems. The features selected in this paper remain environmentally relevant, thus the accuracy of fall detection needs to be improved when the environment changes. In future work, it is necessary to find features that are less dependent on the environment so that the accuracy of fall detection results will not decrease when the environment changes.

Acknowledgement. The work is supported by the National Key Research and Development Program of China under grant 2018YFB0204301, the National Natural Science Foundation (NSF) under grant 62072306, Open Fund of Science and Technology on Parallel and Distributed Processing Laboratory under grant 6142110200407.

References

1. Guo, D., Gu, S., Xie, J., Luo, L., Luo, X., Chen, Y.: A mobile-assisted edge computing framework for emerging IoT applications. ACM Trans. Sens. Netw. **17**(4), 1–24 (2021)
2. Yu, Z., Wang, Z.: Human Behavior Analysis: Sensing and Understanding. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-2109-6
3. Liu, J., Wang, Y., Chen, Y., Yang, J., Chen, X., Cheng, J.: Tracking vital signs during sleep leveraging off-the-shelf WiFi. In: Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing. ACM, June 2015
4. Li, J., Liu, H., Zhang, J.: Design and implementation of an RFID-based exercise information system. In: 2008 Second International Symposium on Intelligent Information Technology Application. IEEE, December 2008
5. Dingxing, Z., Ming, X., Yingwen, C., Shulin, W.: Probabilistic coverage configuration for wireless sensor networks. In: 2006 International Conference on Wireless Communications, Networking and Mobile Computing. IEEE, September 2006
6. Wang, W., Liu, A.X., Shahzad, M., Ling, K., Lu, S.: Understanding and modeling of WiFi signal based human activity recognition. In: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking. ACM, September 2015
7. Yang, Z., Zhou, Z., Liu, Y.: From RSSI to CSI. ACM Comput. Surv. **46**(2), 1–32 (2013)

8. Wang, Y., Wu, K., Ni, L.M.: WiFall: device-free fall detection by wireless networks. IEEE Trans. Mob. Comput. **16**(2), 581–594 (2017)
9. Wang, H., Zhang, D., Wang, Y., Ma, J., Wang, Y., Li, S.: RT-Fall: a real-time and contactless fall detection system with commodity WiFi devices. IEEE Trans. Mob. Comput. **16**(2), 511–526 (2017)
10. Palipana, S., Rojas, D., Agrawal, P., Pesch, D.: FallDeFi: ubiquitous fall detection using commodity Wi-Fi devices. Proc. ACM Interact. Mob. Wearable Ubiquit. Technol. **1**(4), 1–25 (2018)
11. Zhang, J., Tang, Z., Li, M., Fang, D., Nurmi, P., Wang, Z.: CrossSense: towards cross-site and large-scale WiFi sensing. In: Proceedings of the 24th Annual International Conference on Mobile Computing and Networking. ACM, October 2018
12. Lacasa, L., Luque, B., Ballesteros, F., Luque, J., Nuño, J.C.: From time series to complex networks: the visibility graph. Proc. Natl. Acad. Sci. **105**(13), 4972–4975 (2008)
13. Guo, X., Liu, J., Zhou, S., Zhu, E., Dong, S.: Image representation learning by transformation regression. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, January 2021
14. Szegedy, C., et al.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, June 2015
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, June 2016
16. Zeng, Y., Wu, D., Xiong, J., Zhang, D.: Boosting WiFi sensing performance via CSI ratio. IEEE Pervasive Comput. **20**(1), 62–70 (2020)

Data Collection of IoT Devices with Different Priorities Using a Fleet of UAVs

Qing Guo, Zhengzhong Xiang, Jian Peng(✉), and WenZheng Xu

College of Computer Science, Sichuan University, Chengdu, China
{guoqing1,xiangzz}@stu.scu.edu.cn, jianpeng@scu.edu.cn

Abstract. In this paper, we study sensory data collection of IoT devices in a wireless sensor network, employing a given number of UAVs. We observe that most existing studies ignored the different importance of data stored in IoT devices and simply minimized the longest data collection latency of IoT devices. Then, it is possible that the data collection latency of a IoT device may be long, the data collection priority of the IoT device is high and its data should be collected faster than IoT devices with low priorities. Considering the data collection priority of each IoT device, we formulate a novel weighted data collection latency minimization problem to collect data stored in IoT devices using the UAVs, by finding a closed tour for each UAV such that the maximum weighted data collection latency of IoT devices is minimized, where the data collection latency of IoT devices is composed of the hovering time of UAVs for data collection and the flying time of UAVs from one hovering location to another hovering location. To deal with the above NP-hard problem, we first propose a simplified data collection latency minimization problem which does not take account of the data collection priorities of IoT devices and base stations. Then, we devise an approximation algorithm for the simplified problem and further employ it to deal with the weighted data collection latency minimization problem. Finally, we evaluate the performance of the proposed algorithms through experimental simulations. Experimental results show that the proposed algorithms are very promising.

Keywords: IoT devices · Data collection · Flying tour scheduling · Approximation algorithm

1 Introduction

With the dramatically advanced technologies of UAVs over the past few decades, now UAVs are commonly known for features of low cost, swift deployment, high maneuverability and strong expandability. Due to the above inherent advantages, the utilization of UAVs has been widely expanded from military to plenty of applications. Especially, with the expansion of communication equipment, the mobility of UAVs offers a new opportunity for stable data transmission performance

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 218–230, 2022.
https://doi.org/10.1007/978-3-031-19214-2_18

enhancement for IoT devices, as short-range air-to-ground communication links with less effect of signal fading and shadowing can be established between UAVs and IoT devices [8,11,13]. UAVs with flexible location movement also state to fast and best suit the diverse and complex environments of IoT devices without more infrastructures, and such data collection systems are typically appealing for scenarios, e.g., undeveloped areas suffering serve shadowing, sudden or temporary events without the support of conventional terrestrial networks [12]. For example, in an area after disaster, terrestrial communication infrastructures (e.g., signal towers and cables) here may suffer great damage and can not relay data, then UAVs can be deployed to get close to IoT devices and act as mobile data collectors, which can play an important role in disaster area surveillance [5,7,10].

In recent years, there are plenty of studies focused on dispatching UAVs to perform data collection tasks in wireless sensor networks [7,9,14]. Kim *et al.* [7] employed multiple UAVs to collect the data of nodes in a two-dimensional space, such to minimize the data collection latency of nodes. But they ignored the serving time that UAVs need for collecting data, which also contributed to the data collection latency of nodes. Luo *et al.* [9] meticulously optimized the flying trajectory when UAVs are in the transmission range of sensors, such to minimize the flight times of UAVs. Zhan *et al.* [14] investigated a problem of finding flying tours for multiple UAVs to collect data from sensors so that the maximum time spent by the UAVs is minimized, by devising a genetic algorithm. Afshani *et al.* [1] proposed an approximation algorithm for the patrol-scheduling of multiple robots to visit sites at different frequencies, with the aim of minimizing the duration between consecutive visits of each site.

Unlike the existing studies that ignored the different importance of data stored in IoT devices and simply minimized the longest data collection latency of IoT devices, we notice that the data collection priorities of different IoT devices may vary significantly, especially when the UAVs need to repeatedly collect the data of IoT devices over an extended time period. For example, in a disaster rescue scenario where UAVs are deployed to repeatedly collect the fresh data of IoT devices, the data of a IoT device deployed to collect data of trapped people has a higher data collection priority than a IoT device that collects data for normal infrastructure, e.g., roads and bridges. Otherwise, people trapped here who have suffered accidents may wait for a long time, resulting in more casualties. Here, the data collection latency of a IoT device is defined as the consumed time of the tour which contains the IoT device. By existing algorithms, it is possible that the data collection latency of a high-priority IoT device found is just as long as the latency of a low-priority IoT device, and it thus will lead to a result that the data of the high-priority IoT device can not be updated in time.

As the above observation, in this paper, we schedule multiple UAVs to collect the data of IoT devices while considering the data collection priority of each IoT device, such to minimize the maximum weighted data collection latency of IoT devices. The data collection task poses many challenges, including: (i) how to assign IoT devices with data collection priorities to multiple UAVs, such that the data collection task is divided properly; (ii) how to schedule a flying tour for each UAV, so that the maximum weighted data collection latency of IoT devices is minimized; and (iii) how to ensure that the tour of each UAV contains a base station, such that the UAV can return the base station after completing one round of data collection missions.

To address the above challenges, we propose a weighted data collection latency minimization problem to schedule flying tours of multiple UAVs to collect data from IoT devices on the ground. The novelties of this paper lie in we take not only the data collection latency of IoT devices but also the data collection priority of each IoT device into consideration. We also propose an approximation algorithm for the weighted problem, such that the maximum weighted data collection latency of IoT devices is minimized.

The main contributions of this paper are summarized as follows.

- We first formulate a weighted data collection latency minimization problem, which is to find K rooted flying tours for a given number K of UAVs to collect the data of IoT devices with different data collection priorities, such that the maximum weighted data collection latency among the IoT devices is minimized.
- To deal with the weighted data collection latency minimization problem, we simplify the original problem to a simplified data collection latency minimization problem which does not consider the data collection priorities of IoT devices and base stations. For the simplified problem, we propose a $(2 \cdot OPT + 8\frac{r'}{\eta})(1+\epsilon)$-approximation algorithm, where OPT is the optimal value of the simplified problem, r' is the radius of disks (potential hovering areas) of IoT devices, η is the flying speed of UAVs, ϵ is a given constant in [2].
- By invoking the approximation algorithm for the simplified problem, we devise an approximation algorithm for the weighted data collection latency minimization problem, whose approximation ratio performs well in practice.
- We finally evaluate the performance of the proposed algorithms via simulation environments, and experimental results show that the proposed algorithms are very promising. Especially, the maximum weighted data collection latency of the IoT devices delivered by the proposed algorithm for the weighted data collection latency minimization problem is up to 32% shorter than those by existing algorithms.

2 Preliminaries

2.1 Network Model

We consider a data collection scenario where IoT devices are deployed in a to-be-monitored three dimensional area. Denote V as a set of the above IoT devices, i.e., $V = \{v_1, v_2, \ldots, v_n\}$ and $n = |V|$. Let (x_i, y_i, z_i) be the coordinates of each IoT device v_i with $1 \le i \le n$. Without loss of generality, we assume that all IoT devices are on a plane and the coordinate z_i of each IoT device v_i in V is zero [8,9], i.e., $z_i = 0$. Each IoT device v_i transmits its sensing data to a data collector (UAV) when the distance between the UAV and v_i is no more than a data transmission range r, e.g., $r = 100\ m$, when employing orthogonal frequency division multiple access (OFDMA) [8,11]. Denote l_i as the data collection priority of each IoT device v_i, where $0 < l_i \leq 1$. And a larger value of l_i means that the data of v_i should be collected faster.

To gather the data from IoT devices in V, K UAVs are employed as collectors to approach the IoT devices and obtain their data. $\mathbb{BS}$ is a set of K base stations of the UAVs, i.e., $\mathbb{BS} = \{BS_1, BS_2, \ldots, BS_K\}$. UAVs take off from base stations, and go back to base stations after completing one round of data collection missions to replenish energy. For crash avoidance and ease of operation, let z' be a fixed flying altitude of the UAVs, which is no more than the data transmission range r [8,9]. To make a feasible flying scheduling to collect the data of IoT devices in V, we need to not only assign each IoT device in V to the K UAVs, but also arrange the flying tour of each UAV. Suppose that IoT devices in a set $V_k = \{v_1, v_2, \ldots, v_{n_k}\}$ are assigned to UAV k, where $1 \leq k \leq K$, $n_k = |V_k|$, $V_1 \cup V_2 \cup \cdots \cup V_K = V$ and $V_1 \cap V_2 \cap \cdots \cap V_K = \emptyset$. Then, UAV k can collect the data of each IoT device v_i in V_k when it hovers at one point in the neighborhood of IoT device v_i. For example, in Fig. 1, a UAV can collect the data of IoT device v_1 when it hovers at point p_1 in the neighborhood of v_1. The neighborhood of IoT device v_i is represented as a disk s_i and $s_i = \{p|(x-x_i)^2+(y-y_i)^2 \leq r'^2\}$, where $r' = \sqrt{r^2 - z'^2}$ is the radius of disk s_i, p is a point contained in s_i with coordinates (x, y, z'), v_i' is the center of s_i with coordinates (x_i, y_i, z'). Without loss of generality, we assume that any two disks s_i and s_j do not overlap with each other [7–9]. Denote S as a set of disks over IoT devices in V, and $S = \{s_1, s_2, \ldots, s_n\}$. Denote V' as a set of centers of the disks in S, and $V' = \{v_1', v_2', \ldots, v_n'\}$. Figure 1 also shows an example of two flying tours for UAVs in a network.

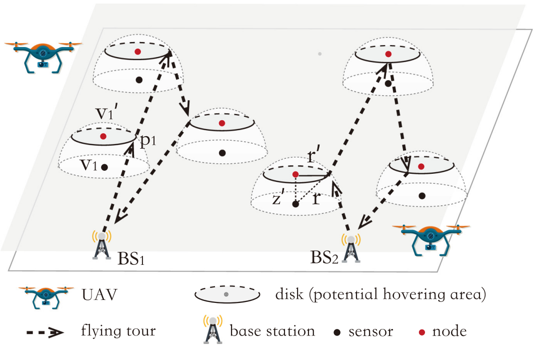


Fig. 1. An illustration of a network.

For each UAV k ($1 \leq k \leq K$), the consumed time of visiting each IoT device v_i in V_k consists of two parts, one is the hovering time $h(v_i)$ for collecting the data of v_i, i.e., $h(v_i) = \frac{D(v_i)}{B^t}$, where $D(v_i)$ and B^t are the volume of stored data and transmission bandwidth of v_i [8], respectively, another one is the flying time

$f(v_{i-1}, v_i)$ from the former IoT device v_{i-1} to v_i, i.e., $f(v_{i-1}, v_i) = \frac{d(p_{i-1}, p_i)}{\eta}$, where η is the flying speed of UAVs, $d(p_{i-1}, p_i)$ is the flying Euclidean distance of a UAV between IoT device v_{i-1} and v_i, p_{i-1} and p_i are points that a UAV hovers over v_{i-1} and v_i, respectively. It's worth noting that here we do not consider the limited flight time constraint of UAVs, as [5] provides a solution to extend an algorithm subject to the constraint.

For each UAV k, denote C_k as the tour to collect all data from IoT devices in V_k assigned to UAV k. The total consumed time $w(C_k)$ of tour C_k is calculated as the sum of hovering time $h(v_i)$ for each IoT device v_i in V_k and the flying time between the IoT devices, i.e.,

$$w(C_k) = \sum_{v_i \in V_k} h(v_i) + \sum_{v_i, v_{i+1} \in V_k} f(v_i, v_{i+1}) + f(BS_k, v_1) + f(v_{n_k}, BS_k), \quad (1)$$

where BS_k is the base station assgined to UAV k.

For each IoT device v_i in V_k that assigned to UAV k, the weighted data collection latency of v_i is $l_i \cdot w(C_k)$, where l_i is the data collection priority of v_i, $w(C_k)$ is the consumed time of tour C_k which contains v_i.

2.2 Problem Definition

Given a set V of IoT devices with coordinates where each IoT device v_i has a data collection priority l_i, a disk set S of potential hovering areas over IoT devices in V, and the hovering time $h(v_i)$ of UAVs over each IoT device v_i, we propose a novel *weighted data collection latency minimization problem*, which is to find K rooted tours for K UAVs to collect the data of IoT devices in V, where 'rooted' means that each tour contains a base station as its root, such that the maximum weighted data collection latency of IoT devices in V, i.e., $\max_{k=1}^{K}\{l_k \cdot w(C_k)\}$, is minimized, where l_k is the maximum data collection priority of IoT devices in V_k that assigned to UAV k, i.e., $l_k = \max_{v_i \in V_k} l_i$, and $w(C_k)$ represents the consumed time of tour C_k.

3 Algorithm for the Simplified Problem

It is easily proven that the weighted data collection latency minimization problem is NP-hard, as its special case when the data transmission range r of IoT devices tends to the flying altitude z' of UAVs, the data collection priority of each IoT device is equal and $K = 1$, i.e., the TSP problem, is NP-hard. To solve the NP-hard weighted problem, in this section, we propose a simplified data collection latency minimization problem which ignores the data collection priorities of IoT devices and base stations. And we further propose an algorithm for the simplified problem which will be employed to deal with the weighted problem in the next section.

Here, we give a detailed definition of the simplified data collection latency minimization problem. Given a set $V = \{v_1, v_2, \ldots, v_n\}$ of IoT devices, a set

S of disks over each IoT device in V, and the hovering time $h(v_i)$ of each IoT device v_i in V, the simplified data collection latency minimization problem is to find K tours that visit all the disks over IoT devices in V, such that the longest data collection latency of each IoT device in V, i.e., $\max_{k=1}^{K} w(C_k^s)$, is minimized, where C_k^s is the tour of UAV k, and $w(C_k^s)$ is the consumed time of tour C_k^s.

We introduce the framework of the algorithm for the simplified problem. Suppose that C_1^*, C_2^*, ..., C_K^* form an optimal solution to the problem, and OPT is the optimal value for the problem, i.e., $OPT = \max_{k=1}^{K} w(C_k^*)$, where $w(C_k^*)$ is the consumed time of tour C_k^*, K is the number of UAVs. With a guess value B ($B \geq OPT$) of OPT, the algorithm first assigns each disk in the set S into some subsets where each subset is disjoint with others. For each subset, the algorithm then constructs an approximate tour to visit each disk s_i contained in the subset, and shortcut the approximate tour into several sub-tours with the consumed time of each sub-tour is no more than $(2B + 8\frac{r'}{\eta})(1 + \epsilon)$, where r' is the radius of each disk, η is the flying speed of UAVs, and ϵ is a given constant in [2]. Obviously, the optimal value OPT can be found through invoking a binary search and an approximate solution thus can be obtained [5]. The range of the binary search is set from 0 to $w(C)$, where $w(C)$ is the consumed time of a tour C which contains all center nodes of disks in S, and the tour C can be obtained by invoking `Christofides' algorithm`.

In the following, we will go over the algorithm. Firstly, we break the disk set S up into several disjoint sub-sets. With the given node set V' which are centers of disks in S, we construct an auxiliary complete graph $G^a = (V', E^a)$, where E^a is a set of edges between any two nodes in V'. Here, $f(s_i, s_j)$ represents the minimum flying time between disk s_i and s_j, i.e., $f(s_i, s_j) = (d(v_i', v_j') - 2r')/\eta$, where v_i' and v_j' are the centers of disks s_i and s_j, respectively; $d(v_i', v_j')$ is the Euclidean distance between node v_i' and v_j', r' is the radius of disks in S, and η is the flying speed of UAVs. Then, with the given guess B ($B \geq OPT$) of OPT, we divide the complete graph G^a into several connected components CC_1, CC_2, ..., CC_q through deleting each edge (v_i', v_j'), once the flying time $f(s_i, s_j)$ is strictly larger than $\frac{B}{2}$. According to CC_1, CC_2, ..., CC_q, we divide the disk set S into q disjoint sub-sets S_1, S_2, ..., S_q, when the center v_i' of each disk s_i in S_t $(1 \leq t \leq q)$ is contained in the connected component CC_t, where $S_1 \cup S_2 \cup \cdots \cup S_q = S$ and $S_1 \cap S_2 \cap \cdots \cap S_q = \emptyset$.

For each disk sub-set S_t $(1 \leq t \leq q)$, suppose that $S_t = \{s_1, s_2, \ldots, s_{n_t}\}$, where $n_t = |S_t|$. A tour C_t' which visits disks $s_1, s_2, \ldots, s_{n_t}$ can be found by invoking the $(1 + \epsilon)$-algorithm in [2] for the traveling salesman problem with disjoint neighborhood. Assume that $C_t' = p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_{n_t}$, where p_i $(1 \leq i \leq n_t)$ is a point located in disk s_i. It's worth noting that although the $(1+\epsilon)$-algorithm in [2] do not consider the hovering time, it is still applicable in this paper according to [4,5]. And the consumed time $w(C_t')$ of tour C_t' is no more than $(1 + \epsilon)w(C_t^*)$, i.e., $w(C_t') \leq (1 + \epsilon)w(C_t^*)$, where C_t^* is a tour that visits all disks $s_1, s_2, \ldots, s_{n_t}$ with a minimum consumed time.

After obtaining q tours C_1', C_2', ..., C_q', we need to check the consumed time $w(C_t')$ of each tour C_t' $(1 \leq t \leq q)$ to see if it is greater than $(2B + 8\frac{r'}{\eta})(1 + \epsilon)$. If $w(C_t') \leq (2B + 8\frac{r'}{\eta})(1 + \epsilon)$, then we can obtain a tour which is as the same as the tour C'_t. Otherwise, we construct sub-tours whose consumed times are no more than $(2B + 8\frac{r'}{\eta})(1+\epsilon)$ by invoking a similar splitting procedure in [5]. When $B \geq OPT$, it can be proved that the number of obtained sub-tours is no more than K.

The algorithm for the simplified data collection latency minimization problem is assigned as `Algorithm` 1.

Algorithm 1: Algorithm for the simplified data collection latency minimization problem

```
Data: Given a set $V$ of IoT devices, a set $S$ of disks over IoT devices in $V$, the
      hovering time $h(v_i)$ of each $v_i$ in $V$.
Result: $K$ tours $C^s_1, C^s_2, \ldots, C^s_K$.
begin
    Let $B^s_l = 1$, $B^s_u = w(C)$
    while $B^s_l + 1 < B^s_u$ do
        Let $B = \lfloor \frac{B^s_l + B^s_u}{2} \rfloor$ /* $B$ is a guess of $OPT$. */
        Obtain $q$ disk sets $S_1$, $S_2$, ..., $S_q$ with $B$
        for $t \leftarrow 1$ to $q$ do
            For the $t_{th}$ disk set $S_t$, obtain a tour $C'_t$ by invoking [2] and
              construct sub-tours whose consumed times are no more than
              $(2B + 8\frac{r'}{\eta})(1+\epsilon)$
        if the number of obtained subtours is no more than $K$ then
            $B^s_l \leftarrow B$; Skip to the next while loop
        else
            $B^s_u \leftarrow B$; Skip to the next while loop
    Construct $K$ tours $C^s_1, C^s_2, \ldots, C^s_K$ from the obtained sub-tours
```

Lemma 1. *Given a set $V = \{v_1, v_2, \ldots, v_n\}$ of IoT devices, a set S of disks over IoT devices in V, and the hovering time $h(v_i)$ of each IoT device v_i in V, there is a $(2OPT + 8\frac{r'}{\eta})(1+\epsilon)$-approximation algorithm,* `Algorithm` *1, for the simplified data collection latency minimization problem, where OPT is the optimal value of the simplified problem, r' is the radius of disks over IoT devices, η is the flying speed of UAVs, and ϵ is a given constant in [2]. The time complexity of* `Algorithm` *1 is $O(n^{O(m')})$, where m' and ϵ are constants in [2], and n is the number of IoT devices in V.*

Proof. The proof is omitted, due to space limitation.

4 Algorithm for the Weighted Data Collection Latency Minimization Problem

In this section, we will deal with the weighted data collection latency minimization problem which considers the data collection priorities of IoT devices and base stations. We will propose an algorithm for the weighted problem while invoking the algorithm for the simplified problem in the previous section. We first show the basic idea of the algorithm for the weighted problem, then introduce the detail of the algorithm.

The basic idea of the algorithm is to deal with the weighted problem in two steps. The first step is to decide the set of IoT devices assigned to each UAV k $(1 \leq k \leq K)$ while invoking the algorithm for the simplified problem, and the second step is scheduling the flying tour of UAV k to collect the data of IoT devices assigned to UAV k. We describe the basic idea of the first step as follows. Suppose that OPT' is the optimal value of the weighted data collection latency minimization problem. We guess a value B' of OPT' with $B' \geq OPT'$, and find a tour cover for each UAV k $(1 \leq k \leq K)$, such that the weighted data collection latency of each IoT device is no more than $\alpha mKB'$, where α is the ratio of `Algorithm` 1 for the simplified problem, K is the number of UAVs, $m = \log_2\lceil \frac{l_{max}}{l_{min}} \rceil$, l_{max} and l_{min} are the maximum and minimum data collection priorities of IoT devices in V, respectively. Through a binary search for the optimal value OPT', we can get a tour cover $\mathcal{C}_k$ for each UAV k.

Then, we introduce the basic idea of the second step. From the tour cover $\mathcal{C}_k$ which assigned to UAV k, we can get a tour C_k by invoking the algorithm in [2]. By invoking the algorithm in [5], we find a proper base station from the K base stations for tour C_k.

We elaborate the detail of the algorithm for the weighted data collection latency minimization problem as follows. Recall that there is a set S of disks over IoT device v_i. According to the data collection priorities of IoT devices in V, we divide disks in S into different groups as follows. Denote by l_{max} and l_{min} as the maximum and minimum data collection priorities of IoT devices in V, respectively, where $l_{max} \geq l_{min} > 0$. And m is logarithm of the ratio of the maximum priority l_{max} and the minimum priority l_{min}, i.e., $m = \log_2\lceil \frac{l_{max}}{l_{min}} \rceil$. Without loss of generality, we assume that the maximum priority among IoT devices in V is 1, i.e., $l_{max} = 1$. For each priority l_i of IoT device v_i in V, we assign $l'_i = \sup\{2^x \cdot l_{max} | x \in \mathbb{Z} \; and \; 2^x \cdot l_{max} \geq l_i\}$ as the virtual priority of v_i. From the definition of the virtual priority l'_i, we have $l_i \leq l'_i < 2l_i$. We assume that the number of different virtual priorities of IoT devices in V is n_s, which is no more than m, i.e., $n_s \leq m$. Due to space limitation, the proof is omitted. For IoT devices whose virtual priorities are same, we assign them to a group. According to the groups of IoT devices, we also assign the disks in S into n_s groups, i.e., S'_1, S'_2, ..., S'_{n_s}.

For each group of disk S'_j $(1 \leq j \leq n_s)$, `Algorithm` 1 is employed to obtain a t-min-max tour cover $\{C_{j,1}, C_{j,2}, \ldots, C_{j,t}\}$ with the consumed time of each tour contained in the tour cover is no more than $\alpha B'$, where t is a smallest integer between 1 and K, B' is the guess value of OPT' $(B' \geq OPT')$, and α is the ratio of `Algorithm` 1. Assume that there are q_j obtained tours $\{C_{j,1}, C_{j,2}, \ldots, C_{j,q_j}\}$ on the disk set S'_j. After the above operation, we get n_s min-max tour covers. Then, we assign the disks visited by each tour contained in the n_s min-max tour covers to the K UAVs as follows.

For each tour $C_{j,i}$ $(1 \leq j \leq n_s, 1 \leq i \leq q_j)$, denote N as the set of disks contained in tour $C_{j,i}$, i.e., $N = S(C_{j,i})$. We assign each disk s in N to a non-free UAV k $(1 \leq k \leq K)$ if the flying time between s and the first tour $C^{j'}$ assigned to the UAV k is no more than $2^{j'-1}B'$, i.e., $f(s, C^{j'}) \leq 2^{j'-1}B'$, and $f(s, C^{j'}) = \min_{\forall u \in C^{j'}}\{f(s,u)\}$, j' is the virtual priority of IoT devices visited by tour $C^{j'}$, u is a disk contained in tour $C^{j'}$ and $u \in S$. If there are still some disks in N that are not assigned to any UAV, then assign them to a free UAV.

Through a binary search for OPT' with the guess B' in range $[0, l_{max} \cdot w(C)]$, we can find a tour cover $\mathcal{C}_k$ for each UAV k, where l_{max} is the maximum data collection priority of IoT devices in V, $w(C)$ is the consumed time of a tour C which contains all center nodes of disks in S by invoking `Christofides' algorithm`.

For each UAV k $(1 \leq k \leq K)$ with the tour cover $\mathbb{C}_k$, a tour C_k that contains a base station in $\mathbb{BS}$ can be obtained by invoking the algorithm in [2,5].

The algorithm for the weighted data collection latency minimization problem is presented in `Algorithm` 2.

Lemma 2. *Given a set V of IoT devices, a set S of disks over each IoT device in V, the hovering time $h(v_i)$ of each IoT device v_i in V and K base stations, there is a $O(mK)$-approximation algorithm,* `Algorithm` 2*, for the weighted data collection latency minimization problem, where $m = \log_2\lceil \frac{l_{max}}{l_{min}} \rceil$, l_{max} and l_{min} are the maximum and minimum priority of IoT devices in V, respectively, K is the number of UAVs. The time complexity of* `Algorithm` 2 *is $O(n^{O(m')})$, where m' is a constant [2], and n is the number of IoT devices in V.*

Proof. The proof is omitted, due to space limitation.

5 Performance Evaluation

5.1 Simulation Environment

In this section, we evaluate the performance of the proposed algorithms through extensive experiments. In a 10 km × 10 km × 100 m three-dimensional Euclidean space, we consider a network which consists of 25 to 150 IoT devices. The data volume of each IoT device is randomly drawn in the range from 100 MB to 1,000 MB, and the data transmission bandwidth of each IoT device is 150 MB/s [8,9]. The radius of the disk over each IoT device is 50 m [3,8]. The number of UAVs varies from 2 to 10. The UAV has a constant flying speed $\eta = 10$ m/s [5]. Base stations are randomly located at the border of the space. For each parameter setting, we obtain average results from 100 instances.

To evaluate the performance of the proposed algorithm `approAlg` for the weighted data collection latency problem, we here introduce three benchmarks. Algorithm `approAlgMultiRoots` finds K flying tours, such that the maximum consumed time among the flying tours is minimized, while ignoring data collection priorities of IoT devices [3]. Algorithm `multiNei` [6] constructs flying tours with the longest tour time $369 \cdot OPT_m + c \cdot r$ for the multi-rooted min-max cycle

Algorithm 2: Algorithm for the weighted data collection latency minimization problem (approAlg)

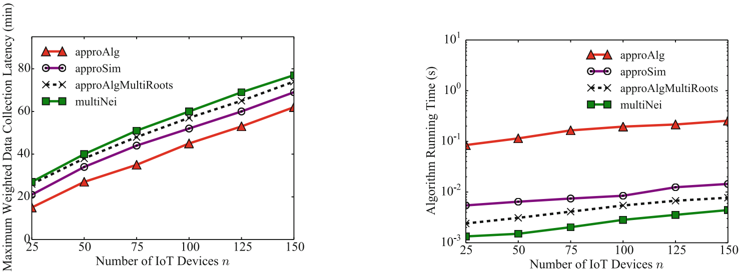
```
Data: Given a set $V$ of IoT devices, a set $S$ of disks over IoT devices in $V$, the
      hovering time $h(v_i)$ of each $v_i$ in $V$, and a set $\mathbb{BS}$ of base stations.
Result: $K$ tours $C_1, C_2, \ldots, C_K$.
begin
    Let $B_l = 1$, $B_u = l_{max} \cdot w(C)$
    while $B_l + 1 < B_u$ do
        Let $B' = \lfloor \frac{B_l+B_u}{2} \rfloor$ /* $B'$ is a guess of $OPT'$. */
        Obtain $n_s$ disk sets $S'_1$, $S'_2$, ..., $S'_{n_s}$
        for $j \leftarrow 1$ to $n_s$ do
            For the $j_{th}$ disk set $S'_j$, obtain a $t$-min-max tour cover {$C_{j,1}$, $C_{j,2}$,
              ..., $C_{j,q_j}$} on $S'_j$ by invoking Algorithm 1 where $t$ is a smallest
              integer in range $[1, K]$, such that the consumed time of each tour is
              no more than $\alpha \cdot 2^{j-1} B'$
            if the tour cover on $S'_j$ is not successfully obtained then
                $B_l \leftarrow B'$; Skip to the next while loop
        for $j \leftarrow 1$ to $n_s$ do
            for $i \leftarrow 1$ to $q_j$ do
                $N \leftarrow S(C_{j,i})$ /*Assign all disks contained in $C_{j,i}$ to $N$*/
                for every non-free UAV $k$ $(1 \le k \le K)$ do
                    $N \leftarrow S(C_{j,i})$; $C^{j'} \leftarrow$ the first tour assigned to the UAV $k$;
                      $N' \leftarrow \{s | s \in N, f(s, C^{j'}) \le 2^{j'-1} B'\}$; Construct a tour
                      $C(N')$ by invoking [4] and assign it to UAV $k$; $N \leftarrow N \backslash N'$
                if $N \neq \emptyset$ then
                    if there is no free UAV then
                        $B_l \leftarrow B'$; Skip to the next while loop
                    else
                        Assign tour $C(N)$ to a free UAV $k'$ as the first tour
        Obtain the set $\mathcal{C}_k$ of tours assigned to UAV $k$
        if all disks in $S$ are assigned to some UAV then
            $B_u \leftarrow B'$; Skip to the next while loop
    for $k \leftarrow 1$ to $K$ do
        Obtain a new tour $C_k$ which contains a base station for the UAV $k$ from
          the set $\mathcal{C}_k$ of tours assigned to UAV $k$ by invoking [4] and [7]
```

cover problem with neighborhoods, where OPT_m is the optimal value of the problem. Algorithm approSim employs Algorithm 1 for the simplified problem mentioned in this paper while invoking the procedure in [5], such to find a base station for each UAV.

5.2 Algorithm Performance

In the following, we study the impact of the network size, the number of UAVs and the speed of UAVs.

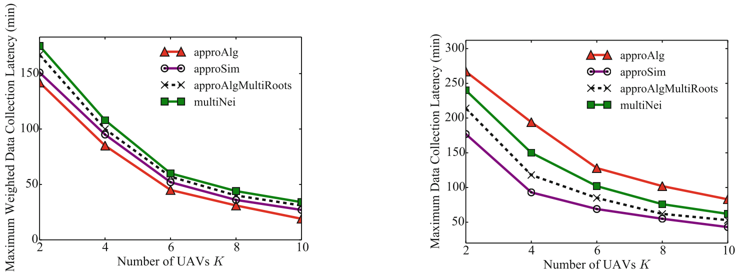
We first evaluate the performance of the proposed algorithm `approAlg` against existing algorithms `approSim` and `approAlgMultiRoots` and `multiNei` by varying the number n of IoT devices from 25 to 150, while the number K of UAVs is 6. Figure 2(a) shows that the maximum weighted data collection latencies by the algorithms increase as the number of IoT devices grows, as UAVs fly longer distances when there are more IoT devices. And it also shows that the maximum weighted data collection latency of IoT devices delivered by the proposed algorithm `approAlg` is about 25% to 32% less than those by existing algorithms. For example, when there are $n = 100$ IoT devices, the maximum weighted data collection latencies by algorithms `approAlg`, `approAlgMultiRoots`, `approSim` and `multiNei` are 46, 52, 58, 61 min. The reason behind this is that the existing algorithms do not take into account the data collection priorities of IoT devices. Finally, Fig. 2(b) shows the running



(a) Maximum weighted data collection latency

(b) Algorithm running time

Fig. 2. Performance of algorithms `multiNei`, `approSim`, `approAlgMultiRoots` and `approAlg` by varying the number n of IoT devices from 25 to 150, while the number K of UAVs is 6.



(a) Maximum weighted data collection latency

(b) Maximum data collection latency

Fig. 3. Performance of different algorithms by increasing the number K of UAVs from 2 to 10, when there are 100 IoT devices.

times of the mentioned algorithms, from which it can be seen that the running time of the proposed algorithm `approAlg` is acceptable in practice, e.g., no more than 0.5 s when there are 150 IoT devices.

We then investigate the algorithm performance by varying the number K of UAVs from 2 to 10 when the number of IoT devices is fixed to 100. Figure 3(a) shows that the maximum weighted data collection latencies by the algorithms decrease as the number K of UAVs increases. This is because that the number of IoT devices assigned to each UAV decreases as the number of UAVs increases. From Fig. 3, we also observe that the maximum weighted data collection latency by algorithm `approAlg` is much shorter than those by algorithms `approAlgMultiRoots`, `approSim` and `multiNei`, while the maximum data collection latency by algorithm `approAlg` is longer than those by the benchmarks.

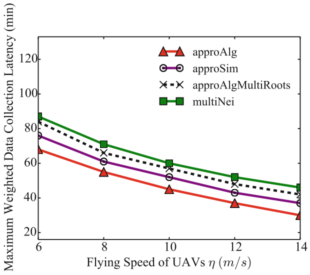


Fig. 4. Performance of different algorithms by increasing the speed η of each UAV from 6 m/s to 14 m/s, when there are 100 IoT devices and 6 UAVs.

We finally study the algorithm performance by increasing the speed η of each UAV from 6 m/s to 14 m/s, when there are 100 IoT devices and 6 UAVs. Figure 4 demonstrates that the maximum weighted data collection latencies by the four algorithms decrease with a larger flying speed of each UAV, and the maximum weighted data collection latency of algorithm `approAlg` is about from 15% to 32% smaller than those by other algorithms.

6 Conclusion

In this paper, we studied a problem of finding flying tours for multiple UAVs such that the maximum weighted data collection latency of IoT devices is minimized, where each IoT device owns a data collection priority. In order to solve the NP-hard problem, we first proposed a simplified data collection minimization problem which ignores the priorities of IoT devices and base stations, then devise an approximation algorithm for the simplified problem. Further, we dealt with the original problem through invoking the approximation algorithm for the simplified problem and finally evaluated the proposed algorithms through experimental simulation. The simulation results demonstrated that the proposed algorithms are very promising.

Acknowledgment. This work was supported by the National Key Research and Development Program of China (2020YFB0704502), the Key Research and Development Program of Sichuan Province, China (22ZDYF3599), the Cooperative Program of Sichuan University and Yibin (2020CDYB-30).

References

1. Afshani, P., et al.: Approximation algorithms for multi-robot patrol-scheduling with min-max latency. In: LaValle, S.M., Lin, M., Ojala, T., Shell, D., Yu, J. (eds.) WAFR 2020. SPAR, vol. 17, pp. 107–123. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-66723-8_7
2. Dumitrescu, A., Mitchell, J.S.: Approximation algorithms for TSP with neighborhoods in the plane. J. Algorithms **48**(1), 135–159 (2003)
3. Deng, L., et al.: Approximation algorithms for the min-max cycle cover problem with neighborhoods. IEEE/ACM Trans. Netw. **28**(4), 1845–1858 (2020)
4. Zhang, J., Li, Z., Xu, W., Peng, J., Liang, W., Xu, Z., et al.: Minimizing the number of deployed UAVs for delay-bounded data collection of IoT devices. In: IEEE Conference on Computer Communications, pp. 1–10. IEEE(2021)
5. Guo, Q., Peng, J., Xu, W., Liang, W., Jia, X., Xu, Z., et al.: Minimizing the longest tour time among a fleet of UAVs for disaster area surveillance. IEEE Trans. Mob. Comput. (2020)
6. Kim, D., Uma, R.N., Abay, B.H., Wu, W., Wang, W., Tokuta, A.O.: Minimum latency multiple data mule trajectory planning in wireless sensor networks. IEEE Trans. Mob. Comput. **13**(4), 838–851 (2013)
7. Kim, D., Xue, L., Li, D., Zhu, Y., Wang, W., Tokuta, A.O.: On theoretical trajectory planning of multiple drones to minimize latency in search-and-reconnaissance operations. IEEE Trans. Mob. Comput. **16**(11), 3156–3166 (2017)
8. Li, Y., Liang, W., Xu, W., Jia, X.: Data collection of IoT devices using an energy-constrained UAV. In: 2020 IEEE International Parallel and Distributed Processing Symposium, pp. 644–653. IEEE (2020)
9. Luo, C., Satpute, M.N., Li, D., Wang, Y., Chen, W., Wu, W.: Fine-grained trajectory optimization of multiple UAVs for efficient data gathering from WSNs. IEEE/ACM Trans. Netw. **29**(1), 162–175 (2020)
10. Liang, Y., et al.: Nonredundant information collection in rescue applications via an energy-constrained UAV. IEEE Internet Things J. **6**(2), 2945–2958 (2018)
11. Mozaffari, M., Saad, W., Bennis, M., Debbah, M.: Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications. IEEE Trans. Wireless Commun. **16**(11), 7574–7589 (2017)
12. Wang, X., Chowdhery, A., Chiang, M.: Networked drone cameras for sports streaming. In: 37th International Conference on Distributed Computing Systems, pp. 308–318. IEEE (2017)
13. Wu, Q., Liu, L., Zhang, R.: Fundamental trade-offs in communication and trajectory design for UAV-enabled wireless network. IEEE Wirel. Commun. **26**(1), 36–44 (2019)
14. Zhan, C., Zeng, Y.: Completion time minimization for multi-UAV-enabled data collection. IEEE Trans. Wireless Commun. **18**(10), 4859–4872 (2019)

Cross-Model Operator Batching for Neural Network Architecture Search

Lingling Ye[1], Chi Zhang[1], Mingxia Li[1], Zhenhua Han[2], and Haisheng Tan[1](✉)

[1] School of Computer Science and Technology, University of Science and Technology of China, Hefei, China
{yeling0,gzhnciha,sa20011036}@mail.ustc.edu.cn, hstan@ustc.edu.cn
[2] Microsoft Research Asia (MSRA), Shanghai, China

Abstract. Recently, automated machine learning (AutoML) and neural architecture search (NAS), regarded as promising techniques to design deep learning (DL) models automatically, have received increasing attention from both industry and academia. NAS will generate a large number of candidate models, which typically consist of numerous common substructures, providing a vast opportunity for cross-model optimization (e.g., operator batching) to improve training efficiency. However, most of the existing AutoML frameworks do not make use of operator batching and we also lack an efficient batching strategy. In this work, we propose a heuristic scheme named `DPBat` to guide the operator batching among multiple models in NAS. For most models, the operator batching of `DPBat` can be finished in just a few seconds, which is negligible compared to the subsequent training. We adopt Microsoft's open source AutoML framework NNI to implement `DPBat` to real NAS scenarios. Extensive experiments show that `DPBat` is highly effective in improving training efficiency and reducing the overhead of operator batching, with a throughput 3.7× higher than the standard practice of running each job without batching.

Keywords: AutoML · NAS · Operator batching

1 Introduction

In recent years, deep learning has achieved great success in various domains, including image classification [7,8], natural language translation [13], and object detection [9]. However, this success has been accompanied by a growing demand for architectural engineering. Most of the complex neural architectures are manually designed (e.g., VGG-16 [10], BERT [4] and GPT-3 [1]), which is time-consuming and requires lots of expertise experience. Therefore, automated machine learning (AutoML) and neural architectures search (NAS) have received more and more attention from both industry and academia. Some institutions have launched their framework that implements the search for neural network architectures, such as Microsoft's NNI, Huawei's Vega, and Amazon's AutoGluon.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 231–242, 2022.
https://doi.org/10.1007/978-3-031-19214-2_19

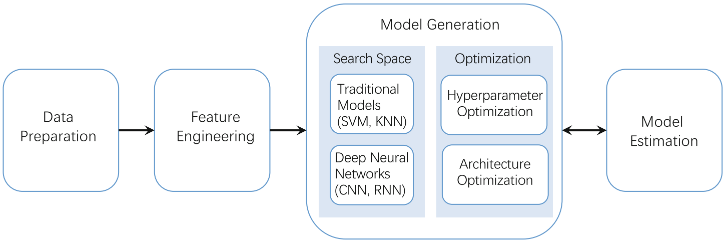

Data Preparation
Feature Engineering
Model Generation
Search Space
Traditional Models (SVM, KNN)
Deep Neural Networks (CNN, RNN)
Optimization
Hyperparameter Optimization
Architecture Optimization
Model Estimation

Fig. 1. An overview of AutoML pipeline

A typical AutoML pipeline contains four parts as Fig. 1 shows: data preparation, feature engineering, model generation, and model evaluation. As a key component of AutoML, the search space defines which neural architectures the NAS method can discover in principle. The number of models covered by the search space is enormous, and searching for an optimal model could take up to hundreds of hours [14]. We first investigate how NAS generates models to reduce the training cost and optimize hardware resource usage. The optimization in model generation can be divided into hyperparameter optimization (HPO) and architecture optimization (AO). Models for hyperparameter tuning often have the same types of operators with the same shape. Analogously, models in architecture optimization scenarios tend to have significant similarities as they share a common skeleton [14]. Operators with the same type and parameters can potentially be *batched* together and computed in a single operator kernel, which enables more fine-grained GPU sharing by using less GPU memory to increase SIMD utilization. Therefore, there are huge opportunities for AutoML frameworks to optimize the training of multiple similar models and improve hardware utilization.

Figure 2 illustrates an example that two models share multiple common operators, where Model 1 and Model 2 both have conv 3×3 and ReLU. After batching, the input of the common operators (conv 3×3, ReLu) are fused along the batch dimension, and the outputs are split when operators (BatchNorm2d) vary.

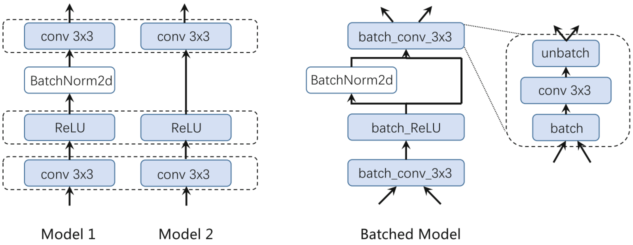


Fig. 2. Operator batching: an example

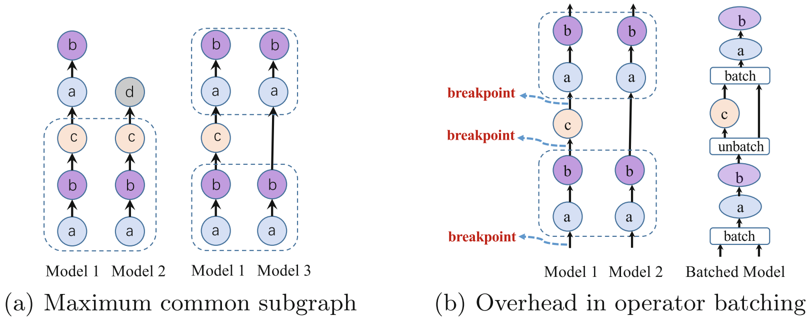
In the literature, there are apparent gaps between the requirement to support this kind of cross-model optimization and the existing operator batching method. [12] came up with the idea of inter-model horizontal fusion, which only deals with HPO scenarios. However, the submitted training tasks generated in NAS are disorder, which [12] would fail to deal with. Besides, the implementation of operators batching in [14] is limited. First, the types of operator batching it supports are limited (Conv2d only). Second, the operator batching strategy in [14] is rudimentary. Specifically, this algorithm uses the idea of breadth-first search (BFS) to compare the operators of each layer between the models until they are different, which means that when the first few layers of the models are different, the strategy's performance will degrade.

To narrow the gaps mentioned above, we propose a scheme to improve the batching efficiency in NAS scenarios. Our objective is to make full use of the similarities among the models and improve training throughput, which is a key performance indicator of training efficiency. Our contributions can be summarized as follows:

- We formulate the DL job clustering and batching problem in NAS scenarios described in Sect. 3. The objective is to maximize the throughput of model training per unit time and help accelerate the process of model generation.
- We propose a novel **D**ynamical **P**rogramming based **Bat**ching strategy, named `DPBat`. `DPBat` includes an efficient cluster algorithm that takes advantage of the similarity among the models generated in NAS. Based on the clustering result, `DPBat` determines an operator batching strategy by comprehensively investigating the performance improvement and overhead.
- We conduct extensive experiments by using Microsoft's open source AutoML framework NNI to evaluate the performance of our algorithm in real NAS scenarios. The experimental results indicate that `DPBat` can significantly improve training efficiency and reduce the overhead of operator batching, achieving up to 3.7× higher throughput than the standard practice of running each job on a separate accelerator.

2 Motivation

Lack of Indicators to Measure Which Models Should be Batched Together. The DNN models generated from the same search space tend to have similarities, and those with the highest similarity should be put together for operator batching. This is not taken into account by the existing batching algorithm due to the lack of indicators that can accurately describe the similarity of models. For example, the maximum common subgraph is not a good indicator. Although a DNN model architecture can be depicted as a data flow graph (DFG), the model similarity is not equivalent to the size of the largest common subgraph. As Fig. 3(a) shows, each model is abstracted into a DFG, where each node represents an operator or a subgraph. Obviously, model 1 and model 2 have the largest common subgraph. But model 1 and model 3 can batch



(a) Maximum common subgraph (b) Overhead in operator batching

Fig. 3. Similarity and overhead in operator batching

more nodes with smaller common subgraphs. So a wiser solution should be to batch model 1 and 3 together instead of model 1 and 2, even though they share a larger common subgraph.

Limitation of the Current Batching Schemes. The idea of inter-model horizontal fusion in [12] only applies to the situation where the models are all the same except for hyperparameters. It can not handle the situation when the architectures of submitted training models get different. Besides, there are limitations in the implementation of operators batching in [14]. The types of operator batching it supports are limited, and there is no efficient algorithm to achieve operator batching between multiple models.

Lack of Consideration for Batching Cost. [12,14] take no consideration of batching cost. Different from models in [12] which have the same architecture, each layer of operators in the model can be batched without breakpoints (the position that generates the batch/unbatch cost). In more general scenarios, operators that can be batched are not continuous. As depicted in Fig. 3(b), the input of common operators needs to be concated along the channel dimension while the output is split at the breakpoint. The operations of concating and splitting bring extra overhead in time and memory. At the same time, batching of different operators brings different performance improvements. The factors mentioned above will affect the choice of operators to be batched.

3 Problem Formulation

3.1 System Model

We consider a system with $\mathcal{D} = \{d_1, ..., d_{|\mathcal{D}|}\}$ computing devices (e.g., GPUs) and a set of training jobs $\mathcal{J} = \{j_1, ..., j_{|\mathcal{J}|}\}$ generated by NAS approaches. Each device d_i has a limited memory d_i^{mem}. The architecture of each job can be depicted as a data-flow graph (DFG) $\mathcal{G}_i(\mathcal{N}_i, \mathcal{E}_i)$. Here, $\mathcal{N}_i$ is the set of nodes belonging to graph $\mathcal{G}_i$, $\mathcal{E}_i$ is the set of directed edges defining the dependence among nodes. A single node in graph $\mathcal{G}_i$ represents an operator (or a sub-graph) with one or multiple input and output tensors. Each node has its own runtime and memory footprint, denote as n_{ij}^t the execution time of node n_{ij} and n_{ij}^{mem} the memory occupied by n_{ij}. The training time of job j_i in one iteration is $j_i^t = \sum_{n_{ij} \in \mathcal{N}_i} n_{ij}^t$. And the memory occupied in the training process of j_i is $j_i^{\text{mem}} = \sum_{n_{ij} \in \mathcal{N}_i} n_{ij}^{\text{mem}}$.

3.2 Batching

If $\mathcal{J}_K = \{j_1, ..., j_k\}$ is a job set selected for operator batching, assuming that all nodes can be divided into b categories according to their attributes. The nodes in the same category can be batched together. We denote as $N_i = \{n_i^1, ..., n_i^{|N_i|}\}$ the nodes in the ith category. After batching, N_i will be replaced by a new BatchNode B_i. The execution time B_i^t is usually smaller than N_i^t. We denote as $p_i^t = N_i^t - B_i^t$ the benefit after batching N_i. The input of B_i need to be concated along the channel dimension and the output are split when the successor node of B_i is not BatchNode, which brings extra overhead in time. Denote as $\check{\mathcal{J}}_K$ the production of batching $\mathcal{J}_K$. All benefits and costs are $P_K^t = \sum_{i=1}^b p_i^t$ and B_K^t, respectively.

3.3 Problem Definition

Based on the above system model, given the set of computing devices $\mathcal{D}$ and jobs $\mathcal{J}$, the execution time j_i^t and occupied memory j_i^{mem} of each job j_i, as well as the possible overhead in operator batching process, our problem is to select the jobs with the most similar model architecture for operator batching without exceeding the device memory limit. Our goal is to maximize the utilization of devices by maximizing the average throughput of training models. Divide the task set $\mathcal{J}$ into several subsets $\mathcal{S} = \{s_1, ..., s_{|\mathcal{S}|} | \forall i \neq j, s_i \cap s_j = \varnothing, s_1 \cup ... \cup s_{|\mathcal{S}|} = \mathcal{J}\}$ based on their similarity. The training jobs in s_i are $\mathcal{J}_i$. For each set s_i, we need to find a batching strategy that maximize $P_i^t - B_i^t$, making $\check{\mathcal{J}}_i^t$ as small as possible. A smaller $\check{\mathcal{J}}_i^t$ means the training process has higher throughput.

4 Algorithm Design

There are several parts to deal with operator batching between multiple models. The first part is to calculate the similarity of two models (Algorithm 1) and then cluster multiple models based on similarity (Algorithm 2). The next part is the design of batching strategy of clustered models (Algorithm 3).

4.1 Clustering Based on Model's Similarity

Since each model can be represented by a DFG, the similarity between models correlates with the similarity between graphs. The methods of measuring graph similarity include maximum common subgraph [2], graph edit distance [6], graph

```
Algorithm 1: similarity
1 Input job $j_i, j_k$
2 Output similarity of $j_i$ and $j_k$
3 Let $H_i$ and $H_k$ be the hash value lists of the topologically sorted nodes from
    graph $g_i$ and $g_k$ ,respectively;
4 $l_{ik} \leftarrow$ length of the longest common subsequence of $H_i$ and $H_k$;
5 $n_i, n_k \leftarrow$ the number of nodes of $j_i$ and $j_k$;
6 return $\frac{2 \times l_{ik}}{n_i + n_k}$
```

isomorphism [5], etc. They cannot usually be solved in polynomial time. We made some modifications based on the longest common subsequence (LCS) and calculated the similarity between models by simplifying the graph's structure. We describe the details in Algorithm 1.

Algorithm 1 topologically sorts the nodes of the model's graph and sets the hash value of each node according to its parameters and attributes. Nodes with the same hash value mean they can be batched together. Therefore, we use an ordered list of hash values H_i to approximate the architecture of the original model's graph g_i (Line 3). And refer to the idea of LCS (Line 4), the final result l_{ik} can be used for measuring jobs' similarity (Line 5 to Line 6).

By approximately calculating the similarity of the models by Algorithm 1, we can cluster the job set $\mathcal{J}$. Divide $\mathcal{J}$ into several subsets based on similarity among models. The number of models in each subset depends on the sum of the model memory, which cannot exceed the device memory limit. We describe the details of how to cluster job set $\mathcal{J}$ in Algorithm 2. At the beginning of each round of clustering, select a job j_i that has not been clustered from the job set and remove it from $\mathcal{J}$ (Line 6). Assign j_i to set s (Line 7). When the model memory in s does not exceed the limit, select a job from the unclustered job set $\mathcal{J}$ and the clustered job set s respectively, and their similarity is the highest among all the current jobs (Line 8 to Line 9). Add candidate model to s without exceeding memory constraints and remove it from $\mathcal{J}$ (Line 10 to Line 12). Otherwise, restart the next round of clustering (Line 13 to Line 15).

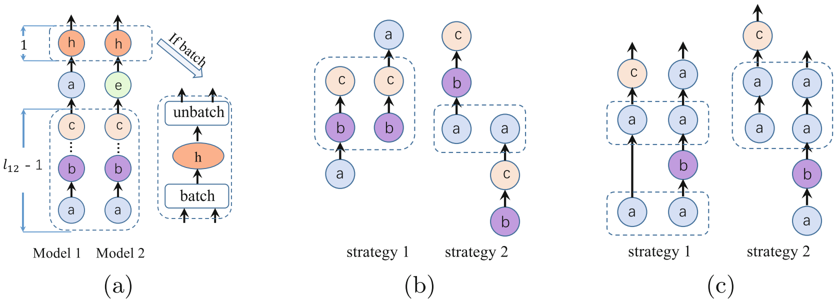


Fig. 4. Possible situations of operator batching

Algorithm 2: clustering

```
Input training jobs $\mathcal{J}$
Output clustered job set $\mathcal{S} = \{s_1, s_2, ...\}$
$\mathcal{S} \leftarrow \varnothing$;
while $\mathcal{J} \neq \varnothing$ do
    $s \leftarrow \varnothing$;
    randomly select a model $j_i$ from $\mathcal{J}$ and remove it from $\mathcal{J}$;
    $s \leftarrow s \cup j_i$;
    while $\mathcal{J} \neq \varnothing$ and $s$ does not exceed device memory do
        $j_{\text{cand}} \leftarrow \arg\max_{j_m \in \mathcal{J}} \max_{j_s \in s} \texttt{similarity}(j_s, j_m)$;
        if $s \cup j_{cand}$ won't exceed device memory then
            $s \leftarrow s \cup j_{\text{cand}}$;
            Remove $j_{\text{cand}}$ from $\mathcal{J}$;
        else
            $\mathcal{S} \leftarrow \mathcal{S} \cup s$;
            break;
```

4.2 Design of Batching Strategy

We first consider batching strategy design for two models and then extend it to multiple models. For a pair of similar models j_1 and j_2, we can get l_{12} (the maximum number of operators can be batched between model j_1 and j_2) by LCS. However, greedy batching of all l_{12} operators maybe not be guaranteed to bring the most benefits. Figure 4(a) shows one possible scenario. Because the length of the operator sequence that can be continuously batched is too short, the benefit of batching operator h may be less than the additional cost of integrating and splitting tensors, leading to negative returns. In this case, the strategy that batching the maximum number of operators is suboptimal.

Besides, the benefits of operator batching are also related to the type of operators. As Fig. 4(b) shows, there are two batch strategies. Although strategy 2 batches fewer operators, it may yield greater benefits than strategy 1. The number of breakpoints also affects the training time of the batched model. As Fig. 4(c) shows, there are two types of fusion strategies that batch the same type and number of operators. It can be concluded that strategy 2 is better than strategy 1 because of fewer breakpoints and less overhead.

Therefore, in order to maximize the benefits of operator batching and reduce the additional overhead, we propose `DPBat` (Algorithm 3), which takes breakpoints, operator types, etc. into account. We guide the multi-model operator batching through the optimal batching strategy of the two models. The details are described in Algorithm 3. We use a 4-dimensional array to record the net benefit generated during the operator batching process. For $dp[i][j][0..1][0..1]$, the 0 and 1 in the last two dimensions indicate whether the ith and jth elements are in the *batched* state, respectively. When the last two dimensions are 1 simultaneously, it means that the ith and jth elements are identical and can

$$dp[i][j][1][1] = max\begin{cases} dp[i-1][j-1][1][1] + p & \text{continuous batching} \\ dp[i-1][j-1][0][1] + p - batch\ cost & \text{start a new continuous batching} \\ dp[i-1][j-1][1][0] + p\ - batch\ cost & \text{start a new continuous batching} \\ dp[i-1][j-1][0][0] + p\ - batch\ cost & \text{start a new continuous batching} \end{cases}$$

$$dp[i][j][1][0] = max\begin{cases} dp[i][j-1][1][1] - unbatch\ cost & \text{end a continuous batching} \\ dp[i][j-1][1][0] & \text{phase without batch} \\ dp[i-1][j][0][0] & \text{phase without batch} \\ dp[i-1][j][1][0] & \text{phase without batch} \end{cases}$$

$$dp[i][j][0][1] = max\begin{cases} dp[i-1][j][1][1] - unbatch\ cost & \text{end a continuous batching} \\ dp[i-1][j][0][1] & \text{phase without batch} \\ dp[i][j-1][0][0] & \text{phase without batch} \\ dp[i][j-1][0][1] & \text{phase without batch} \end{cases}$$

$$dp[i][j][0][0] = max\begin{cases} dp[i-1][j][0..1][0] & \text{phase without batch} \\ dp[i][j-1][0][0..1] & \text{phase without batch} \end{cases}$$

Fig. 5. Transition equation

be batched. Other values indicate that the ith and jth elements are different and can not be batched. In addition to adding the benefit p of batching, dp also needs to subtract the corresponding batch/unbatch cost at the breakpoint. The specific transition equation is shown in Fig. 5. For a job set s_i, which includes multiple models with similar architectures. We select a model from s_i and $\check{s}_i$ respectively. Their similarity is the highest among all current model pairs. $\check{s}_i$ stores those models that have been batched (Line 6 to Line 8). Using the transition equation in Fig. 5 to calculate the maximum net benefit of batching two models. The batching strategy λ of two models corresponding to the maximum value in dp is optimal. We incorporate the strategy λ obtained at each round into the final result λ^* until job set s_i becomes empty. (Line 9 to Line 11).

Algorithm 3: DPBat

```
1  Input similar job set $s_i = \{j_{i1}, j_{i2}, ...\}$
2  Output batching strategy $\lambda^*$
3  $\check{s}_i \leftarrow \{j_{i1}\}$ and remove $j_{i1}$ from $s_i$;
4  $\lambda^* \leftarrow \varnothing$ ;
5  while $s_i \neq \varnothing$ do
6      Select a pair of jobs $j_1, j_2$ with the highest similarity, $j_1 \in s_i$, $j_2 \in \check{s}_i$. Let
         $H_1, H_2$ be the hash value lists of their topologically sorted nodes;
7      $\check{s}_i \leftarrow \check{s}_i \cup j_1$ ;
8      Remove $j_1$ from $s_i$;
9      Calculate the net benefit brought by different operator batching strategies
         using equation in figure 5 ;
10     Let $\lambda$ be the batching strategy corresponding to the maximum value in dp;
11     $\lambda^* \leftarrow \lambda^* \cup \lambda$
```

5 Evaluation

In this section, we evaluate the performance of `DPBat` in real NAS scenarios and compare it with three baselines. Overall, the key findings include: `DPBat` significantly improves training efficiency and reduces the overhead of operator batching. `DPBat` achieves up to 3.7× higher training throughput than running each job serially, which is a common practice employed by the AutoML framework.

5.1 Experiment Settings

To evaluate `DPBat` in real scenarios, we used Microsoft's NAS tool NNI which can separate the cross model optimization from model generation. We follow the same configurations as Retiarii [14], select representative NAS solutions MnasNet [11], MobileNetV2-based model space and reinforcement learning exploration strategy. In the experiment, the NAS approach will generate 1000 models in 10 batches(100 models each batch). `DPBat` and the other baselines are given the same set of models in the same order for a fair comparison. These models use the same batch size, which is 8 images (ImageNet's training images [3]) per mini-batch. We implemented the experiments on 4 NVIDIA Tesla P100 GPUs of 16GB GPU memory. The performance is measured by averaging the throughput over 1000 mini-batches.

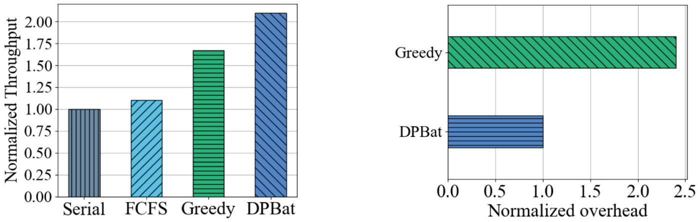
5.2 Three Baselines

We compare `DPBat` with the following three baselines.

- **Serial**: each training job is executed on a single accelerator, which is employed by most DL frameworks [14].
- **FCFS**: *FCFS* is the policy used by NNI's cross-graph optimization engine and it clusters the jobs by order of arrival rather than similarity. Training jobs arrive in batches of 100 models, sequentially dividing the task set $\mathcal{J}$ into several subsets. Each subset contains the maximum number of models before the GPU runs out of memory. For example, training jobs $\{j_1, j_2, ..., j_i\}$ are divided into subset s_1, $\{j_{i+1}, j_{i+2}, ..., j_k\}$ are divided into s_2 and so on. The design of the operator batching strategy is also extended from two models to multiple models. For a pair of models, use the idea of BFS to compare the DFG of the two models layer by layer and stop batching when the layer depth is the same but the layer nodes are different.
- **Greedy**: *Greedy* is the policy described in Retiarii [14] which fuses all common operators. It does not consider batch/unbatch cost and different benefits of batching different kinds of operators, which means setting the batch/unbatch cost and benefit in `DPBat` to 0.

5.3 Experiment Results

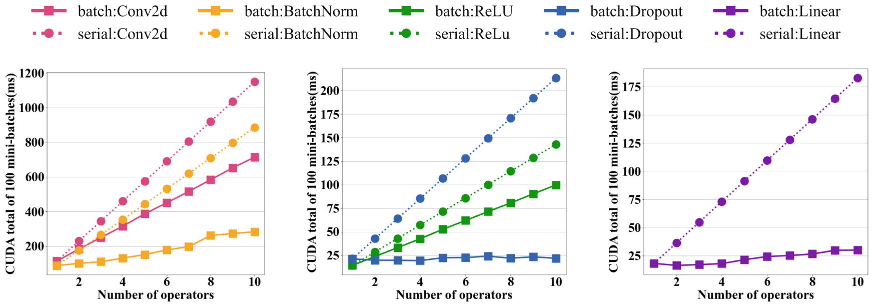
In this part, we present the experimental results on 1000 models and dissect the source of improvement brought by `DPBat`. In all cases, our algorithm `DPBat` outperforms the other baseline.



(a) Training throughput (b) Batching overhead

Fig. 6. Performance of different algorithms

The Overall Performance. Figure 6(a) illustrates the four algorithms' average throughput of 1000 models. DPBat achieves higher throughput than all baselines, 2.1× (up to 4.7×) over **Serial**, 1.92× over **FCFS**, 1.25× over **Greedy**. **FCFS** cannot make full use of the similarity between models because of the lack of clustering. Moreover, its batching strategy cannot select all the operators that can be batched either. **Greedy** focuses on the number of operators that can be batched. While maximizing the number of batched operators, the additional batch/unbatch overhead increases. It also ignores the fact that the benefit of batched operators is related to the operator's type. Figure 6(b) shows the average batch/unbatch cost of 1000 models. Because taking breakpoints into account, DPBat can significantly reduce additional overhead compared to **Greedy**.



(b) (c) (d)

Fig. 7. Performance of different operators

Sources of Improvements. To understand why DPBat achieves better performance than the other baselines, we perform a deeper analysis using the PyTorch profiler to measure the time and memory consumption of the model's operators. The advantage of DPBat and **Greedy** is that they can dynamically select batched models according to models' characteristics, which leads to much higher utilization of GPU memory. Batched operators enable more fine-grained GPU sharing by using less GPU memory to increase SIMD utilization. `DPBat` performs better than Greedy mainly because of its awareness of operator batching costs.

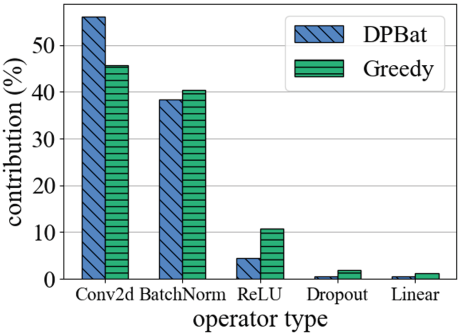


Fig. 8. Contribution of different operators to the performance improvement

We analyze the contribution of different types of operators to performance improvement. The operator types in the training models mainly include ReLU, Dropout, Linear, BatchNorm, and Conv2d. Figure 7 shows the running time of different types of operators. When batching the same number of operators, the benefits are obviously different. Batching Conv2d brings the highest benefits, followed by BatchNorm. As Fig. 8 shows, Conv2d and BatchNorm are the main sources of the benefit brought by operator batching. `DPBat` is better than **Greedy** in picking out the type of operator that brings the most performance improvement.

6 Conclusion

In this paper, we study the multi-model operator batching strategy in the NAS scenario. By characterizing the model architecture as a DFG, calculating the similarity of graphs approximately, and batching common operators of models to improve training efficiency. Our objective is to maximize the throughput of model training per unit time. We propose a heuristic algorithm named `DPBat` to guide the operator batching among multiple models. Based on Microsoft's AutoML framework NNI, we apply `DPBat` to real NAS scenarios. Experiment results show that `DPBat` significantly improves training efficiency and reduces the overhead of operator batching. Furthermore, `DPBat` achieves up to 3.7× higher training throughput than running each job on a separate accelerator, which is a common practice employed by the AutoML framework. Although we only focus on models whose DFGs are directed acyclic graphs, we believe our results will inspire future work on optimizing batching strategy between multiple models in a more general setting.

Acknowledgements. This work is partially supported by NSFC under Grant 62132009, and the Fundamental Research Funds for the Central Universities at China.

References

1. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
2. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. Pattern Recogn. Lett. **19**(3–4), 255–259 (1998)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Fortin, S.: The graph isomorphism problem (1996)
6. Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. Pattern Anal. Appl. **13**(1), 113–129 (2010)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25. Curran Associates, Inc. (2012)
9. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
10. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
11. Tan, M., et al.: MnasNet: platform-aware neural architecture search for mobile. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2820–2828 (2019)
12. Wang, S., Yang, P., Zheng, Y., Li, X., Pekhimenko, G.: Horizontally fused training array: an effective hardware utilization squeezer for training novel deep learning models. Proc. Mach. Learn. Syst. **3**, 599–623 (2021)
13. Wu, Y., et al.: Google's neural machine translation system: bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)
14. Zhang, Q., et al.: Retiarii: a deep learning exploratory-training framework. In: 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2020), pp. 919–936 (2020)

Reliability-Aware Comprehensive Routing and Scheduling in Time-Sensitive Networking

Jiaqi Feng, Tong Zhang(✉), and Changyan Yi

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
{fengjiaqi,zhangt,changyan.yi}@nuaa.edu.cn

Abstract. Nowadays, Time-Sensitive Networking (TSN) has widespread application in many industrial fields, aiming to provide deterministic low-latency network transmission. Traffic in TSN is roughly divided into three categories: Time-Triggered (TT) traffic, Audio-Video-Bridging (AVB) traffic, and Best-Effort (BE) traffic. These different traffic needs to travel the network satisfying their respective reliability and performance requirements. Existing traffic routing and scheduling mechanisms mainly focus on TT traffic but pay little attention to other traffic types. In this paper, we present a novel Optimization Modulo Theories (OMT) formulation for a comprehensive traffic routing and scheduling problem in TSN. Based on this, we propose a novel reliability-aware routing and scheduling mechanisms for all traffic types, in order to improve their own transmission reliability and performance. We conduct extensive evaluations to validate the effectiveness of the proposed mechanisms, and the results confirm that the proposed mechanism can really guarantee the reliability and latency requirements of TT flows and improve the transmission utility of all flows to a large extent.

Keywords: Time-Sensitive Networking · Optimization Modulo Theories · Routing · Scheduling · Reliability-aware

1 Introduction

With the development of Internet and emergence of new industries, extensive real-time applications need network connections with millisecond-level delay or even lower [1]. In this case, the IEEE Time-Sensitive Networking working group develops a series of standards and enhance Ethernet to TSN. TSN complies with the standard Ethernet protocol system, thus can interconnect isolated industrial control networks. Through various enhancing mechanisms like time synchronization, traffic classification, shaping, scheduling and reliability standards, TSN can provide low-latency and low-jitter transmission guarantees and support traffic of different service types to transmit in the common network.

TSN divides traffic into three categories: Time-Triggered (TT) traffic, Audio-Video-Bridging (AVB) traffic, and Best-Effort (BE) traffic. TT traffic represents

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 243–254, 2022.
https://doi.org/10.1007/978-3-031-19214-2_20

the highest-criticality traffic, which is used to provide deterministic low-latency transmission for time-sensitive applications. AVB is a type of traffic with lower criticality, which mainly supports various multimedia applications based on audio and video and has more slack delay requirement than TT traffic. BE is the traffic type with the lowest criticality, which includes traditional Internet traffic and has no performance guarantees. To provide different levels of transmission reliability for these traffic types, Frame Replication and Elimination for Reliability (FRER) mechanism [2] allows the network to provide redundant paths for critical frame transmissions and assigns each traffic type a redundancy level (RL) based on how critical it is. And to support different levels of delay and bandwidth guarantees, different traffic types enter different queues at switch egress ports and the Time Aware Shaper (TAS) controls the opening and closing of gates at the exit of each queue and in-queue frame transmission according to the gate control list (GCL). In this case, to optimize the overall network transmission performance, the traffic control mechanism should characterize all routing paths and transmission time of all types of traffic. However, existing mechanisms mainly focus on TT traffic scheduling under various performance goals and scenarios [3–11] but leave out other traffic types' utilities and lead to suboptimal overall transmission performance.

In this paper, we formalize the joint routing and scheduling problem for all types of traffic with diverse reliability and delay requirements as a mixed ingeger programming problem based on Optimization Modulo Theories (OMT). Therefore, a larger solution space can provide better scheduling options, either improving the schedulability or enabling better scheduling quality. Based on formalization, we propose a reliability-aware comprehensive routing and scheduling mechanisms, overally optimizing transmission utility of all traffic in the condition of meeting delay and reliability requirements. We distinguish the reliability level of different traffic, and then the routing paths and GCLs are solved, improving the overall network transmission efficiency.

2 Related Work

At present, there are many studies on TT flow routing and scheduling, but less work focuses on other traffic types. Craciunas et al. put forward a complete set of constraints and some optimization suggestions in [3], but they do not propose a complete scheduling optimization. In [4], a scheduling mechanism of TT flows with the goal of minimizing the sum of application response time is proposed. In [5], a variety of solutions are proposed to minimize the TT queue number, including ILP and greedy algorithms. Pahlevan et al. [6] propose a list algorithm to minimize the total transmission time, and Gavrilut et al. [7] study the scheduling of TT and AVB flows with the objective of maximizing the soft real-time flow utility function. In [8], a greedy algorithm is proposed to schedule AVB flows in order to maximize bandwidth utilization. In [9], a strict priority algorithm is proposed to schedule all flows. However, they do not take into account the effect of routing on the network, which can lead to bandwidth underutilization and reduced schedulability. With the study of routing, Schweissguth et al. [10] proposed an integer linear programming to minimize the sum of all flow latencies. In [11], the author pays attention to schedulability of AVB flows, iteratively executes k-shortest path algorithm and GRASP algorithm to search for the optimal solution under this path, but the shortest path algorithm is not necessarily an effective method that helps to enhance schedulability. Moreover, we can see that reliability level has not been elaborately characterized in existing studies, while it is an indispensable prerequisite of traffic going through the network reliably.

3 System Model

In this section, we establish the network architecture model of TSN, including end devices, switches and physical links. On this basis, we model TSN traffic according to their temporal and spatial characteristics.

3.1 Network Architecture Model

The network can be represented by a directed graph $G = (V, E)$, where V is the set of nodes, which refers to all devices in the network, and includes subsets of switches and end stations, denoted as ES and SW separately. Therefore, $V = ES \cup SW$. E is the set of directed edges connecting nodes to each other. The directed edge from $v_a \in V$ to $v_b \in V$ represents a unidirectional communication link from v_a to v_b. Hence, the full-duplex communication link between v_a and v_b is represented as two edges, $[v_a, v_b] \in E$ and $[v_b, v_a] \in E$. Figure 1 shows a network with four end stations and six switches, where the black double arrow represents the physical full duplex link, and the blue dotted lines indicate different redundant traffic routes in the FRER mechanism.

Each switch egress port has 8 queues to store different types of traffic. Link attributes are represented by a tuple $\langle [v_a, v_b].s, [v_a, v_b].d \rangle$, where $[v_a, v_b].s$ denotes the bandwidth capacity of the link, and $[v_a, v_b].d$ denotes the propagation delay on the medium. This model can describe links with different capacities and delays.

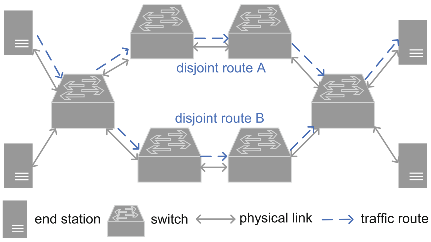


Fig. 1. Network model in FRER

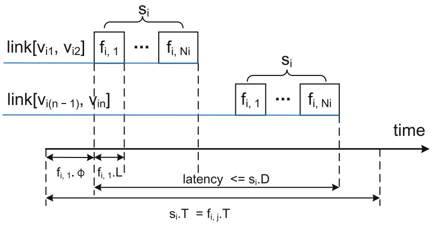


Fig. 2. Traffic model and parameters

3.2 Traffic Model

In TSN, devices communicate with each other through flows. A flow refers to a group of periodic or aperiodic transmission of unicast data from a sender to a receiver, and a frame is the data transfer unit within a flow. Typically, the sender and receiver are both end stations, and the intermediate forward nodes are switches. We use S to denote the set of all flows in a network, $s_i \in S$ to denote a flow. The flow path from the source node to the destination node through intermediate nodes is expressed as $[[v_{i1}, v_{i2}], ..., [v_{i(n-1)}, v_{in}]]$. The attributes of a flow are represented by a tuple $\langle s_i.T, s_i.D, s_i.S, s_i.rl\rangle$, where $s_i.T$ represents the period of the flow, $s_i.D$ indicates the maximum allowable end-to-end delay, $s_i.S$ denotes the message size in one cycle, and $s_i.rl$ is the redundancy level, which means the number of frame replicas. If s_i is aperiodic, $s_i.T = \infty$. If s_i has no deadline, $s_i.D = \infty$.

The j^{th} frame transmission instance of flow s_i on link $[v_a, v_b]$ is defined by $f_{i,j}^{[v_a,v_b]}$, where $[v_a, v_b]$ is a part of the communication path of s_i, and the subscript i and j represent the indices of flow and frame, respectively. Because of Ethernet's Maximum Transmission Unit (MTU), a message may be divided into multiple frames, each of which is smaller than or equal to the MTU. The attributes of a frame instance are represented by a tuple $\langle f_{i,j}^{[v_a,v_b]}.\rho, f_{i,j}^{[v_a,v_b]}.T, f_{i,j}^{[v_a,v_b]}.\phi, f_{i,j}^{[v_a,v_b]}.L\rangle$. The four elements represent the size of frame in bytes, the period, offset, and transmission duration of the frame, respectively. $f_{i,j}^{[v_a,v_b]}.L$ is jointly determined by the link capacity $[v_a, v_b].s$ and frame size $f_{i,j}^{[v_a,v_b]}.\rho$, and the calculation formula is $f_{i,j}^{[v_a,v_b]}.L = \frac{f_{i,j}^{[v_a,v_b]}.\rho}{[v_a,v_b].s}$. We define the variable $s_i^{[v_a,v_b]}.q$ as the ID of the queue in which the flow resides. Figure 2 shows the traffic model and the associated parameters.

4 Joint Routing and Scheduling Mechanism

In this section, we formalize the reliability-aware joint routing and scheduling problem. We first propose the routing constraints, guaranteeing flows are transmitted along practical paths and follow reliability levels. Then we propose scheduling constraints, which not only formally regulate the transmission behavior of flows, but also ensure the definite end-to-end performance metrics. Finally, we analyze the utilities of different types of traffic as well as their relationships with the end-to-end delay, and get their utility functions. According to these functions, the optimization objective is set to improve the transmission utility of all traffic.

4.1 Route Constraints

In order to describe the routing effect, we define a variable $u_{i,m}^{[v_a,v_b]}$, which denotes whether the m^{th} replica of flow s_i uses link $[v_a, v_b]$. The value of $u_{i,m}^{[v_a,v_b]}$ equals 1 if the flow replica passes through $[v_a, v_b]$ and 0 otherwise.

Source Node Constraint. For each flow s_i in the network, it must start from one end station and finish at another end station. Therefore, the flow will only depart from but do not enter into the source node.

$$\begin{gathered}\forall s_i \in S, \forall v_x \in V, \forall m \in \{x \in \mathbb{N}^+ \mid 0 < x \leq s_i.rl\}: \\ \sum_{[v_{i1},v_x] \in E} u_{i,m}^{[v_{i1},v_x]} - \sum_{[v_x,v_{i1}] \in E} u_{i,m}^{[v_x,v_{i1}]} = 1\end{gathered} \tag{1}$$

For each s_i, the number of its outgoing links must be larger by 1 than that of its ingoing links at its source node v_{i1}.

Destination Node Constraint. For each flow s_i, its route must reach the destination node. Therefore, similar to the previous constraint, the destination node of the route only has incoming but no outgoing. The constraint is as follows:

$$\begin{gathered}\forall s_i \in S, \forall v_x \in V, \forall m \in \{x \in \mathbb{N}^+ \mid 0 < x \leq s_i.rl\}: \\ \sum_{[v_{in},v_x] \in E} u_{i,m}^{[v_{in},v_x]} - \sum_{[v_x,v_{in}] \in E} u_{i,m}^{[v_x,v_{in}]} = -1\end{gathered} \tag{2}$$

For each s_i, the number of its ingoing links must be larger by 1 than that of its outgoing links at its destination node v_{in}.

Intermediate Node Constraint. Besides source and destination nodes, each flow will also go through intermediate nodes. The intermediate nodes have two situations. If flow s_i passes through an intermediate node, there should be one input and one output. If flow s_i does not pass through the node, there should be no input or output. These two movements can be represented by the constraint: as follows:

$$\begin{gathered}\forall s_i \in S, \forall v_a \in V \backslash \{v_{i1}, v_{in}\}, \forall m \in \{x \in \mathbb{N}^+ \mid 0 < x \leq s_i.rl\}: \\ \sum_{[v_a,v_b] \in E} u_{i,m}^{[v_a,v_b]} - \sum_{[v_b,v_a] \in E} u_{i,m}^{[v_b,v_a]} = 0\end{gathered} \tag{3}$$

where v_a, v_b represent the nodes on flow s_i's path other than the source and destination nodes of flow s_i.

Avoid Loop Constraint. When looking for the right route, there is no doubt that one link should not be passed repeatedly, and the generation of loops should be avoided. Therefore, we add constraints to make it impossible for all flows to be routed via the loop. The constraint is as follows:

$$\begin{gathered}\forall s_i \in S, \forall v_a, v_b \in V, \forall m \in \{x \in \mathbb{N}^+ \mid 0 < x \leq s_i.rl\}: \\ \sum_{[v_a,v_b] \in E} u_{i,m}^{[v_a,v_b]} \leq 1\end{gathered} \tag{4}$$

Non-overlap Constraint. TSN addresses reliability in a spatially redundant manner by FRER, which relies on multi-path routing. For traffic with different criticality levels, redundancy levels are assigned in advance. The routes of flow replicas generated by a certain flow cannot overlap except the source and destination links.

$$\forall s_i \in S, \forall v_x \in V \backslash \{v_{i1}, v_{i2}, v_{i(n-1)}, v_{in}\}, \forall m, r \in \{x \in \mathbb{N}^+ |\, 0 < x \leq s_i.rl\}, m \neq r : \\ \sum_{[v_x,v_a] \in E} u_{i,m}^{[v_x,v_a]} + \sum_{[v_x,v_b] \in E} u_{i,r}^{[v_x,v_b]} \leq 1 \tag{5}$$

4.2 Scheduling Constraints

The task of scheduling is to get GCLs, thus each flow's queue assignment on each passing egress port $s_i^{[v_a,v_b]}.q$ and each frame's transmission offset on each passing egress port $f_{i,j}^{[v_a,v_b]}.\phi$ should be decided.

Frame Constraint. For frames of any flows, this constraint guarantees that each frame must be transmitted no earlier than time zero but within the frame cycle.

$$\forall s_i \in S, \forall [v_a, v_b] \in E, \forall f_{i,j}^{[v_a,v_b]} \in s_i^{[v_a,v_b]}, \forall m \in \{x \in \mathbb{N}^+ |\, 0 < x \leq s_i.rl\}, u_{i,m}^{[v_a,v_b]} \neq 0 : \\ (f_{i,j}^{[v_a,v_b]}.\phi \geqslant 0) \wedge (f_{i,j}^{[v_a,v_b]}.\phi + f_{i,j}^{[v_a,v_b]}.L \leqslant f_{i,j}^{[v_a,v_b]}.T) \tag{6}$$

Link Constraint. For any two frames, this constraint is to ensure they will not overlap in time on the same link.

$$\forall s_i, s_j \in S, i \neq j, \forall [v_a, v_b] \in E, \forall f_{i,k}^{[v_a,v_b]} \in s_i^{[v_a,v_b]}, \\ \forall f_{j,l}^{[v_a,v_b]} \in s_j^{[v_a,v_b]}, \forall \alpha \in [0, \frac{hp_i^j}{s_i.T} - 1], \forall \beta \in [0, \frac{hp_i^j}{s_j.T} - 1], \\ \forall m \in \{x \in \mathbb{N}^+ |\, 0 < x \leq s_i.rl\}, \forall r \in \{x \in \mathbb{N}^+ |\, 0 < x \leq s_j.rl\}, u_{i,m}^{[v_a,v_b]} + u_{j,r}^{[v_a,v_b]} \geqslant 2 : \\ (f_{i,k}^{[v_a,v_b]}.\phi + \alpha \times f_{i,k}^{[v_a,v_b]}.T \geqslant f_{j,l}^{[v_a,v_b]}.\phi + \beta \times f_{j,l}^{[v_a,v_b]}.T + f_{j,l}^{[v_a,v_b]}.L) \vee \\ (f_{j,l}^{[v_a,v_b]}.\phi + \beta \times f_{j,l}^{[v_a,v_b]}.T \geqslant f_{i,k}^{[v_a,v_b]}.\phi + \alpha \times f_{i,k}^{[v_a,v_b]}.T + f_{i,k}^{[v_a,v_b]}.L) \tag{7}$$

where $hp_i^j = lcm(s_i.T, s_j.T)$ is the hyper period of s_i and s_j. The hyper period is calculated by the function $lcm()$ that gives the least common multiple of flows' periods. And α, β are indices of flows' round within the hyper period.

Stream Transmission Constraint. For each frame, its transmission must be along the flow routing path in order. According to the time synchronization protocol, we define δ to be the worst-case difference between any two synchronized local clocks. The constraint is as follows:

$$\forall s_i \in S, \forall [v_a, v_x], [v_x, v_b] \in E, \forall f_{i,j}^{[v_a,v_x]} \in s_i^{[v_a,v_x]}, \forall f_{i,j}^{[v_x,v_b]} \in s_i^{[v_x,v_b]}, \\ \forall m \in \{x \in \mathbb{N}^+ |\, 0 < x \leq s_i.rl\}, u_{i,m}^{[v_a,v_x]} + u_{i,m}^{[v_x,v_b]} \geqslant 2 : \\ (f_{i,j}^{[v_x,v_b]}.\phi - \delta) \geqslant (f_{i,j}^{[v_a,v_x]}.\phi + f_{i,j}^{[v_a,v_x]}.L + [v_a, v_x].d) \tag{8}$$

This constraint states that only after received on the previous link $[v_a, v_x]$ can a frame be scheduled on the subsequent link $[v_x, v_b]$.

Latency Constraint. In order to satisfy the timing requirements of flows, we need to make sure that difference between the arrival and sending time of a flow does not exceed the given maximum end-to-end latency. $[v_{i1}, v_{i2}]$ denotes the link starting from the source node of flow s_i and $[v_{i(n-1)}, v_{in}]$ the last link reaching the destination node. At the same time, $f_{i,1}^{[v_a,v_b]}$ and $f_{i,N_i}^{[v_a,v_b]}$ are first and last frames of $s_i^{[v_a,v_b]}$ respectively, where $N_i = \lceil \frac{s_i.S}{MTU} \rceil$. And consider the worst-case local clock differences between nodes δ, the constraint is shown below:

$$\begin{aligned} &\forall s_i \in S, \forall m \in \{x \in \mathbb{N}^+ |\ 0 < x \leq s_i.rl\}, u_{i,m}^{[v_{i1},v_{i2}]} + u_{i,m}^{[v_{i(n-1)},v_{in}]} \geqslant 2 : \\ &\quad (f_{i,1}^{[v_{i1},v_{i2}]}.\phi + s_i.D) \geq (f_{i,N_i}^{[v_{i(n-1)},v_{in}]}.\phi + f_{i,N_i}^{[v_{i(n-1)},v_{in}]}.L) \end{aligned} \tag{9}$$

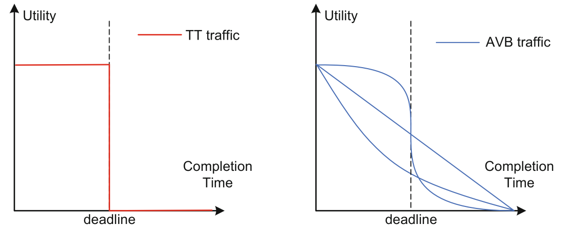
Stream Isolation Constraint. In order to ensure deterministic transmission order among flows, only when all frames of one flow leave the downstream egress queue can the frames of another flow be scheduled from the upstream port. In this context, the frame transmission order on the egress link is deterministic. The constraint is shown below:

$$\begin{aligned} &\forall s_i, s_j \in S, i \neq j, \forall [v_a, v_b] \in E, s_i^{[v_a,v_b]}.q = s_j^{[v_a,v_b]}.q, \forall f_{i,k}^{[v_a,v_b]} \in s_i^{[v_a,v_b]}, \forall f_{j,l}^{[v_a,v_b]} \in s_j^{[v_a,v_b]}, \\ &\quad \forall \alpha \in [0, \frac{hp_i^j}{s_i.T} - 1], \forall \beta \in [0, \frac{hp_i^j}{s_j.T} - 1], \forall m \in \{x \in \mathbb{N}^+ |\ 0 < x \leq s_i.rl\}, \\ &\quad \forall r \in \{x \in \mathbb{N}^+ |\ 0 < x \leq s_j.rl\}, u_{i,m}^{[v_a,v_b]} + u_{j,r}^{[v_y,v_a]} \geqslant 2 \text{ or } u_{j,r}^{[v_a,v_b]} + u_{i,m}^{[v_x,v_a]} \geqslant 2 : \\ &\quad (f_{i,N_i}^{[v_a,v_b]}.\phi + f_{i,N_i}^{[v_a,v_b]}.L + \alpha \times s_i.T + \delta \leqslant f_{j,1}^{[v_y,v_a]}.\phi + \beta \times s_j.T + [v_y, v_a].d) \vee \\ &\quad (f_{j,N_i}^{[v_a,v_b]}.\phi + f_{j,N_i}^{[v_a,v_b]}.L + \beta \times s_j.T + \delta \leqslant f_{i,1}^{[v_x,v_a]}.\phi + \alpha \times s_i.T + [v_x, v_a].d) \end{aligned} \tag{10}$$

4.3 Optimization Objective

In TSN, different types of traffic may have different sensitivities to their end-to-end latency. The sensitivity can be represented by different utility functions of latency. As shown in Fig. 3, the red line represents the TT traffic utility. When the latency is within the deadline, the utility value remains the same, but once exceeds the deadline, the utility value will immediately drops to zero. The blue dotted lines represent utility functions of AVB traffic, whose utility values gradually decline as latency increases. As is shown in the figure, the decreasing utility function can be diverse, such as linear function, power function, Sigmoid function, and so on. BE flow has no latency requirements, so it does not have utility in context with latency.

We set $\mathcal{U}_i(t_i)$ to denote the utility function of flow s_i, where t_i refers to the end-to-end latency of flow s_i and equals $f_{i,N_i}^{[v_{i(n-1)},v_{in}]}.\phi + f_{i,N_i}^{[v_{i(n-1)},v_{in}]}.L - f_{i,1}^{[v_i1,v_i2]}$. On this basis, we determine the optimization goal of routing and



(a) TT utility function (b) AVB utility function

Fig. 3. Completion-time-dependent utility functions

scheduling, that is maximizing the transmission utility of traffic. The optimization objectives are formally expressed as follows:

$$\text{Maximize} \qquad \sum_{s_i \in S} \mathcal{U}_i(t_i)$$

$$\mathcal{U}_i(t_i) = \begin{cases} s_i.u & ,t_i \le s_i.D,\ s_i \in S^{TT}, \\ 0 & ,t_i > s_i.D,\ s_i \in S^{TT}, \\ f_i(t_i) & ,s_i \in S^{AVB}. \end{cases}$$

where $s_i.u$ is the maximum utility value of flow s_i, $f_i(t_i)$ denotes the diverse utility functions of AVB flows shown in Fig. 3.

4.4 Mechanism Design

Through solving the joint routing and scheduling problem defined above, we will get the transmission path and offsets on of every flow that satisfy all constraints and maximize the overall utility. Satisfiability Module Theory (SMT) checks the satisfiability of logical formulas in certain background theories, such as linear integer arithmetic ($\mathcal{LA}(\mathbb{Z})$) and bit-vectors ($\mathcal{BV}$). We consider that all routing and scheduling constraints are already expressed in terms of the conjunction normal form, and are thus suitable for an SMT solver to check their satisfiability. OMT is a new branch of SMT and can provide the best solution for a given minimum or maximum goal on the basis of checking satisfiability. We regard each constraint as a disjunctive paradigm in the conjunctive paradigm, and the objective function as the optimization goal of OMT. Therefore, as shown in Algorithm 1, we use the OMT solver to solve our problem, where the $check(constrains)$ function checks the satisfiability of the constraints.

On the basis of optimal solution of TT and AVB flows, we address routing and scheduling for BE flows. On one hand, BE flow is random and aperiodic, so it is impossible to know its arrival time and traffic pattern in advance. On the other hand, BE flows do not have latency requirement, but expect high throughput. In this case, each time a BE flow enters the network, we assign it the path with the most remaining idle time slots after TT and AVB flow allocations, and let it fill these remaining slots as much as possible to maximize bandwidth utilization.

Algorithm 1. Based OMT Joint Routing And Scheduling

```
Input: network topology $G(V, E)$; characteristics of all flows $S$;
Output: $Result$: offset, queue ID, and routing path for each frame of all flows, utility values for all flows;
constrains ← (1)...(10); $Result$ ← ∅;
if $check(constrains)$ then
    while $true$ do
        if $OMTsolver(constains, S)$ $output\ solution$ then
            $newResult \leftarrow OMTsolver(constrains, S)$;
            $constrains.add(utility > newResult.utility)$;
            $Result \leftarrow newResult$;
        else
            break;
        end if
    end while
end if
return $Result$
```

5 Evaluation

In this section, we conduct a series of simulations to verify the effectiveness of our proposed mechanism. Firstly, we introduce our simulation settings, and then we compare several common solutions.

5.1 Simulation Setup

The optimization solver by our code is deployed on an a machine with Intel i7-9750H 2.60 GHz CPU and 16 GB memory. The objective function and related constraints are implemented in Python and solved by the cp-model module of Google's open source software OR-Tools [12].

Linear, ring, tree, and snowflake are main topological types of industrial control networks. To evaluate the mechanisms comprehensively, we choose the actual Orion Crew Exploration Vehicle (CEV) network as the test topology (Fig. 4), which contains the above topologies locally. In order to increase the reliability of network, we add some links to facilitate spatial redundancy.

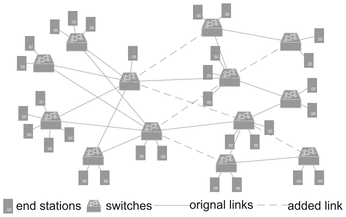


Fig. 4. Orion CEV network topology with added links

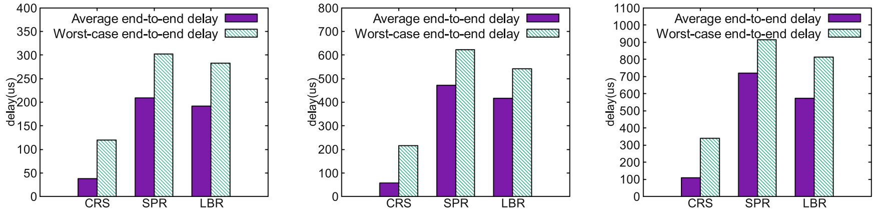
In this group of simulations, we randomly generate a set of flows, selecting the source and destination nodes of each flow stochastically from the end stations in Fig. 4. The flow sizes are generated randomly from {100B, 200B, 400B, 800B} and periods from {1 ms, 2 ms, 4 ms, 8 ms}. We set the redundancy level of TT flows to 2, and of other flows to 1. We set the time operation granularity to 1 µs, the bandwidth capacity of all links to 1 Gbps, and we ignore the link propagation delay.

5.2 Evaluation Results

We consider three different scenarios in the simulation, including 10, 20, and 30 flows respectively to build diverse traffic loads, and the flow set is a mixture of TT and AVB flows.

In our simulations, we compare our proposed OMT-based comprehensive routing and scheduling (CRS) mechanism with two-step solutions that compute routing first and then scheduling of traffic. We adopt two reference mechanisms for comparison. One is the shortest path routing (SPR), in which each flow is allocated to the shortest path in the network topology and a feasible scheduling solution is obtained on these paths, the other is the load balancing routing (LBR), which routes flows as evenly as possible on all paths and obtains a feasible scheduling solution based on these paths. The performance metrics we measured in the simulation are worst-case and average end-to-end delays of AVB flows and of all flows, as well as overall utility of all flows.

As shown in Fig. 5, in the three scenarios, under our proposed CRS, the average and worst-case end-to-end delays of AVB flows are smaller than the results under SPR and LBR. The delays of SPR are the largest, because it only considers the shortest path for routing, but does not consider bandwidth utilization or solution space of scheduling from the overall network. Then the delays of LBR are between those of CRS and SPR. On one hand, it considers load balancing. On the other hand, it does not consider routing together with scheduling, leaving out more superior joint solutions for overall utility. The overall results show that our proposed CRS can indeed improve the transmission utilities of AVB flows.

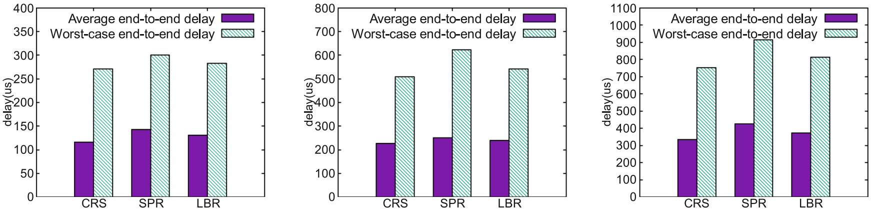


(a) Scenario 1

(b) Scenario 2

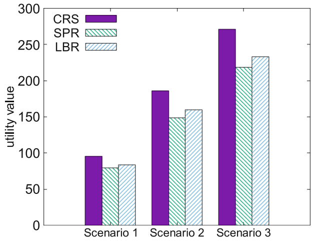
(c) Scenario 3

Fig. 5. AVB flows scheduling results under different scenarios



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Fig. 6. All flows scheduling results under different scenarios



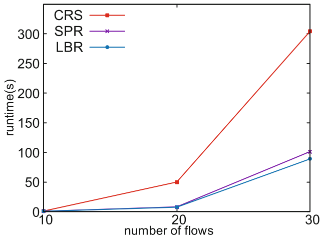


Fig. 7. Utility value for each scenario **Fig. 8.** Solver runtimes for each solution

Figure 6 shows the end-to-end delays of all flows under three scenarios. In the scenario 1 and 2 with light traffic load, the routing paths solved by LBR can already provide enough solution space for scheduling, so the joint solution of CRS does not show an obvious advantage, and its average end-to-end delays is slightly smaller than LBR. In the scenario 3 with heavy traffic load, shows a great advantage of larger solution space so its overall delay is obviously smaller than those of LBR and SPR.

Figure 7 depicts the total utility of all flows in three scenarios. The columns in three different colors represent the utility values under CRS, SPR and LBR respectively. We can see in all cases, the total utility is the largest under CRS, and as the number of flows increases, the advantage of CRS becomes more significant.

As shown in Fig. 8, we can see that as the number of flows increases, the runtimes for the three solutions also increase, especially the proposed CRS. The reason is that although the joint solution provides a larger solution space, it also requires more time to process. However, the solving process is offline, which will not harm actual flow transmission. In this case, we can pre-process the solution space to exclude some infeasible solutions in advance, which is applicable to larger and more complex networks.

6 Conclusions

In this paper, we propose a reliability-aware comprehensive routing and scheduling mechanism that is applicable to all traffic types in TSN to improve transmission performance. In the constraints of routing and scheduling, we take into account the reliability capability of TSN through spatial redundancy. Besides, we analyze the transmission utilities of different traffic types and maximize the overall utility in the optimization objective. By solving the optimization problem using OMT, we propose a comprehensive routing and scheduling mechanism. On the premise of ensuring delay requirements of TT traffic, the proposed mechanism can effectively improve the transmission utility of all traffic types. We verify our proposed mechanisms through a group of simulations. The evaluation results in multiple traffic scenarios show that the proposed mechanism can improve the overall network performance, and the transmission utility of all flows.

Acknowledgement. The authors gratefully acknowledge the anonymous reviewers for their constructive comments. This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62002165, 62132007,62002164, Natural Science Foundation of Jiangsu Province under Grant BK20200445, and Basic Science Research Fund in Nanjing University of Aeronautics and Astronautics (NS2021068).

References

1. Teener, M.J.: IEEE 802 time-sensitive networking: extending beyond AVB. IEEE Web, vol. 19 (2015). https://standards.ieee.org/
2. IEEE standard for local and metropolitan area networks-frame replication and elimination for reliability. IEEE STD 802.1CB-2017, pp. 1–102 (2017). https://doi.org/10.1109/IEEESTD.2017.8091139
3. Craciunas, S.S., Oliver, R.S., Ag, T.: An overview of scheduling mechanisms for time-sensitive networks. In: Proceedings of the Real-Time Summer School LÉcole dÉté Temps Réel (ETR), pp. 1551–3203 (2017)
4. Craciunas, S.S., Oliver, R.S.: Combined task-and network-level scheduling for distributed time-triggered systems. Real-Time Syst. **52**(2), 161–200 (2016)
5. Raagaard, M.L., Pop, P.: Optimization algorithms for the scheduling of IEEE 802.1 time-sensitive networking (TSN). Technical University of Denmark, Lyngby, Denmark, Technical report (2017)
6. Pahlevan, M., Tabassam, N., Obermaisser, R.: Heuristic list scheduler for time triggered traffic in time sensitive networks. ACM SIGBED Rev. **16**(1), 15–20 (2019)
7. Gavriluţ, V., Pop, P.: Traffic-type assignment for TSN-based mixed-criticality cyber-physical systems. ACM Trans. Cyber-Phys. Syst. **4**(2), 1–27 (2020)
8. Zhang, C., et al.: Packet-size aware scheduling algorithms in guard band for time sensitive networking. CCF Trans. Netw. **3**(1), 4–20 (2020). https://doi.org/10.1007/s42045-020-00031-0
9. Heilmann, F., Fohler, G.: Size-based queuing: an approach to improve bandwidth utilization in TSN networks. ACM SIGBED Rev. **16**(1), 9–14 (2019)
10. Schweissguth, E., Danielis, P., Timmermann, D., Parzyjegla, H., Mühl, G.: ILP-based joint routing and scheduling for time-triggered networks. In: Proceedings of the 25th International Conference on Real-Time Networks and Systems, pp. 8–17 (2017)
11. Gavriluţ, V., Zhao, L., Raagaard, M.L., Pop, P.: AVB-aware routing and scheduling of time-triggered traffic for TSN. IEEE Access **6**, 75229–75243 (2018)
12. Kruk, S.: Practical python AI projects: mathematical models of optimization problems with google or-tools (2018)

Fundamental Analysis of 3D 6G-Localization Using Reconfigurable Intelligent Surface

Yang Chen[1], Yubin Zhao[1](✉), Xiaofan Li[2], and Dunge Liu[3]

[1] School of Microelectronics Science and Technology, Sun Yat-Sen University,
Zhuhai 519082, China
cheny965@mail2.sysu.edu.cn, zhaoyb23@mail.sysu.edu.cn

[2] School of Intelligent System Science and Engineering, Jinan University,
Zhuhai 519070, China
lixiaofan@jnu.edu.cn

[3] State Key Laboratory of Space-Ground Integrated Information Technology,
Space Star Technology Co., Ltd., Beijing 100041, China

Abstract. Reconfigurable intelligent surface (RIS) is a promising technique in the 6G communication system, which effectively improves the wireless propagation channel. Moreover, the RIS also benefits localization performance since it helps avoid the non-line-of-sight channel when there are obstacles. In this paper, we mainly analyze the 3D localization performance of the millimeter-wave (mmWave) system with a given fixed RIS. The Cramér lower bound (CRLB) is derived for our proposed 3D RIS-based wireless propagation channel. We analyze the localization accuracy of time-of-arrival (TOA) and angle-of-arrival (AOA). The results indicate that the RIS-based localization method can significantly improve localization accuracy, and centimeter-level localization can be attained. In addition, the localization based on TOA outperforms that based on the AOA when the number of the Rx and RIS units is fixed.

Keywords: Reconfigurable intelligent surface (RIS) · Millimeter-wave · Wireless localization · Cramér-Rao lower bound (CRLB)

This work was partially supported by National Nature Science Foundation of China (No. 62171484), Zhuhai Fundamental and Application Research (No. ZH22017003210006PWC), Fundamental Research Funds for the Central Universities (No. 21621420), the Open Research Fund of State Key Laboratory of Space-Ground Integrated Information Technology under grant NO. 2018_SGIIT_KFJJ_ST_01.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 255–266, 2022.
https://doi.org/10.1007/978-3-031-19214-2_21

1 Introduction

The 6G not only can provide intelligent communication for all things but also can achieve high-precision positioning due to the excellent angular resolution. The 6G brings a myriad of new opportunities for wireless localization and sensing [1]. Moreover, the 6G is folded increases significantly for wireless localization. The reconfigurable intelligent surface (RIS), which has aroused widespread discussion, is considered as one of the leading enabling techniques of the 6G [2] and is composed of a large number of low-cost passive units that can reconfigure their physical parameters under the control of bias voltage [12,16], without any need for additional baseband processing units, and radio frequency (RF) modules [7]. A RIS unit can operate as an intelligent reflector beyond Snell's law [14] or as a lens with nearly a continuous phase profile [8].

The RIS has the following three advantages for wireless positioning. Firstly, they provide a variable signal propagation channel due to the phase and amplitude of waveform propagation can be controlled in the air [18]. Secondly, they can be deployed flexibly to extend the communication distance [17]. Thirdly, they are effective since a large number of cost-effective passive units are equivalent to large-scale antennas [9,10,19,21]. The RIS is widely used in many practical communication scenarios, such as cell edge communication and passive beamforming, etc., [3,5,10], to improve the signal quality of the receivers. The RIS has been similarly investigated in several studies in the localization literature, e.g., [4,6,13,20]. In [4], localization in the near-field range of a RIS, functioning as a lens, is studied. A single input single output (SISO) 2D localization problem with synchronized signaling and multiple RIS with a uniform linear array (ULA) has been investigated in [20] by deriving the Cramér-Rao lower bound (CRLB) bounds. The CRLB has been derived in [13] for a 2D localization in the presence of the RIS-aided MISO system with a ULA. In [6], He et al. have proposed a joint localization and communication for a 2D wireless system comprising multiple RIS.

However, the fundamental localization analysis for the 6G MIMO system using RIS has not been thoroughly investigated yet. Therefore, this paper mainly analyzes the fundamental localization performance of the millimeter-wave (mmWave) system with a given fixed RIS. First, we present a 3D RIS-based wireless propagation channel. After that, the Fisher information matrix (FIM) and the corresponding CRLB are derived for the MIMO 6G system. Finally, We analyze the localization accuracy of time-of-arrival (TOA) and angle-of-arrival (AOA). The results indicate that the RIS-based localization can improve localization accuracy. Our main contributions are two folds.

- We employ the method of driving CRLB to analyze the estimation performance for the 6G-localization with a given number of RIS units. Such a method provides a general analytical formulation for the 6G-localization estimation based on TOA or AOA. Thus, the RIS-based localization method can significantly improve localization accuracy.
- We perform the CRLB simulation for the mobile station (MS) location. Our results indicate that the TOA-based localization outperforms that based on

the AOA since the excellent multipath resolution. Thus, the 6G system can choose TOA as a localization estimation method instead of AOA, even if the SNR is very low.

2 3D System and Channel Model

2.1 3D System Model

Table 1. Summary of the used notations

Term	Definition
$s_{B_i R_u}$	The Tx signal for the $B_i R_u$-th BS
$y_{M_j R_u}$	The Rx signal for the $M_j R_u$-th MS
$\boldsymbol{\alpha}_B$	The antenna response matrix for the BS
$\boldsymbol{\beta}_M$	The antenna response matrix for the MS
$\boldsymbol{\Phi}$	The RIS signal matrix
$\widetilde{\mathbf{H}}$	The communication channel matrix
$\mathbf{N}$	The white Gaussian noise matrix
$\mathbf{H}$	The propagation gain matrix
N_B	The number of the BS
N_R	The number of the RIS units
N_M	The number of the MS
θ_{R_u}	The elevation AOD of the R_u-th path
φ_{R_u}	The azimuth AOD of the R_u-th path
ϑ_{R_u}	The elevation AOA of the R_u-th path
ϕ_{R_u}	The azimuth AOA of the R_u-th path
$\mathbf{b}$	The position of the BS
$\mathbf{m}$	The position of the MS
$\mathbf{r}$	The positions of the RIS

We consider a RIS-assisted wireless localization system as depicted in Fig. 1, which mainly consists of three components: 1) the base station (BS); 2) the RIS; 3) the MS. In the wireless positioning system, the BS is the transmitter used to generate the transmitted signals. The RIS is used to reflect the incident signal to reconfigure the wireless propagation channel intelligently. The MS with unknown locations is the receiver, such as smart devices, intelligent sensor nodes, and underwater robots.

As a MIMO system, both the BS and MS are equipped with massive antennas. Define N_B, N_R, and N_M as the number for the BS, RIS and MS, respectively. Therefore, the set of the number of the BS, RIS, and MS are defined as

$\mathcal{N}_B = \{1, 2, \cdots, N_B\}$, $\mathcal{N}_R = \{1, 2, \cdots, N_R\}$ and $\mathcal{N}_M = \{1, 2, \cdots, N_M\}$ respectively.

Since the small wavelength of mmWave, they can fit within the compact form. Therefore, they can be viewed as two points, and their positions are denoted as $\mathbf{b} = [b_x, b_y, 0]^T$ and $\mathbf{m} = [m_x, m_y, 0]^T$, respectively. Compared with the BS and MS, the RIS has a much larger size. Thus the RIS needs to be considered separately. The positions of the R_u-th reflecting element are denoted as $\mathbf{r} = [\mathbf{r}_1, \cdots, \mathbf{r}_{R_u} \cdots, \mathbf{r}_{N_R}]$, where $\mathbf{r}_{R_u} = [r_{x_{R_u}}, r_{y_{R_u}}, r_{z_{R_u}}]^T$, in which $u \in \mathcal{N}_R$. The values of $\mathbf{b}$ and $\mathbf{r}$ are assumed to be known, while the value of $\mathbf{m}$ is unknown and requires to be estimated.

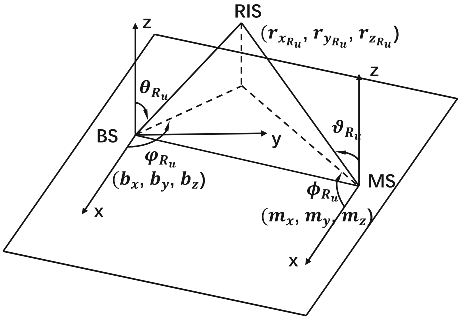


Fig. 1. 3D channel model.

We focus on the system is obstructed line-of-sight, where there exist N_R reflection paths through RIS. The TOA of R_u-th path is denoted as τ_{R_u}. The elevation and azimuth angle-of-departure (AOD) of the R_u-th path are represented as θ_{R_u} and φ_{R_u}. The elevation and azimuth AOA of the R_u-th path are denoted as ϑ_{R_u} and ϕ_{R_u}. Since the positions of BS and RIS are fixed and known, we can obtain the values of θ_{R_u} and φ_{R_u} by means of the geometrical relationship between them. The values of τ_{R_u} and ϕ_{R_u} are unknown and require to be estimated. In addition, the position estimate is equivalent to the azimuth AOA and TOA estimates due to the geometrical relationship. The related notations are summarized in Table 1.

As indicated in Fig. 1, we attain the geometric relationship between the elevation AOA or TOA and position, which is denoted as

$$\phi_{R_u} = -\arctan\left[\frac{|m_y - r_{y_{R_u}}|}{|m_x - r_{x_{R_u}}|}\right], \tag{1}$$

and

$$\tau_{R_u} = \frac{1}{c}(\|\mathbf{m} - \mathbf{r}_{R_u}\|_2 + \||\mathbf{r}_{R_u} - \mathbf{b}\|_2), \tag{2}$$

where c is the light speed.

2.2 Channel Model

Assume that the RIS is composed of N_R small but large spacing auxiliary localization units, whose matrix is $\boldsymbol{\Phi} = \rho \operatorname{diag}\left[e^{j\psi_1}, \cdots, e^{j\psi_{N_R}}\right]^T$, where $e^{j\psi_u}, u \in \mathcal{N}_R$ is an element-wise power operation, and $\psi_u, u \in \mathcal{N}_R$ represents the phase shifts of reflecting unit at RIS.

We ignore the bounce reflections from the ground or other scatterers since such paths get attenuated much more significantly than the paths through RIS. Based on the system model given above, the $N_M \times N_B$ channel matrix is expressed as

$$\widetilde{\mathbf{H}} = \boldsymbol{\beta}_M(\mathbf{H}\boldsymbol{\Phi})\boldsymbol{\alpha}_B^H \tag{3}$$

where the matrices $\boldsymbol{\alpha}_B$ and $\boldsymbol{\beta}_M$ are the array response matrices at BS and MS, the diagonal matrix $\mathbf{H}$ is the propagation gain matrix of N_R paths. The array response matrices $\boldsymbol{\alpha}_B \in \mathbb{C}^{N_B \times N_R}$ and $\boldsymbol{\beta}_M \in \mathbb{C}^{N_M \times N_R}$, which depend on the angular parameters, is defined as

$$\boldsymbol{\alpha}_B = \begin{bmatrix} 1 & \cdots & e^{j(1-1)k\bar{\omega}_{N_R}} \\ \vdots & \ddots & \vdots \\ e^{j(N_{\mathrm{B}}-1)k\bar{\omega}_1} & \ldots & e^{j(N_{\mathrm{B}}-1)k\bar{\omega}_{N_R}} \end{bmatrix} \tag{4}$$

$$\boldsymbol{\beta}_M = \begin{bmatrix} 1 & \cdots & e^{j(1-1)k\breve{\omega}_{N_R}} \\ \vdots & \ddots & \vdots \\ e^{j(N_{\mathrm{M}}-1)k\breve{\omega}_1} & \ldots & e^{j(N_{\mathrm{M}}-1)k\breve{\omega}_{N_R}} \end{bmatrix} \tag{5}$$

where $\bar{\omega}_{R_u} = \sin\theta_{R_u}\cos\varphi_{R_u}$, and $\breve{\omega}_{R_u} = \sin\vartheta_{R_u}\cos\phi_{R_u}, u \in \mathcal{N}_R$, the parameter $k = 2\pi d/\lambda$, where d and λ are the separation between Tx and Rx antennas at BS or MS and the wavelength of transmitted signal, respectively. The diagonal matrix $\mathbf{H} = \operatorname{diag}[\mathbf{h}]$, where the $N_R \times 1$ vector $\mathbf{h} = [h_1, \cdots, h_{N_R}]^{\mathrm{T}}$ represents the propagation gains of N_R reflection paths.

2.3 Received Signal Model

For a 6G system, on the Tx side, the transmitted signal is $\mathbf{S} = \left[s_{B_1}, \cdots, s_{B_i}, \cdots, s_{B_{N_B}}\right]^T \in R^{N_B \times 1}$, where $s_{B_i}(t) = \sum_{n=1}^{N_M} Ae^{j2\pi n f_0(it-\tau_{R_u})}$, then f_0 and A are the carrier frequency and baseband pulse amplitude with pule length T_s, respectively.

On the Rx side, the received signal $\mathbf{Y} = \left[y_{M_1}, \cdots, y_{M_j}, \cdots, y_{M_{N_M}}\right]^T \in \mathbb{C}^{N_{\mathrm{M}} \times 1}$ at the MS is expressed as

$$\mathbf{Y} = \widetilde{\mathbf{H}}\mathbf{S} + \mathbf{N} \tag{6}$$

where matrix $\mathbf{N} \in \mathbb{C}^{N_M \times 1}$ is an additive white Gaussian noise with the elements independently drawn from $\mathcal{CN}\left(0, \sigma^2\right)$, and the transmit power is $P_{BS} = \mathbb{E}\left\{\boldsymbol{S}^{\mathrm{H}}\boldsymbol{S}\right\}$.

3 Cramér-Rao Lower Bound on Position Estimation

The CRLB, which is expressed as the inverse of the FIM, sets the lower bound of the covariance matrix of any unbiased estimate of unknown parameters [18]. CRLB not only presents the lower bound of estimation error but also indicates the correlation between estimation error and location position. Therefore, analyzing CRLB can evaluate the performance of the parameter estimation method.

Based on the 3D communication channel model, the parameters to be estimated can be defined as

$$\boldsymbol{\eta} = [\phi_1, \cdots, \phi_{N_R}, \tau_1, \cdots, \tau_{N_R}, h_1, \cdots, h_{N_R}]^T \tag{7}$$

We denote the unbiased estimate of $\boldsymbol{\eta}$ as $\hat{\boldsymbol{\eta}}$ satisfies the following inequality

$$\mathbb{E}\left[(\hat{\boldsymbol{\eta}} - \boldsymbol{\eta})(\hat{\boldsymbol{\eta}} - \boldsymbol{\eta})^H\right] \succeq \mathbf{J}_{\boldsymbol{\eta}}^{-1} \tag{8}$$

where $\boldsymbol{A} \succeq \boldsymbol{B}$ should be represented as matrix $\boldsymbol{A} - \boldsymbol{B}$ is non-negative. The matrix $\mathbf{J}_\eta$ is the $3N_R \times 3N_R$ FIM, and $[\mathbf{J}_\eta]_{p,p}^{-1}$ is the CRLB for the p-th parameter estimate. The (p, q)-th entry of $\mathbf{J}_\eta$ is determined as

$$[\mathbf{J}_{\boldsymbol{\eta}}]_{p,q} = \mathbb{E}\left[\frac{\partial \ln f(\boldsymbol{Y}; \boldsymbol{\eta})}{\partial \boldsymbol{\eta}_p} \frac{\partial \ln f(\boldsymbol{Y}; \boldsymbol{\eta})}{\partial \boldsymbol{\eta}_q}\right] \tag{9}$$

where $f(\boldsymbol{Y}; \boldsymbol{\eta})$ is the probability distribution function (pdf) of the received signal $\mathbf{Y}$ conditioned on $\boldsymbol{\eta}$, and $\boldsymbol{\eta}_p$ is the p-th entry of $\boldsymbol{\eta}$, then $p, q \in \mathcal{N}_R$. The proof of the identity in Eq. (9) is given in [15]. With Eq. (6), the pdf can be denoted as

$$f(\boldsymbol{Y}; \boldsymbol{\eta}) = \frac{1}{(2\pi)^{\frac{N_M}{2}} \det^{\frac{1}{2}}(\mathbf{C})} e^{-\frac{1}{2}\left[(\mathbf{Y}-\boldsymbol{\mu})^H \mathbf{C}^{-1} (\mathbf{Y}-\boldsymbol{\mu})\right]} \tag{10}$$

where matrix $\boldsymbol{\mu}$ and $\mathbf{C}$ are the mean and variance matrix of the received signal $\mathbf{Y}$ conditioned on $\boldsymbol{\eta}$.

To further attain the FIM $\mathbf{J}_\eta$, we employ the following Lemma:

Lemma 1. *For the received signal* $\mathbf{Y} \in \mathbb{C}^{N_\mathrm{M} \times 1}$ *follows the complex Gaussian distribution* $\mathcal{CN}(\boldsymbol{\mu}, \mathbf{C})$*, the* (p, q)*-th entry of the FIM is expressed as*

$$\begin{aligned}[\mathbf{J}_{\boldsymbol{\eta}}]_{p,q} =& 2\,\mathrm{Re}\left\{\frac{\partial \boldsymbol{\mu}^H}{\partial \boldsymbol{\eta}_p} \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\eta}_q}\right\} \\ &+ \mathrm{tr}\left\{\boldsymbol{C}^{-1} \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{\eta}_p} \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{\eta}_q}\right\}\end{aligned} \tag{11}$$

where symbol Re *and* tr *in the preceding equation indicate real part of a complex matrix and matrix trace, respectively.*

Proof. Please refer [11].

By using the preceding Lemma, since $\boldsymbol{C} = \sigma^2\mathbf{I}$ does not depend on $\boldsymbol{\eta}$, the (p,q)-th entry of $\mathbf{J}_{\boldsymbol{\eta}}$ in Eq. (9) is rewritten as

$$[\mathbf{J}_{\boldsymbol{\eta}}]_{p,q} = \frac{2}{\sigma^2}\,\mathrm{Re}\left\{\frac{\partial\boldsymbol{\mu}^H}{\partial\boldsymbol{\eta}_p}\frac{\partial\boldsymbol{\mu}}{\partial\boldsymbol{\eta}_q}\right\} \tag{12}$$

where the mean vector $\boldsymbol{\mu} = \widetilde{\mathbf{H}}\boldsymbol{S}$.

First, for the azimuth AOA or TOA, we are more interested in the absolute MS position $\boldsymbol{m} = [m_x, m_y, 0]^T$. We use the chain rule to decompose $\mathbf{J}_{\mathbf{m}}$. The position FIM is obtained as

$$\mathbf{J}_{\mathbf{m}} = \mathbf{F}_{\boldsymbol{\eta}\mathbf{m}}\mathbf{J}_{\eta}\mathbf{F}_{\boldsymbol{\eta}\mathbf{m}}^T \tag{13}$$

where $\mathbf{F}_{\boldsymbol{\eta}\mathbf{m}} = \frac{\partial\eta^T}{\partial\mathbf{m}}$ is the operator of firstorder partial derivatives.

Next, by computing the partial derivatives of the azimuth AOA and TOA with respect to the MS position, we obtain the submatrices matrix of the $\mathbf{F}_{\boldsymbol{\eta}\mathbf{m}}$

$$[\mathbf{F}_{\boldsymbol{\phi}\mathbf{m}}]_{R_u} = \frac{\partial\phi_{R_u}}{\partial\boldsymbol{m}} = [\sin\phi_{R_u}, -\cos\phi_{R_u}]^T \tag{14}$$

$$[\mathbf{F}_{\boldsymbol{\tau}\mathbf{m}}]_{R_u} = \frac{\partial\tau_{R_u}}{\partial\boldsymbol{m}} = \frac{1}{c}[\cos\vartheta_{R_u}, \sin\vartheta_{R_u}]^T \tag{15}$$

Because the $2 \times N_R$ submatrix $\mathbf{F}_{\mathbf{hm}}$ is a zero matrix, then based on the multiplication principle of the partitioned matrix, Eq. (13) is simplified as

$$\mathbf{J}_{\boldsymbol{m}} = \frac{2}{\sigma^2}\widetilde{\mathbf{F}}_{\boldsymbol{\eta}\mathbf{m}}\tilde{\mathbf{J}}_{\eta}\widetilde{\mathbf{F}}_{\boldsymbol{\eta}\mathbf{m}}^T \tag{16}$$

where the matrix $\widetilde{\mathbf{F}}_{\boldsymbol{\eta}\mathbf{m}}$ consists of the submatrices $\mathbf{F}_{\boldsymbol{\varphi}\mathbf{m}}$ and $\mathbf{F}_{\boldsymbol{\tau}\mathbf{m}}$, and the matrix $\tilde{\mathbf{J}}_{\eta}$ is the $2N_R \times 2N_R$ submatrix as

$$\tilde{\mathbf{J}}_{\boldsymbol{\eta}} = \begin{bmatrix} \mathbf{J}_{\boldsymbol{\phi}} & \mathbf{J}_{\boldsymbol{\tau\phi}} \\ \mathbf{J}_{\boldsymbol{\phi\tau}} & \mathbf{J}_{\boldsymbol{\tau}} \end{bmatrix} \tag{17}$$

where $\mathbf{J}_{\boldsymbol{\phi\tau}} = \mathbf{J}_{\boldsymbol{\tau\phi}}$.

According to Eq. (9) and Eq. (10), we have

$$[\mathbf{J}_{\boldsymbol{\phi}}]_{R_u} = \frac{1}{3}N_M(N_M-1)(2N_M-1)(k\breve{\omega}_{R_u})^2\gamma_{R_u} \tag{18}$$

$$[\mathbf{J}_{\boldsymbol{\tau}}]_{R_u} = \frac{4}{3}N_M(N_M+1)(2N_M+1)(\pi^2\kappa^2)\gamma_{R_u} \tag{19}$$

where $\gamma_{R_u} = \frac{\int_0^{T_s}\left|\rho h_{R_u}s_{B_i}(t)\right|^2\,\mathrm{d}t}{\sigma^2}$ indicate the signal-to-noise ratio (SNR) of the R_u-th path, and $\kappa^2 = \frac{\int_{-\infty}^{\infty} f^2|S_{B_i}(f)|^2\,\mathrm{d}f}{\int_{-\infty}^{\infty}|S_{B_i}(f)|^2\,\mathrm{d}f}$ represent the squared effective bandwidth of $s_{B_i}(t)$, respectively, with $S_{B_i}(f)$ being the Fourier transform of $s_{B_i}(t)$. The CRLB for the TOA positioning is expressed as

$$\varepsilon_{TOA} = \mathrm{tr}\left[\left(\mathbf{F}_{\boldsymbol{\tau}\mathbf{m}}\mathbf{J}_{\boldsymbol{\tau}}\mathbf{F}_{\boldsymbol{\tau}\mathbf{m}}^T\right)^{-1}\right] = \frac{\mathrm{tr}\left(\mathbf{F}_{\boldsymbol{\tau}\mathbf{m}}\mathbf{J}_{\boldsymbol{\tau}}\mathbf{F}_{\boldsymbol{\tau}\mathbf{m}}^T\right)}{\det\left(\mathbf{F}_{\boldsymbol{\tau}\mathbf{m}}\mathbf{J}_{\boldsymbol{\tau}}\mathbf{F}_{\boldsymbol{\tau}\mathbf{m}}^T\right)} \tag{20}$$

Therefore, we have $\varepsilon_{TOA} \approx \frac{3c^2}{N_R N_M (N_M+1)(2N_M+1)\pi^2 \gamma_{R_u}(4\kappa^2)}$. The CRLB for the AOA positioning is obtained in a similar way, we have $\varepsilon_{AOA} \approx \frac{3}{N_R N_M (N_M-1)(2N_M-1)\gamma_{R_u}(k\breve{\omega}_{R_u})^2}$.

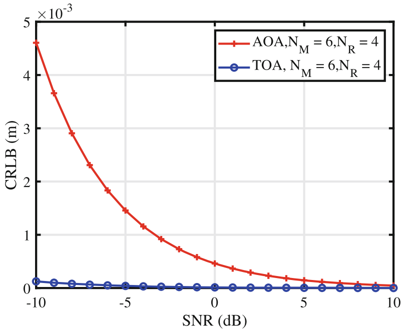


Fig. 2. Simulation comparisons of ε between different positioning approaches.

4 Simulation

4.1 Simulation Setting

We set up the system parameters to be known or have been estimated: the wavelength of mmWave signal is 0.006 m, and the RIS is a uniform planar array in the vertical plane. The effective squared bandwidth κ is 500 MHz. To simplify the simulation, we consider $\gamma_{R_u} = \gamma_0$, which ranges from -10 to 10 dB.

4.2 Different Positioning Approaches

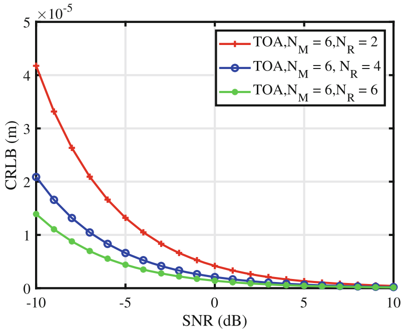


Fig. 3. Simulation comparisons of ε under different N_R using TOA.

Firstly, we evaluate different positioning approaches on a MIMO system. The number of the RIS units and the MS are 4 and 6, respectively. The azimuth and elevation AoA of the R_u-th path are both $\pi/4$, and the separation d is $\lambda/2$. We adapt the SNR from -10 dB to 10 dB. Figure 2 indicates the simulation comparisons of ε between different positioning approaches. It can be clearly observed that the localization based on TOA significantly outperforms that based on the AOA. These comments can be explained to some extent by the detailed expression of ε_{TOA} and ε_{AOA}.

Noted that the N_M has a greater influence on AOA-based localization, while TOA-based localization is affected by the effective bandwidth κ. For Tx signals, the squared effective bandwidth κ is usually greater than 500 MHz, thus $(3c^2)/\left(4\kappa^2\right)$ is quite small. Therefore, AOA-based localization has to largely increase the number of receive antennas to obtain a comparable accuracy with TOA-based localization.

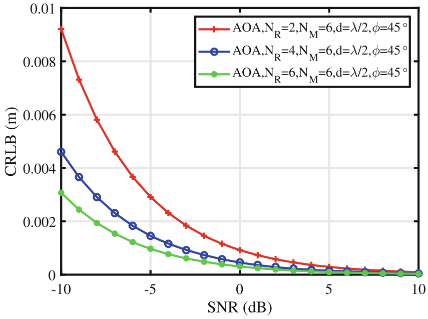


Fig. 4. Simulation comparisons of ε under different N_R using AOA.

4.3 TOA Based Positioning

Next, we evaluate TOA positioning approaches on a MIMO system. The number of the MS is 6. The number of RIS units has been increased from 2 to 6. We adapt the SNR from -10 dB to 10 dB simultaneously. The results are indicated in Fig. 3, which denotes simulation comparisons of ε under different N_R using TOA. In addition, we observe that the CRLB decreases as the N_R and SNR increases, and the CRLB for the MS position estimation can obtain 10^{-3} m, i.e., millimeter-level localization can be attained using the RIS, which meets the requirements for the 6G system.

4.4 AOA Based Positioning

Finally, we evaluate AOA positioning approaches on a MIMO system. The elevation AOA of the R_u-th path is $\pi/4$. We adapt the SNR from -10 dB to 10

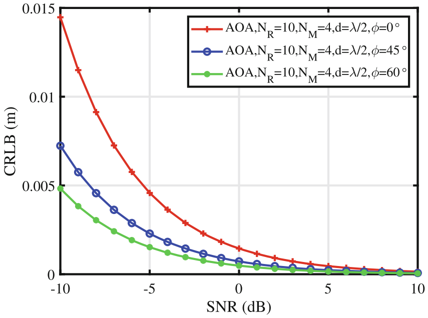


Fig. 5. Simulation comparisons of ε under different azimuth ϕ using AOA.

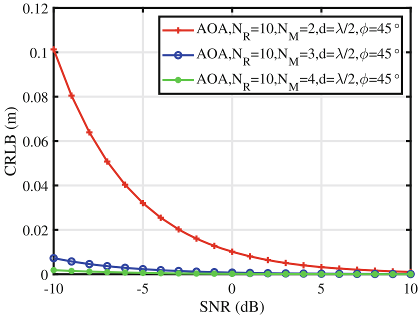


Fig. 6. Simulation comparisons of ε under different N_M using AOA.

dB. Firstly, we evaluate the impact of N_R. In this experiment, The number of RIS units has been increased from 2 to 6. The results indicate that the CRLB is inversely proportional to the SNR, and higher accuracy can be attained using the RIS with a mass of RIS units, as shown in Fig. 4. Secondly, we evaluate the influence of azimuth ϕ. In this experiment, The azimuth change from 0 to $\pi/3$. Figure 5 indicates that the CRLB depends on the direction of the incident wave. When the azimuth $\phi \rightarrow 0$, we have $\sin\phi \rightarrow 0$, the boundary diverges indefinitely. Since $\phi \rightarrow 0$, the visible aperture of the antenna array tends to 0, which will cause the angular resolution to disappear. Thirdly, we evaluate the influence of N_M. In this experiment, The number of MS has been increased from 2 to 3. Figure 6 denotes that the dependence of the CRLB on N_M. In addition, the boundary is proportional to N_M^3. Therefore the CRLB dependence on the number of Rx antennas is greatly strong, with dual dependence. On the one hand, the SNR increases as N_M increases. On the other hand, the antenna aperture also increases with the number of antennas N_M. Finally, we evaluate the influence of the distance d. The distance between the Rx antennas has been increased from $\lambda/2$ to 2λ. Figure 7 represents that the CRLB decreases with the distance d. Higher accurate localization can be obtained when the distance d and azimuth ϕ are large, more MS and more RIS units are utilized.

In summary, we demonstrate that the positioning accuracy can be enhanced using the RIS. It provides ideas for multi-path recognition for the 6G system. In addition, in order to improve positioning accuracy, it is more advantageous to adopt TOA.

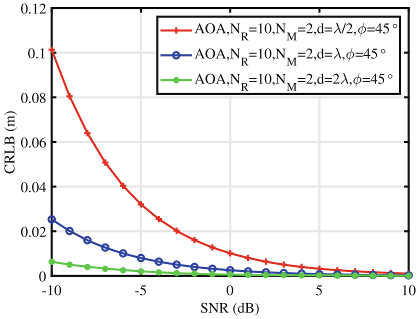


Fig. 7. Simulation comparisons of ε under different d using AOA.

5 Conclusion

In this paper, we take advantage of the mmWave signal technology and introduce RIS into the 6G communication system to make the localization more accurate. The first contribution is to model the 3D RIS-assisted wireless localization system. Secondly, we derive the CRLB for location estimation. Finally, we analyze the TOA and AOA localization accuracy. Extensive simulation results indicate that the RIS-based localization method can significantly improve localization accuracy, and centimeter-level localization can be attained. Moreover, the simulations also denote that the localization based on TOA outperforms that based on the AOA when the number of the Rx and RIS units is fixed.

References

1. Alexandropoulos, G.C., Khayatzadeh, R., Kamoun, M., Ganghua, Y., Debbah, M.: Indoor time reversal wireless communication: experimental results for localization and signal coverage, pp. 7844–7848 (2019)
2. Basar, E., Di Renzo, M., De Rosny, J., Debbah, M., Alouini, M.S., Zhang, R.: Wireless communications through reconfigurable intelligent surfaces. IEEE Access **7**, 116753–116773 (2019)
3. Basar, E., Wen, M., Mesleh, R., Di Renzo, M., Xiao, Y., Haas, H.: Index modulation techniques for next-generation wireless networks. IEEE Access **5**, 16693–16746 (2017)
4. Guidi, F., Dardari, D.: Radio positioning with EM processing of the spherical wavefront. IEEE Trans. Wireless Commun. **20**(6), 3571–3586 (2021)

5. Han, Y., Tang, W., Jin, S., Wen, C.K., Ma, X.: Large intelligent surface-assisted wireless communication exploiting statistical CSI. IEEE Trans. Veh. Technol. **68**(8), 8238–8242 (2019)
6. He, J., Wymeersch, H., Sanguanpuak, T., Silven, O., Juntti, M.: Adaptive beamforming design for mmWave RIS-aided joint localization and communication, pp. 1–6 (2020)
7. Hu, S., Rusek, F., Edfors, O.: Beyond massive MIMO: the potential of positioning with large intelligent surfaces. IEEE Trans. Signal Process. **66**(7), 1761–1774 (2018)
8. Huang, C., et al.: Holographic MIMO surfaces for 6G wireless networks: opportunities, challenges, and trends. IEEE Wirel. Commun. **27**(5), 118–125 (2020)
9. Huang, C., Mo, R., Yuen, C.: Reconfigurable intelligent surface assisted multiuser miso systems exploiting deep reinforcement learning. IEEE J. Sel. Areas Commun. **38**(8), 1839–1850 (2020)
10. Huang, C., Zappone, A., Alexandropoulos, G.C., Debbah, M., Yuen, C.: Reconfigurable intelligent surfaces for energy efficiency in wireless communication. IEEE Trans. Wireless Commun. **18**(8), 4157–4170 (2019)
11. Kay, S.M.: Fundamentals of Statistical Signal Processing: Estimation Theory. PTR Prentice-Hall, Englewood Cliffs (1993)
12. Kumar, P.P., Sreelakshmi, K., Sangeetha, B., Narayan, S.: Metasurface based low profile reconfigurable antenna, pp. 2081–2085 (2017)
13. Ma, T., Xiao, Y., Lei, X., Xiong, W., Ding, Y.: Indoor localization with reconfigurable intelligent surface. IEEE Commun. Lett. **25**(1), 161–165 (2021)
14. Renzo, M., et al.: Smart radio environments empowered by AI reconfigurable metasurfaces: an idea whose time has come. EURASIP J. Wirel. Commun. Netw. **2019**, 129 (2019)
15. Scharf, L.L., Demeure, C.: Statistical Signal Process: Detection, Estimation and Time Series Analysis. Addison-Wesley Publishing Company, Boston (1991)
16. Tang, W., et al.: Wireless communications with programmable metasurface: transceiver design and experimental results. China Commun. **16**(5), 46–61 (2019)
17. Tang, X., Wang, D., Zhang, R., Chu, Z., Han, Z.: Jamming mitigation via aerial reconfigurable intelligent surface: passive beamforming and deployment optimization. IEEE Trans. Veh. Technol. **70**(6), 6232–6237 (2021)
18. Trees, H.L.V.: Optimum array processing (detection, estimation, and modulation theory, part IV). Wiley-Interscience **5**(50), 100 (2002)
19. Wu, Q., Zhang, R.: Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming. IEEE Trans. Wireless Commun. **18**(11), 5394–5409 (2019)
20. Wymeersch, H., Denis, B.: Beyond 5G wireless localization with reconfigurable intelligent surfaces, pp. 1–6 (2020)
21. Zhou, G., Pan, C., Ren, H., Wang, K., Nallanathan, A.: Intelligent reflecting surface aided multigroup multicast miso communication systems. IEEE Trans. Signal Process. **68**, 3236–3251 (2020)

UltrasonicG: Highly Robust Gesture Recognition on Ultrasonic Devices

Zhanjun Hao, Yuejiao Wang(✉), Daiyang Zhang, and Xiaochao Dang

School of Computer Science and Engineering, LanZhou Northwest Normal University, LanZhou, China
yuejiaowangiot@126.com

Abstract. In the current critical situation of novel coronavirus, the use of contactless gesture recognition method can reduce human contact and decrease the probability of virus transmission. In this context, ultrasound-based sensing has been widely concerned for its slow propagation speed, low sampling rate, and easy access to devices. However, limited by the complexity of gestural movements and insufficient training data, the accuracy and robustness of gesture recognition are low. To solve this problem, we propose UltrasonicG, a system for highly robust gesture recognition on ultrasonic devices. The system first converts a single audio signal into a Doppler shift and subsequently extracts the feature values using the Residual Neural Network (ResNet34) and uses Bi-directional Long Short-Term Memory (Bi-LSTM) for gesture recognition. The method effectively improves the accuracy of gesture recognition by combining the information of feature dimension with time dimension. To overcome the challenge of insufficient dataset, we use data extension to expand the dataset. We have conducted extensive experiments and evaluations on UltrasonicG in a variety of real scenarios. The experimental results show that UltrasonicG can recognize 15 kinds of gestures with a recognition distance of 0.5 m. And it has a high accuracy and robustness with a comprehensive recognition rate of 98.8% under different environments and influencing factors.

Keywords: Ultrasonic sensing · Gesture recognition · Data extension · ResNet · Bi-LSTM

1 Introduction

In recent years, the world has suffered from a sudden new coronavirus that has had a widespread impact on people's lives. Especially in recent times, with recurrences in several countries and regions around the world, and the epidemic prevention and control situation remains severe. One of the main ways of COVID-19 transmission is cross-contact [1]**.** And the use of public equipment can indirectly cause human-to-human contact, raising the risk of virus transmission. Therefore, contactless gesture recognition becomes an effective means to reduce the risk of contact infection in epidemic prevention and control.

Previous research work on gesture recognition has been based mainly on RF signals and visual techniques. Implementing gesture recognition based on RF signals requires

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 267–278, 2022.
https://doi.org/10.1007/978-3-031-19214-2_22

specialized hardware such as Universal Software Radio Peripherals (USRP), Frequency Modulated Carrier Wave (FMCW), etc. It leads to high costs and hinders widespread deployment. Visual-based gesture recognition technology has a high recognition rate, but it depends on the brightness and background color of the environment and can easily expose user privacy.

The advent of acoustic technology has overcome the limitations of the above mentioned technologies. Acoustics has been used by many researchers to solve gesture recognition problems due to its slow propagation speed, low sampling rate and easy access to devices. Gao et al. [2] captures gesture movements using lightweight MobileNet by using dual speakers and microphones in smartphones. LLAP [3] is able to realize two-dimensional gesture tracking by measuring the phase change of the received signal. Strata [4] is able to achieve more accurate recognition of gestures by estimating the Channel Impulse Response (CIR) of the reflected signal.

Implementing an acoustic-based fine-grained and highly robust gesture recognition system has two challenges due to the complexity of gesture movements. The first challenge comes from the lack of training data. There are currently few open source datasets based on acoustic gesture recognition, but neural networks need sufficient training data. The second challenge comes from how to recognize fine-grained gesture movements. The acoustic work described above models the entire hand as a single reflection point. It ignores the multipath effects caused by finger movements and doesn't provide sufficient resolution for gesture recognition.

To this end, this paper proposes the implementation of UltrasonicG, a highly robust gesture recognition system on ultrasonic devices. First, the gesture action data are collected using the ultrasonic device ASDP, and the amplitude information is used as the feature value to denoise and smooth, then use Short-Time Fourier Transform (STFT) to extract the Doppler shift of the motion data, and use the ResNet34 to extract the feature value, and finally introduce Bi-LSTM to classify and recognize actions. Especially, we use data extension to address the above challenge of insufficient training data. Data extension relies on our observation and analysis of experimental data. The spectrograms obtained under different gesture speed, hand movement direction and distance to the transceiver influence factors will produce corresponding patterns, and observing the change pattern of the spectrograms under different patterns makes the data after extension cover more actual situations. Specifically, the contributions of this paper are as follows:

- We propose data extension to automatically generate data without user participation to meet the challenge of insufficient training data. In addition, we make the dataset public.
- We improve the accuracy of action recognition by feeding the multiscale semantic features extracted by the ResNet34 into the Bi-LSTM. This method enables the classification network to combine the information of feature dimension with time dimension.
- We conduct rigorous performance evaluations of the proposed approach in diverse real scenarios. The experimental results show that Ultrasonic can reach a recognition accuracy of about 98.8% and has good robustness.

2 Related Work

In this section we present the current research related to gesture recognition in terms of computer vision, Wi-Fi and acoustic waves.

Computer vision-based detection techniques use one or more cameras to capture images of gesture movements to identify the associated actions. Camgoz et al. [5] proposed an end-to-end deep learning approach to recognize continuous sign language gestures from video frames using SubNet and Connectionist Temporal Classification (CTC). Umadevi and Divyasri et al. [6] used a segmentation method based on the skin background minus the hand area to identify five different hand gestures from video capture data. Wi-Fi-based contactless gesture sensing is able to extract Channel State Information (CSI) from Wi-Fi signal data as the sensing medium. Chen et al. [7] used ABLSTM to implement basic action recognition on raw continuous CSI data. WiCatch [8] uses a data fusion-based interference canceling algorithm and support vector machines to accomplish gesture classification. Widar 3.0 [9] achieves cross-scene action recognition by extracting BVP features from CSI to estimate the velocity component of the action and the recognition rate is as high as 92.4%.

With speakers and microphones being widely used in electronic devices such as smartphones, smart speakers and smart watches, acoustic sensing has gained the attention of many researchers. FingerIO [10] is able to accurately track moving objects by transmitting Orthogonal Frequency Division Multiplexing (OFDM) modulated acoustic signals and analyzing the signal variations caused by the moving object. UltraGesture [11] measures the CIR amplitude of the reflected signal to identify the gesture. Wu et al. [12] proposed the EchoWrite system, which is scalable to different forms of devices, does not require a training process, and allows for user information security authentication through text input. Mao et al. [13] proposed a gesture motion tracking system that uses a 4-element microphone array and dual speakers to measure the propagation distance and angle of arrival (AoA) of reflected signals. Wang et al. [14] used a frequency hopping mechanism to mitigate the frequency selective fading problem caused by multipath effects, and achieved a breakthrough in accuracy and robustness with respect to the limitations of acoustic gesture recognition.

Unlike existing solutions mentioned above that either expose the user's privacy or interfere more with the multipath effect, not to mention the requirement for accuracy and real-time in daily-life use. UltrasonicG can solve the above problem and achieve fine-grained and highly robust gesture recognition.

3 System Design

3.1 Overview

The system proposed in this paper is divided into four main stages: data collection, data processing, feature extraction and gesture classification. The system flow is shown in Fig. 1. In the data collection and processing section, two speakers are used as transmitters to send a single 20 kHz audio signal, a microphone is used as a receiver, and the receiving device records and stores the original echo signal. After processing, the original echo signal is converted to a Doppler shift, then filtered using a Butterworth low-pass filter and a short-time Fourier transform, followed by a Gaussian filter to smooth the image. Finally, we use data extension to expand the dataset. In the feature extraction stage, the features of the spectrogram are extracted using the ResNet34 algorithm to generate feature vectors. The gesture classification stage inputs the feature vectors into the Bi-LSTM network for classification and recognition.

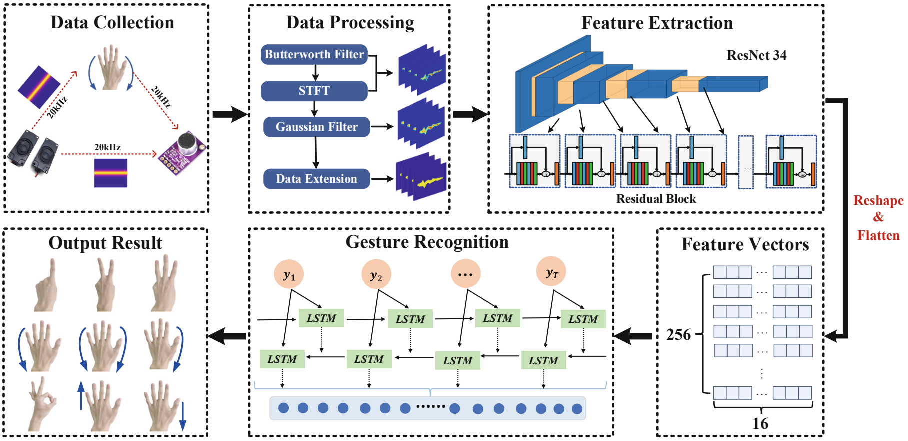


Fig. 1. System flow chart

3.2 Data Collection and Processing

Figure 2 shows a schematic diagram of the 15 gesture types and their corresponding Doppler effects, with the horizontal axis representing time, the vertical axis representing frequency, X → indicating hand motion along the X-axis and double arrows (e.g., X ↔) indicating back-and-forth motion along the X-axis.

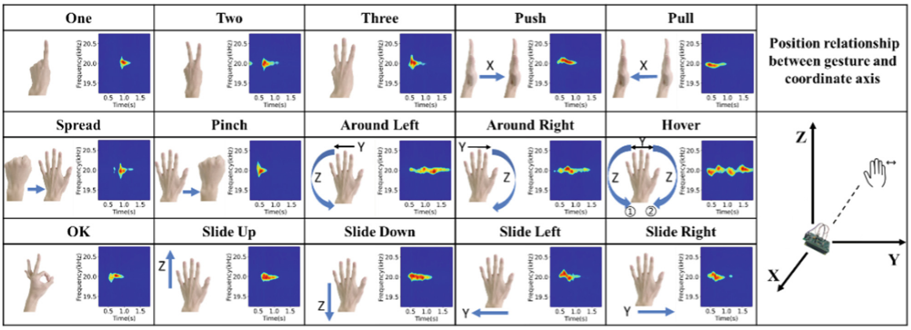


Fig. 2. Schematic diagram of hand gestures and their corresponding doppler patterns

Data Collection. Life noise frequency is usually located at [1000, 4000] Hz, in order to ensure that the signal frequency used in the experiment does not conflict with the life noise frequency, this paper sets the speaker to send a single 20 kHz audio signal.

Data Processing. Figure 3 shows the gesture action data processing process, where the horizontal axis represents time and the vertical axis represents frequency. The interference of background noise is first eliminated using a Butterworth bandpass filter with a frequency of [19000, 21000] Hz, after which the Doppler shift caused by the gesture motion is extracted using STFT, and we estimate the frequency change of the signal after reflection by calculating the Doppler shift to obtain the image shown in Fig. 3(a).

$$\triangle f = f_0 \times \left| 1 - \frac{v_s \pm v_f}{v_s \mp v_f} \right| \tag{1}$$

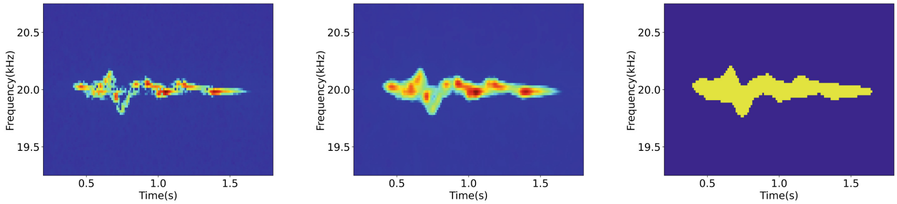
where f_0 is the frequency of the signal sent by the speaker (20 kHz), v_s is the speed of sound (340 m/s), and v_f is the speed of gesture movement (maximum movement speed 4 m/s). So the synthetic frequency shift is about 470.6 Hz and the effective frequency range should be within [19530, 20470] Hz.

To eliminate isolated noise generated by sudden hardware noise, the point where the STFT value changes most dramatically is set as the threshold value, which is set to 0.15. After that we use a Gaussian filter to smooth the image. For two-dimensional images, the following Gaussian functions are used for smoothing:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{2}$$

where x is the distance of the horizontal axis from the origin, y is the distance of the vertical axis from the origin and σ is the standard deviation of the Gaussian distribution. The processed image is shown in Fig. 3(b).

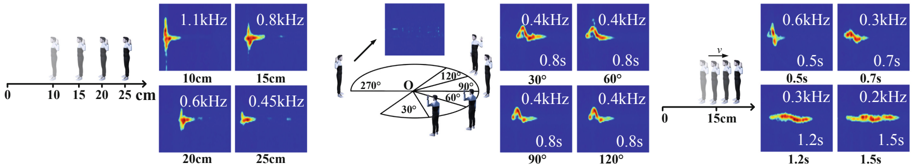
To mitigate the effect of absolute amplitude and to obtain higher quality images, we normalize and binarize the amplitude images, respectively. By the above operation, a clearer spectrum is obtained as shown in Fig. 3(c).



(a) Bandpass Filtering Data (b) Gaussian Smoothing Data (c) Image Enhancement Data

Fig. 3. Gesture data processing

Finally, use data extension to expand the dataset. Data extension relies on our key observations of the experimental data. Based on the initial experimental results, we mainly consider the hand-to-device distance, gesture speed, and angle of arrival as the three factors that may affect the spectrogram in the experiment. The experimental results are shown in Fig. 4, where the horizontal axis represents time and the vertical axis represents frequency fluctuations.



(a) Different Distances (b) Different Angles (c) Different Speeds

Fig. 4. Exploration of data extension

Hand-to-Device Distance. We executed the spread gesture with the hand at 10 cm, 15 cm, 20 cm and 25 cm from the device, and the experimental results are shown in Fig. 4(a). The width of the foreground region in the image represents the time to perform the gesture, and the height represents the range of frequency fluctuations caused by the Doppler shift. Without considering the gesture execution time, we observe the fore-ground area of the image is scaled in the vertical direction. The larger frequency fluctuations, the closer the hand to the device, instead, the farther the hand to the device. Based on the law of frequency fluctuation with distance, the frequency fluctuation range becomes 1.3 times of the original one for every 5 cm decrease in distance. We can randomly scale the vertical foreground area in the range of 0.7–1.3 times to simulate more hand-to-device distance cases.

Different Arrival Angles. We performed the push gesture several times at 15 cm from the device, holding it at 30°, 60°, 90°, 120°and 270° respectively. As shown in Fig. 4(b), because the speaker and microphone are omni-directional, there is no difference in the foreground area of the image when the gesture is executed in front of the device. The Doppler shift caused by the gesture motion is not acquired when the execution gesture angle is 270°.

Different Gesture Speeds. We executed the slide left gesture several times at 15 cm from the device in four speeds from fast to slow. As shown in Fig. 4(c), as the speed gradually slows down, the foreground area of the image is stretched proportionally in the horizontal direction and compressed proportionally in the vertical direction. In practice, a gesture lasts 0.5–2.5 s. Based on the law of frequency fluctuation with time, when the foreground area time is t, we randomly expand the data in the range of t − 0.5 s to t + 0.5 s, and the frequency fluctuation range decreases by 0.05 kHz for each 0.1 s increase in time.

In summary, we find that the angle-of-arrival factor does not require data extension, while the hand-to-device distance and gesture speed require data extension. Data extension allows users to expand the dataset in a short period of time to meet the needs of the classifier model even when a small amount of data is collected.

3.3 Feature Extraction and Gesture Classification

In this paper, we use the ResNet34 [15] to extract features, and its structure is shown in Fig. 5. The ResNet34 model has 34 convolutional layers and includes a total of 16 residual learning units. The spectrogram obtained from data extension is used as the input to ResNet34, ensuring that the input images are all 64 × 64 pixels in size. After each convolutional layer and before the activation function, Batch Normalization is used to accelerate the convergence. The output of the last residual block is reshaped and flattened to obtain the feature vector $y = \left[y_1, y_2, ...y_T\right],$, the total number of feature vectors is $256(T = 256)$, and the length of each feature vector is 16.

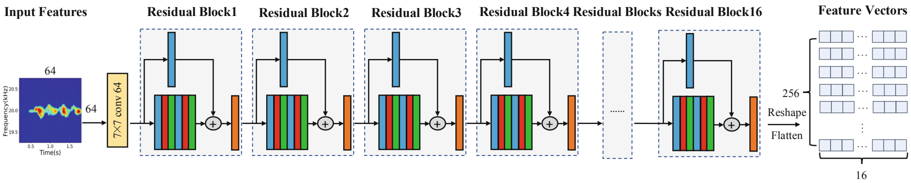


Fig. 5. Structure diagram of ResNet34 network

Bi-LSTM algorithm is used as a gesture recognition classifier, and its structure is shown in Fig. 6. The feature vectors y extracted by the ResNet34 are passed to two LSTM layers and each of which has $T(T = 256)$ LSTM memory cell. To improve the generalization ability of the model, the dropout probability is set to 0.8. These two layers perform sequence feature extraction in opposite directions, and each LSTM memory cell will be computed by three gating units.

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{3}$$

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right) \tag{5}$$

$$h_t = \sigma\left(W_0 \cdot \left[h_{t-1}, x_t\right] + b_0\right) * \tanh(C_t) \tag{6}$$

where h_{t-1} is the output of the spectrogram sequence at the previous moment, x_t is the input of the spectrogram sequence at the current moment, W and b are the weight term and bias term to be learned, respectively. σ is the Sigmoid operation,f_t is the output result of the forgetting gate at moment t, i_t is the information of the spectrogram sequence to be activated at moment t, C_{t-1} and C_t are the state of the spectrogram feature sequence at moment $t - 1$ and moment t, respectively, and h_t is the output result of the output gate at moment t.

After calculation, $H_{forward}$ and $H_{backward}$ can be obtained. Then, we concatenate and flatten $H_{forward}$ and $H_{backward}$ to obtain the vector p. Since the classifier eventually needs to recognize 15 gestures, we design a fully connected neural network with 15 output neurons. Finally, a softmax operation is performed on the output of the fully connected layer to accurately classify the different gestures.

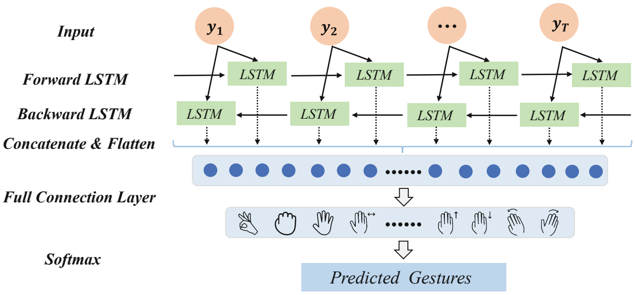
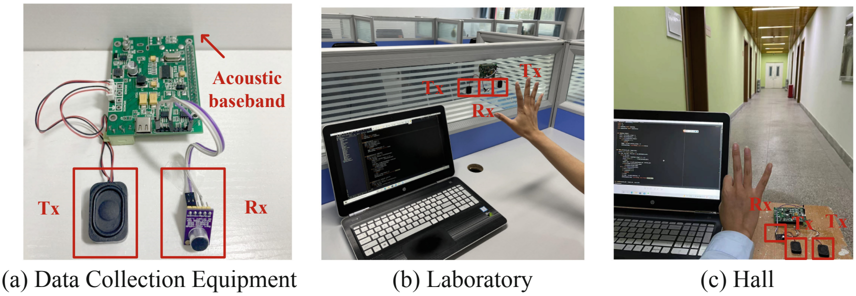


Fig. 6. Structure diagram of Bi-LSTM netwok

4 Experimentation and Evaluation

4.1 Experiment Setup

In this paper, the Acoustic Software Defined Radios Platform (ASDP) [16] equipped with one microphone and two speakers was selected as the data collection tool, and the platform is shown in Fig. 7(a). Set the speaker to emit a 20 kHz continuous single audio signal and set the microphone sampling rate to 44.1 kHz.



(a) Data Collection Equipment (b) Laboratory (c) Hall

Fig. 7. Experimental equipment and experimental environment

In this paper, hand gestures are executed in two scenarios: laboratory and Hall. Laboratories containing regularly distributed tables and chairs with a high impact of multipath effects, which can be used as complex environments. Halls are almost free of obstacles, and the multipath effect has little effect, so they can be used as open environments. The experimental scenario is shown in Fig. 7(b)(c).

We invited 6 male volunteers and 6 female volunteers to perform 15 gestures and collected 450 sets of data under 4 practical influencing factors: distance, speed, noise and angle. After that, 20× data extension is done on the training data, and the amount of extended data is 9000.

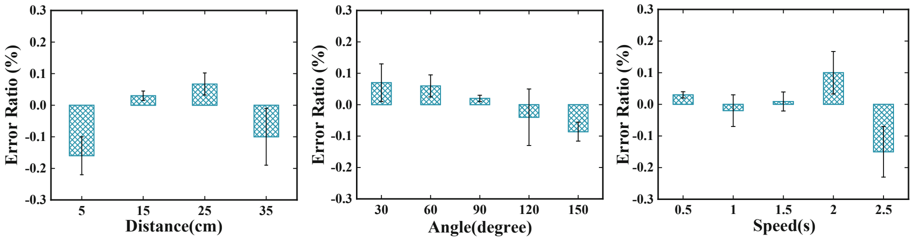
For our experimental evaluation, we explored the error rate of action recognition at different distances from the device, at different angles, and at different gesture speeds, with the error rate defined as follows:

$$error\ ratio = \frac{N_{est} - N_{truth}}{N_{truth}} \times 100\% \tag{7}$$

where N_{est} is the number of predicted gestures of a certain type, and N_{truth} is the number of real recorded gestures of a certain type. Note that the error rate of one test may be negative and the evaluation results are averaged over the test records.

4.2 Experiment Evaluation

Analysis of Different Influencing Factors. In real scenarios, the distance between the user and the transceiver, the angle of arrival and the speed of the gesture will be different each time the user performs the action. In order to find the best distance, the best angle and the best speed when executing the gesture, we explore the different influencing factors from the perspective of error rate, and the experimental results are shown in Fig. 8.

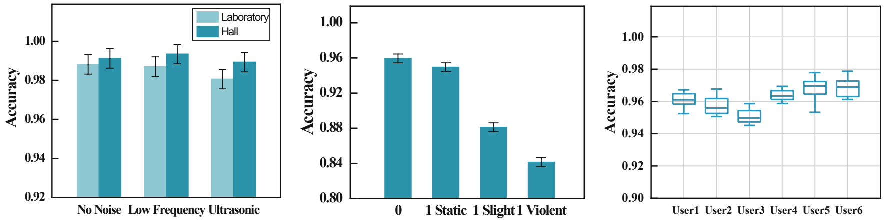


(a) Impact of Distance (b) Impact of Angle (c) Impact of Speed

Fig. 8. Impact on different distance, angle and speed

The effect of distance on the error rate of gesture recognition is shown in Fig. 8(a). It can be seen from the figure that the average gesture recognition error rate reaches 0.02% when the distance from the device is 15 cm. And as the distance increases, the error rate of gesture recognition keeps increasing, which is due to the increased interference of the reflected signal by multipath effect. When the distance decreases to less than 15 cm, the signal reflected by the hand is not completely received by the microphone, and the error rate of gesture recognition rises to 0.17%. The experimental results prove that the system can maintain a good performance within 35 cm range. The effect of angle of arrival on the error rate of gesture recognition is shown in Fig. 8(b). The figure shows that the lowest error rate can be achieved when the experimenter performs the gesture at 90° to the device, which can reach 0.02%. This is because the direction of hand motion is perpendicular to the signal domain and has a greater effect on the signal. When the experimenter is at other angles to the device, the error rate of hand gesture recognition increases slightly, which is due to the fact that the movement of the hand generates a horizontal motion component, resulting in a smaller amplitude of the signal. The effect of speed on the error rate of gesture recognition is shown in Fig. 8(c). As can be seen from the figure, when the gesture duration is 1.5 s, the error rate of gesture recognition is the lowest, which can reach 0.01%. When the gesture duration is too short, it is difficult for the microphone to receive the complete signal, and when the gesture duration is too long, the signal change caused by Doppler shift is weak. The experimental results prove that the gesture speed can achieve good performance in the range of 2.5 s.

Analysis of Robust Performance. In order to test the robustness of the system, this paper designs experiments in terms of three aspects: environmental noise, number of interfering persons and different users. 1) Ask an experimenter to perform 15 gestures in a noise-free, low-frequency noise, 19 kHz ultrasonic noise laboratory and hall environment at a distance of 15 cm from the device, respectively.2) Ask an experimenter to perform the experiment in four situations with 0, 1 static, 1 mild and 1 severe interference. 3) Ask Six experimenters to perform the same gesture, with experimenters 1–3 being female (3 being elderly) and experimenters 4–6 being male (4 being elderly). The experimental results are shown in the Fig. 9.

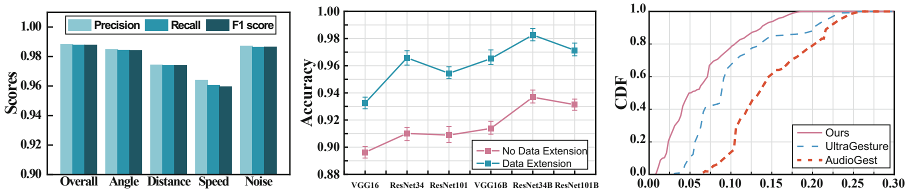


(a) Different Environment and Noise (b) Different Interference States (c) Different Users

Fig. 9. Robustness exploration

The results in Fig. 9(a) show that the gesture recognition accuracy of the system is maintained above 98% in all cases. From the environmental point of view, it can be seen that the gesture recognition results are higher in the hall than in the laboratory, due to the fact that the laboratory contains regularly distributed tables and chairs with a high influence of multipath effects. From the perspective of noise, it can be seen that noise has almost no effect on the experimental results, which verifies that the data processing method proposed in this paper can remove noise well. As can be seen from Fig. 9(b), the accuracy of gesture recognition is 95.9% when the interferer is static. As the interference level increases, the gesture recognition accuracy decreases continuously, and when serious interference occurs, the gesture recognition accuracy still remains above 84%. The experimental results show that the system proposed in this paper has certain anti-interference capability. Figure 9(c) shows that the system has a high recognition rate for the actions performed by all six experimenters. From a gender perspective, it can be seen that the recognition results are slightly higher for males than females, due to the larger palms of males. The poorer recognition results in the elderly are due to the slower execution of gestures and the smaller signal changes caused by Doppler shift in the elderly. The combined results show that the system has high robustness.

Overall System Performance Evaluation. In order to evaluate the overall performance of the system, we conduct a comprehensive investigation in three aspects: different practical factors, data processing methods and data extension. The experimental results are shown in Fig. 10.



(a) Different Influencing Factors (b) Different Models (c) Error Rate

Fig. 10. Overall system performance exploration

Figure 10(a) shows the evaluation results of the system under different practical influence factors. The figure shows that the system exceeds 96% in the three evaluation metrics of precision, recall and F1 score under each influence factor, and the overall system performance reaches 98%. Recognition results are slightly degraded when the gesture is too far from the device or when the gesture speed is too slow. Figure 10(b) shows the gesture recognition accuracy of multiple alternative models with and without data extension, respectively. Among them, VGG16, ResNet34 and ResNet101 are CNN models, and VGG16B, ResNet34B and ResNet101B are "CNN + Bi-LSTM" models. From the model perspective, it can be seen that the best results can be achieved by using a combination of ResNet34 extracted feature values and Bi-LSTM classification recognition. In addition, it is possible to cover more gesture variations by performing 20× data extension. Figure 10(c) shows the Cumulative Distribution Function (CDF) of each of the three methods. As can be seen from the figure, the CDF value of the method proposed in this paper is able to reach 1 as soon as possible, while the value of the CDF of the other two methods rises more slowly. The combined results show that our proposed method can achieve higher accuracy and robustness in gesture recognition.

5 Conclusion

In this paper, we propose UltrasonicG, a system for implementing highly robust gesture recognition on ultrasonic devices. The system can recognize 15 types of gestures with high accuracy and robustness. To achieve fine-grained gesture recognition, ResNet34 is used to extract feature values and Bi-LSTM for gesture classification. To further improve the robustness of the system, data extension is used for different gesture speed and transceiver distance influencing factors. Finally, we constructed a dataset containing gestural behaviors and made it open source. The experimental results show that the system recognizes a distance of 0.5 m with an overall correct rate of 98.8%. In future work, we will further investigate how to recognize two-handed and continuous dynamic gesture behaviors. In addition, we will deploy the scheme to smart devices.

References

1. Fan, C.: Prediction of epidemic spread of the 2019 novel coronavirus driven by spring festival transportation in China: a population-based study. Int. J. Environ. Res. Public Health **17**(5), 1679 (2020)
2. Gao, Y.: EchoWhisper: exploring an Acoustic-based Silent Speech Interface for Smartphone Users. Proc. ACM Interact. Mobile Wear. Ubiq. Technol. **4**(3), 1–27 (2020)
3. Wang, W., Liu, A.X.: Device-free gesture tracking using acoustic signals. In: Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, pp. 82–94 (2016)
4. Yun, S., Chen, Y.C.: Strata: fine-grained acoustic-based device-free tracking. In: Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, pp. 15–28 (2017)
5. Cihan Camgoz, N., Hadfield, S.: SubuNets: end-to-end hand shape and continuous sign language recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3056–3065 (2017)
6. Namdeo, A.: Performance measures for a three-unit compact circuit. Int. J. Adv. Trends Comput. Sci. Eng. **4**, 15107–15115 (2020)
7. Chen, Z.: WiFi CSI based passive human activity recognition using attention based BLSTM. IEEE Trans. Mob. Comput. **18**(11), 2714–2724 (2018)
8. Tian, Z.: WiCatch: A Wi-Fi based hand gesture recognition system. IEEE Access **6**, 16911–16923 (2018)
9. Zheng, Y.-Zhang, Y.: Zero-effort cross-domain gesture recognition with Wi-Fi. In: Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, pp. 313–325 (2019)
10. Nandakumar, R., Iyer, V.: FingeriO: using active sonar for fine-grained finger tracking. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 1515–1525 (2016)
11. Ling, K.: Ultragesture: fine-grained gesture sensing and recognition. IEEE Trans. Mobile Comput. (2020)
12. Zou, Y., Yang, Q.: EchoWrite: an acoustic-based finger input system without training. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 778–787 (2019)
13. Mao, W., He, J.: Cat: high-precision acoustic motion tracking. In: Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, pp. 69–81 (2016)
14. Wang, Y.: Push the limit of acoustic gesture recognition. In: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications (2020)
15. He, K.-Zhang, X.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
16. Cai, C.: Acoustic software defined platform: a versatile sensing and general benchmarking platform. IEEE Trans. Mobile Comput. (Early access), 1–15 (2021)

Joint Federated Learning and Reinforcement Learning for Maritime Ad Hoc Networks: An Integration of Personalized Collaborative Route Planning

Chengzhuo Han[1](✉), Tingting Yang[2,3], and Huapeng Cao[3]

[1] School of Cyber Science and Engineering, Southeast University, Nanjing, China
hcz_dmu@163.com
[2] Peng Cheng Laboratory, Shenzhen, China
[3] National Institute of Defense Technology Innovation, Academy of Military Science China, Beijing, China

Abstract. Maritime Ad hoc networks are a type of decentralised wireless network with rapid networking and multi-hop routing, which are independent of fixed base stations. Recently, Ad hoc networks have started to play an increasingly important role in military command, emergency rescue, disaster relief, temporary meetings, and other occasions. However, as the network topology changes rapidly and the node energy and network bandwidth are limited, discovering and maintaining reliable transmission paths have become a highly topical challenge. In order to solve the problem that distributed routing planning of large-scale Ad hoc networks cannot adapt dynamic changes in network topology, and considering the differences of network nodes, this paper proposes federated reinforcement learning to improve the efficiency of distributed routing planning through the joint learning of similar nodes. Different network nodes have different routing policies, but the routing tables of neighboring nodes are very similar. Therefore, our federated reinforcement algorithm learns nodes with similar routing policies. In this study, a communication system simulation software is specially designed to evaluate the performance of the proposed algorithm.

Keywords: Federated reinforcement learning · Routing planning · Maritime ad hoc network

1 Introduction

Ad Hoc networks are a distinct type of wireless communication network. And Ad Hoc networks has a certain flexibility in the networking process and a reasonably strong ability to adapt to the environment relatively fast. Within a limited area,

Supported by organization 2020YFB1806800.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 279–290, 2022.
https://doi.org/10.1007/978-3-031-19214-2_23

more mobile conditions can be provided to improve the working environment for the operation of mobile communication equipment and meet specific work needs. Ad Hoc networks can also be widely used to provide wireless network support in disaster rescue, remote area development, national defence [11], campus teaching [13] and maritime communications. In wireless maritime Ad Hoc networks, network communication depends on the cooperation between vessels and information forwarding between vessels [2]. As vessels move, the network topology changes dynamically. In wireless self-organising networks, all vessels have equal status and virtually the same complexity. Two vessels that are far away and cannot communicate directly can forward control and data messages via multi-hop relay to complete the communication process. Wireless maritime Ad Hoc networks have enormous potential, which can be better applied in various communication fields.

The deployment of multi-hop relay and forwarding has broad future application prospects, particularly in deep-sea areas where there are few users as it can save deployment costs and makes data transmission between users more flexible. However, many problems related to the reliability of multi-hop relay transmission still need to be solved to ensure the reliability of service transmission, especially how to avoid packet congestion in the network. To this end, some recent works have proposed various solutions [5].

Under the new situation, Ad Hoc network communication can be regarded as a layered control network system composed of multiple agents, which adopts edge computing and relies on the distributed parallel mode among intelligent groups to share information and make collaborative decisions, and finally completes the communication task [8]. At the same time, edge computing reduces the network communication load and improves the system operation efficiency through independent decision-making, key information sharing and task collaboration. In the process of multi-agent execution, cooperative and efficient routing is crucial to improve network performance. This problem is called multi-agent communication planning. Designed to generate good communication routes that guide packets from the source node to the specified destination node.

Recently, many scholars have solved large-scale problems by assigning global control to local agents, which is a significant improvement over centralized reinforcement learning [3]. Unfortunately, in the case of limited communication, each agent is only partially observable of the environment, so it is easy to fall into local optimality. However, for large collaborative communication problems, the centralized RL approach is usually not feasible because: 1) Collecting all the maritime observations in the network to form a global state, which in practice causes high latency; 2) The joint action space of each agent grows exponentially with the increase of the number of agents. Therefore, it is more effective and reasonable to make the large-scale cooperative communication as a cooperative multi-agent decision-making system, that is, each agent controls by local observation.

Distributed wireless maritime Ad Hoc networks use distributed scheduling [9], where nodes share local observations to avoid congestion during message transmission. In distributed networks, nodes only need to maintain and forward the information of neighbour nodes to complete resource scheduling, therefore reducing frequent signalling forwarding between nodes, and greatly reducing overheads compared with centralised networks [6]. Therefore, the in-depth study of distributed wireless multi-hop maritime Ad Hoc networks is of great significance to the development and future application of wireless communication networks.

We treated each node of the maritime Ad Hoc network as an agent and transformed the routing planning problem into a multi-agent communication problem. This paper combines reinforcement and federated learning and proposes that the resulting combined federated reinforcement learning should be combined to solve the above issues. In reinforcement learning to learning as a testing evaluation process, the agent chooses an environment action and the environment, after accepting the action state change, simultaneously produces a strengthening feedback signal (award or punish) to the agent.

Federated learning stores the data of each node locally so the federated system can establish a virtual common model without violating data privacy laws and regulations by exchanging encrypted parameters [7,12]. In this paper, the actions selected in federated reinforcement learning (FRL) not only affect the current node reinforcement value, but also affect the neighbouring states and final reinforcement value. This virtual model is in effect a combined optimal model; however, when creating virtual models, the data itself does not move, nor does it compromise privacy or affect data compliance. In this way, constructed models achieve adjacent region goals in their respective regions. The main contributions of this paper are as follows.

1. **Modeling and Formulation:** We formulate the distributed joint routing problem under maritime and network as markov decision process. For distributed decision making, we aim to reduce the total cost of communication computation while considering the impact of other agents' decision results on the current agent. In addition, we hope that the algorithm can take into account the similarity and difference between nodes.
2. **Algorithm Design:** Joint reinforcement learning proposed by us can improve the efficiency of distributed routing planning through joint learning of similar nodes. Considering the differences of network nodes, it solves the problem that distributed routing planning of large-scale Ad hoc networks cannot adapt to the dynamic changes of network topology.
3. **Experimental Verification and Evaluation:** We performed extensive simulations to evaluate the FRL algorithm. The simulation results not only verify the theoretical tradeoff of FRL, but also show that the FRL algorithm can effectively reduce the total cost of the system and improve the level of algorithm personalization.

The remainder of this paper is organized as follows. A typical mobile maritime Ad Hoc network model is given in Sect. 2 and problem formulation is presented in Sect. 2.4. FRL algorithm is proposed, as well as its advantages in processing heterogeneous data are demonstrated in Section in Sect. 3. In Sect. 4, Simulation results of packet routing planning demonstrate the superiority of the proposed method. We conclude this paper with future work in Sect. 5.

2 Problem Formulation

Maritime Ad Hoc networks receive signals wirelessly. Information can be forwarded to other nodes beyond the wireless transmission range of its own node, that is, any network topology can be formed through wireless connection. It is also a self-organising, infrastructure-free wireless network.

2.1 The Maritime Ad Hoc Network Model

A typical mobile maritime Ad Hoc network model is shown in Fig. 1. In this model, every node in the network is mobile, there is no fixed infrastructure, and the status of nodes is equal. Each node (mobile terminal) is responsible for forwarding packets, finding routes, and maintaining paths. A node faces both a user and a device. Due to the wireless coverage of nodes, fixed object blocking and other reasons, communication between nodes in maritime Ad Hoc networks is generally multi-hop. As shown in Fig. 1, nodes A and I cannot communicate directly, but can communicate through the path A-B-D-F-I.

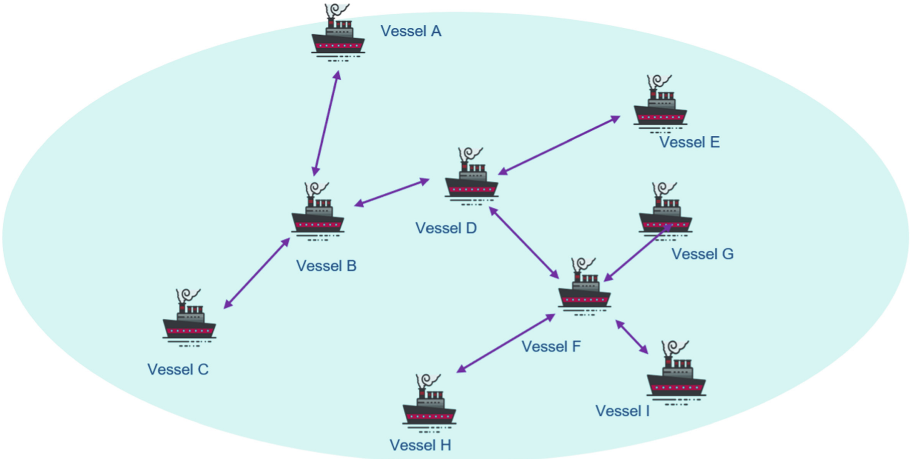


Fig. 1. An illustration of Ad Hoc network

2.2 Data Packet

Network data is transmitted in packets, and each packet has a sending node, destination node, and a current node [10]. We use the arrival time and arrival rate of packets to evaluate routing decisions. If the packet arrives at the destination node before the specified time or exceeds the specified time, the current packet will be deleted, and new packets will be injected into the network. The data packets $\mathcal{P} = \{P_1, \cdots, P_n\}$ can be transmitted on nodes $\mathcal{J} = \{J_1, \cdots, J_m\}$. The packet has parameters $i, j, c \in \{1, \cdots, m\}$, where i represents the sending node, j represents the receiving node, and c represents the current node. Nodes include sending queues, receiving queues, sending power, growth rate, and other attributes. The growth rate is expressed as $\lambda \in \{0, 1\}$, it represents the change of the number of packets on nodes.

2.3 Optimization Objectives

The goal is to minimise the total time delay for transferring data depending on the network state. How to select the route, i.e., which node is the next packet hop to different nodes, can be summarised as a mathematical agent action selection problem. In this interpretation, the node plays the role of an agent, the packet route can be represented by the node action, and the channel quality can be expressed as the edge weight.

We transformed the maritime Ad Hoc network packet routing problem into a multi-agent behaviour selection problem. Corresponding to the multi-agent approach, we use $s \in S$ to represent the state set of adjacent nodes, s represents a specific state, $a \in A$ represents a limited action set, and a represents a specific action. Let $T(S, a, S') \sim P_r(S, a, S')$ be the agent transition model which predicts the next state s' based on the current state S and action a, where the P_r represents the probability of taking action a from s to s'; $R(s, a) = E[R_{t+1} \,|s, a]$ be an immediate reward for an action taken by an agent.

2.4 Problem Formulation

In this section, we propose a formula for the time delay minimisation problem based on reinforcement learning. A certain agent behavioural strategy leads to a positive reward in the environment, and then the tendency of the agent to enact this behavioural strategy in the future will be strengthened [4]. The agent's goal is to discover the optimal strategy in each discrete state to maximise the desired discount reward. We assume that the source domain is $U_A = \{(x_i^A, y_i^A)\}_{i=1}^{M_A}$, and the target domain is $U_B = \{(x_i^B, y_i^B)\}_{i=1}^{M_A}$, D_A and D_B are the hidden special invariants between the source domain and the target domain respectively. We define the classification function of the target domain as:

$$\psi(d_i^A) = \frac{1}{L_A} \sum_{j}^{L_B} y_i^B d_i^B (d_i^A)' = \Phi^B \Omega(d_i^A) \tag{1}$$

The objective function is shown as follows:

$$\arg \min_{\Theta^A, \Theta^B} L_1 = \sum_{i}^{M_c} l_1(y_i^B, \Psi(d_i^A)) \tag{2}$$

$$\arg \min_{\Theta^A, \Theta^B} L_2 = \sum_{i}^{M_{AB}} l_2(d_i^B, d_i^B) \tag{3}$$

The overall objective function is shown as follows:

$$\arg\min_{\Theta^A,\Theta^B} L = L_1 + \gamma L_2 + \frac{\lambda}{2}(\left\|\Theta^A\right\|^2 + \left\|\Theta^B\right\|^2) \tag{4}$$

3 Proposed Algorithms

To achieve efficient route allocation with lower time delays, isolated routing problems are transformed into multi-agent cooperative optimisation problems. We propose a federated reinforcement learning algorithm, which attaches a federated learning mechanism with similar nodes to reinforcement learning.

3.1 Motivation for Algorithm

In order to minimise the total packet forwarding process time, i.e., the waiting time plus transfer time, it is necessary to make optimal routing decisions based on the observations of surrounding nodes. Considering the policy similarity of neighbouring nodes, we used federated reinforcement learning to schedule the next hop packet selection.

The traditional centralized routing decision algorithm is not suitable for this scenario, especially when the number of packets is large. Another scenario is that centralized dispatching can lead to significant wait times when the packet is in an area where communication is poor. Based on the above problems, we consider to use a distributed routing decision algorithm. Meanwhile, since this problem has many influencing factors and is entangled with each other, it is not convenient to solve it in an analytical way, so we use the method of federated reinforcement learning to solve it. Intelligent routing algorithm based on reinforcement learning is able to handle higher dimensions of state characteristic information network, adaptive to different application scenarios and changes in the network environment, the reinforcement learning model and gives the intelligent routing algorithm not only focus on the current routing effect, more predictable future network status changes, and in advance to avoid network congestion what might happen in the future.

3.2 The Learning Common Policy Features of Similar Nodes

In an maritime Ad Hoc network, similar nodes have similar data and routing policies. They are expected to improve the inference accuracy of the model through joint learning. We cannot just apply federated learning to both sides of the data because the routing policies of different nodes are different. Both parties establish a reinforcement learning routing decision model, which have been recognised by their users in data acquisition. The problem is then how to establish high-quality models at each terminal. Due to incomplete or insufficient data, the reinforcement learning model at each end may not be established or lacks the ideal effect. Federated reinforcement learning can solve this problem by ensuring that the data of each node does not go out locally, allowing the federated system to optimise the learning model of all parties through an encrypted parameter exchange. However, when creating virtual models, the data itself does not move, nor does it compromise privacy or affect data compliance. Consequently, the constructed models serve only local goals within their respective regions.

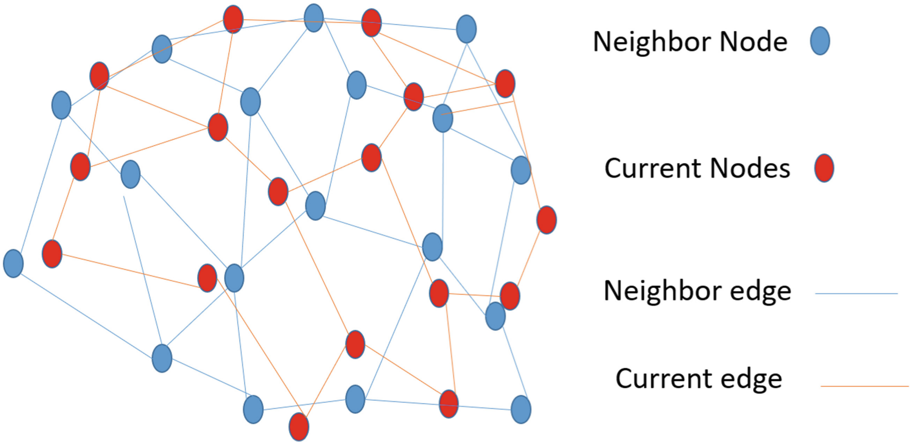


Fig. 2. Node association learning

Partition neighbor path planning based on federated learning focuses on how to map the data of neighbor nodes and current nodes from the original feature space to the new feature space. In this way, the data distribution of the base neighbor node is roughly the same as that of the current node, so that the labeled data samples of the base neighbor can be better used for classification training in the new space, and finally the data of the current node can be classified. To this end, we carry out feature mapping of nodes with close distance, so that neighbor nodes can be used to guide the model parameters of joint nodes with the trained model. Of course, there should be some structural similarity between the topology diagram of neighbor nodes and the current node. As shown in Fig. 2, we first train the neural network according to the red node data, and then take the trained neural network as the alternative network of the actual node. When new nodes join, or the data packet transmission rule of the current network changes, for example, the blue node and red topology are updated online by using federated learning method.

The reinforcement signal provided by the environment in federated reinforcement learning is an evaluation (usually a scalar signal) of the action generated by the agent, rather than telling the agent how to generate the correct action. Since the external environment provides little information, the agent must learn with similar nodes. Therefore, agents gain knowledge in an action-by-action evaluation environment and improve action plans to adapt to the environment. The aim of the reinforcement learning system is to dynamically adjust the parameters to achieve the maximum reinforcement signal. As the reinforcement signal R and the action a generated by the agent do not have a clear functional description, the gradient information R/a cannot be obtained. Therefore, in the reinforcement learning system, a random unit is needed. With this random unit, the agent will search in the possible action space and find the correct action.

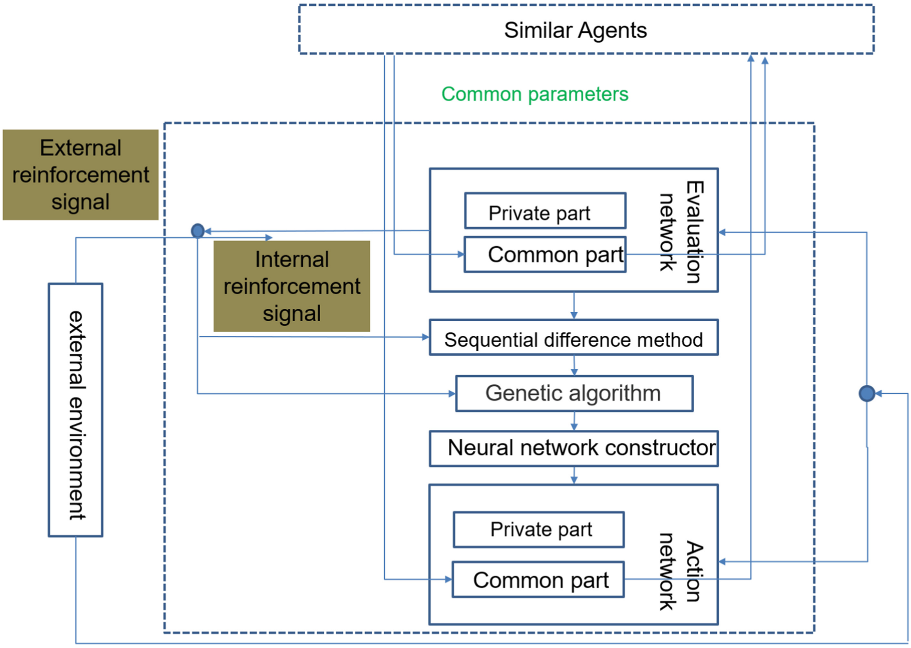


Fig. 3. An illustration of association learning

3.3 Cooperative Scheduling Mechanism Based on Transmission Task Completion

In reinforcement learning, the target of an agent is formally represented as a special signal, called reward, which is transmitted to the agent through the environment. At each time, reward is a single scalar value. Informally, an agent's goal is to maximize the total reward it receives. This means that it's not the immediate rewards that need to be maximized, but the cumulative rewards that need to be maximized over time. The use of reward signals to formalize goals is one of the most distinctive features of reinforcement learning.

The multi-agent path planning algorithm designed in this paper introduces the design reward of transmission task completion to carry out cooperative optimization under the framework of reinforcement learning, as demonstrated in Fig. 3. The principle of cooperative optimization algorithm is to decompose a complex objective function into simple sub-objective functions, and then carry out cooperative optimization of these sub-objective functions. Specifically, collaborative optimization is to optimize each sub-objective function while considering the results of other sub-objective functions, so that the optimization results among sub-objective functions can be consistent. The consistency of optimization results means that the values of each variable can be consistent in the optimization results of each sub-objective function.

The completion degree of this task represents the completion degree of transmission, and the feedback of task execution takes the difference between decision-making route and baseline route as reference. Effect prediction action coordination is mainly responsible for interaction eigenvalues of interested agents within the communication range. The information exchange of task completion is helpful for Agent coordination and strategy formulation in real scenes, and the interactive environment map information is helpful for a single Agent to execute decisions and avoid falling into local optimal solutions.

In this architecture, target behavior is learned from downstream task-specific rewards without any communication oversight. However, complex real-world tasks may need to take into account the interaction of agents after they complete their actions, such as the occurrence of congestion. Therefore, this capability needs to be enhanced by using a multi-round communication method, through which agents coordinate before taking action on the environment. First of all, each agent wants to transmit its own expected action and other agents accept the expected action of other agents at the same time. Then, according to the expected action of other agents, it changes its own action through the expected return and makes the real action. The agent then interacts with the real action environment. The state transition function of the decision is given by:

$$\begin{aligned} p({s_n}', a'|s_{\,n}, a) = Pr(s_{n+1,t} = {s_n}', A_{n+1,t} \\ = a', R_{n+1,t} = a'|s_{\,n,t} = s_n, A_{n,t} = a) \end{aligned} \tag{5}$$

$$p({s_n}', r|s_{\,n}, a) = Pr(s_{t+1,t} = s, R_{t+1,t} = a'|s_{\,t} = s, A_t = a) \tag{6}$$

3.4 Common Network Parameter Aggregation Methods

Each neural network is composed of two modules, namely a private network module and a common network module. In a private network, the federated reinforcement learning algorithm allows it to retain the private features. With the adjacent nodes' features from the common network, the action network output nodes can effectively complete a random search and greatly improve the possibility of selecting suitable actions. Furthermore, the entire action network can be trained online. With auxiliary network environment modelling, evaluation of networks based on the current status and external reinforcement signal simulation environment is used to predict a scalar value. This allows one step, and multi-step, prediction by the action network current actions to strengthen the signal applied to the environment, advance to the relevant action network to provide the candidate actions of intensive signals, and provide more information on rewards and punishments (internal reinforcement signal) [1]. This reduces uncertainty and speeds up learning.

The network operation is divided into two parts: reinforcement feedback calculation and joint parameter calculation. In reinforcement feedback calculation, the time-series differential prediction method (TD) and back-propagation algorithm (BP) are used to learn the evaluation network whilst genetic operation of the mobile network is conducted, and the internal reinforcement signal is used as the mobile network fitness function. Joint parameter calculation determines the weighted average of the parameters of similar nodes so that they can learn from each other. The private network provides more effective internal reinforcement signals to the mobile network, compelling it to produce more appropriate actions. The common network signals enable both the mobile and evaluation networks to learn together with similar nodes, thus greatly accelerating the learning of the two networks.

4 Performance Evaluation

Experimental Setup. The connections between nodes represent specific channels. When multiple data packets are transmitted on the network, they become congested at important nodes, which seriously affects the transmission capability of the entire system. We used federated reinforcement learning to make routing decisions and plan the routing choices of each packet at different nodes.

Simulation Results and Analysis. To simplify the simulation, we assume that the order of packets in the transmission queue does not change. Therefore, if the current packet is blocked, all subsequent packets will be blocked. To ensure that the total number of packets in the network will not exceed the upper limit, when the number of packets reaches the upper limit, one packet will be generated for every delivered packet. The packet generation rule $p^{i,j,k}$ is as follows:

$$p^{i,j,k} = p^{i,i,k} \; when \; p^{i,j,k} = p^{i,k,k} \tag{7}$$

$$i, k = random(0, n) \tag{8}$$

In the simulation we adopted this method to solve the maritime Ad Hoc network routing decision problem. To demonstrate the advantages of the FRL method, we chose to use the shortest path algorithm and Q-learning method for the simulation. The shortest path algorithm is a commonly used algorithm in the field of routing planning. The shortest path problem is a classical algorithm problem in graph theory, which aims to find the shortest path between two nodes in a graph. The learning algorithm allows the system to select the optimal action set by using the experienced action sequence in the Markov environment.

In Fig. 4, we depict the average delivery time versus the number of packets. Average delivery time is the time it takes for a packet to travel from its source to its destination. The number of packets was gradually increased from 500 to 5000, to study the effect of packet density. The trend of the points in the figure shows

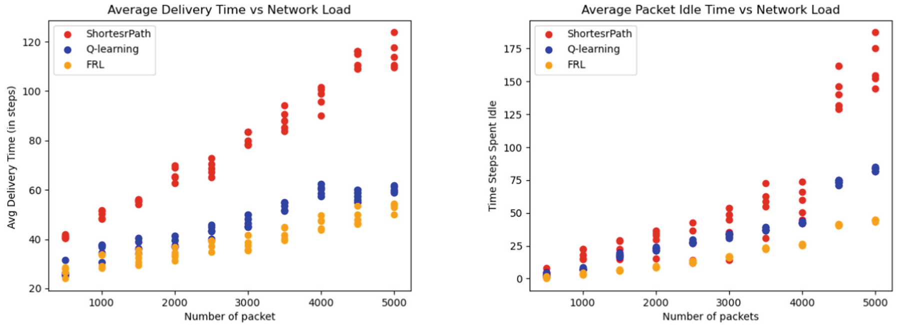


Fig. 4. Simulation results

that the average delivery time increases with packet density. The FRL algorithm has a slightly better performance than the Q-learning algorithm and is clearly better than the shortest path algorithm, thus reflecting the superiority of the algorithm. The relationship between the number of packets and the average packet idle time is shown in the Fig. 4. It can be seen that the FRL algorithm performs better in terms of average packet idle time. Therefore, nodes using the FRL algorithm have superior scheduling ability and avoid long idle packet times.

Through simulation, it was verified that the FRL algorithm can better solve packet congestion, ensure the speed of network transmission and make full use of node performance to avoid long packet idle times.

5 Conclusion

This paper investigated the distributed routing planning problem in maritime Ad Hoc networks with rapid topology changes and limited network bandwidth. With the aim of maximising throughput, the problem of transferring data efficiently was transformed into a congestion avoidance problem. Considering the differences in network nodes, the FRL is proposed to improve the efficiency of distributed routing planning through joint learning of similar nodes. Based on the dynamic data of the dedicated communication simulation system, the simulation results verify the performance of our method. In future work, we will study the application of FRL in private networks.

Acknowledgements. Key Projects of the National Defense Foundation Strengthening Plan under Grant 2020-JCJQ-ZD-020-05, National Defense Technology Innovation Research Institute under Grant 2020KCYWXGK4098, Talent Program of Liaoning Province under Grant XLYC1807149, National Key R&D Program of China under Grant 2020YFB1806800.

References

1. Chen, X., Yuan, Y., Lu, L., Yang, J.: A multidimensional trust evaluation framework for online social networks based on machine learning. IEEE Access **7**, 175499–175513 (2019)
2. Entezari-Maleki, R., Gharib, M., Rezaei, S., Trivedi, K.S., Movaghar, A.: Modeling and evaluation of multi-hop wireless networks using SRNS. IEEE Trans. Netw. Sci. Eng. **8**(1), 662–679 (2021)
3. Hanawal, M.K., Hayel, Y., Zhu, Q.: Effective utilization of licensed and unlicensed spectrum in large scale ad hoc networks. IEEE Trans. Cogn. Commun. Netw. **6**(2), 618–630 (2020)
4. Hwang, K.S., Jiang, W.C., Chen, Y.J., Hwang, I.: Model learning for multistep backward prediction in dyna-q learning. IEEE Trans. Syst. Man Cybern. Syst. **48**(9), 1470–1481 (2018)
5. Kim, B.S., Kim, K.I., Roh, B., Choi, H.: Hierarchical routing for unmanned aerial vehicle relayed tactical ad hoc networks. In: Proceedings of International Conference on Mobile Ad Hoc and Sensor Systems, pp. 153–154 (2018)
6. Liu, J., Guo, S., Shi, Y., Feng, L., Wang, C.: Decentralized caching framework toward edge network based on blockchain. IEEE Internet Things J. **7**(9), 9158–9174 (2020)
7. Mowla, N.I., Tran, N.H., Doh, I., Chae, K.: Federated learning-based cognitive detection of jamming attack in flying ad-hoc network. IEEE Access **8**, 4338–4350 (2020)
8. Naseer Qureshi, K., Bashir, F., Iqbal, S.: Cloud computing model for vehicular ad hoc networks. In: Proceedings of IEEE International Conference on Cloud Networking, pp. 1–3 (2018)
9. Peng, J., Li, X., Li, X.: Research on election interval of distributed wireless ad hoc networks. IEEE Access **8**, 110164–110171 (2020)
10. Ramli, N.I.S., Hisham, S.I., Ismail, N.S.N., Ramalingam, M.: Performance comparison between AODV and DSR in mobile ad-hoc network (MANET). In: Proceedings of IEEE ICSECS-ICOCSIM, pp. 217–221 (2021)
11. Rukaiya, Khan, S.A.: Self-forming multiple sub-nets based protocol for tactical networks consisting of SDRS. IEEE Access **8**, 88042–88059 (2020)
12. Sattler, F., Wiedemann, S., Mller, K.R., Samek, W.: Robust and communication-efficient federated learning from non-I.I.D. data. IEEE Trans. Neural Netw. Learn. Syst. **31**(9), 3400–3413 (2020)
13. Shi, Y., Li, W., Zeng, W.: A study on interaction of college English classroom in the mobile internet environment. In: Proceedings of IEEE IWCMC, pp. 1766–1769 (2021)

LF-DWNet: Robust Depth Estimation Network for Light Field with Disparity Warping

Yuxin Zhao[1], Zhenglong Cui[1], Rongshan Chen[1], Da Yang[1],
and Hao Sheng[1,2,3](✉)

[1] State Key Laboratory of Virtual Reality Technology and Systems,
School of Computer Science and Engineering, Beihang University, Beijing 100191,
People's Republic of China
{zhaoyuxin,zhenglong.cui,rongshan,da.yang,shenghao}@buaa.edu.cn

[2] Beihang Hangzhou Innovation Institute Yuhang, Xixi Octagon City,
Yuhang District, Hangzhou 310023, People's Republic of China

[3] Faculty of Applied Sciences, Macao Polytechnic University, Macao SAR 999078,
People's Republic of China

Abstract. Light field (LF) images can store multi-view geometry characteristics about the observed scene, which can be helpful in depth estimation. Depth estimation has attracted much attention in recent years for its widely use in the computer vision tasks. Many approaches have been proposed to estimate the depth of LF images, including conventional methods and learning-based methods. But most of them are hard to apply to different complex situations. We propose a robust depth estimation network for LF images with disparity warping (LF-DWNet), which is robust in large disparity pixels, occlusions, and noise areas. To reduce the effect of large disparity pixels, we introduce the disparity warping processing on EPI. To extract the depth feature from warped EPI and reduce the effect of occlusions and noise areas, we design a feature extraction module based on the attention mechanism. To make full use of the depth feature our attention-based module gets, we need to guide the depth estimation by the global structure information. Besides, our LF-DWNet can integrate the depth feature from multi streams of attention-based feature extraction modules and get more credible depth map. Experiments on both synthetic and real-world datasets demonstrate the effectiveness of our method.

Keywords: Light field · Depth estimation · Disparity warping · Attention mechanism · Global integration network

1 Introduction

Light field (LF) images acquired by LF cameras can store both spatial and angular information of the observed scene. Due to the unique advantages of containing multi-view geometry characteristics, LF images can be helpful in depth estimation and many other fields. Depth estimation has been widly used in 3D reconstruction, target tracking, virtual reality, and other fields [1], which has attracted much attention in recent years.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 291–302, 2022.
https://doi.org/10.1007/978-3-031-19214-2_24

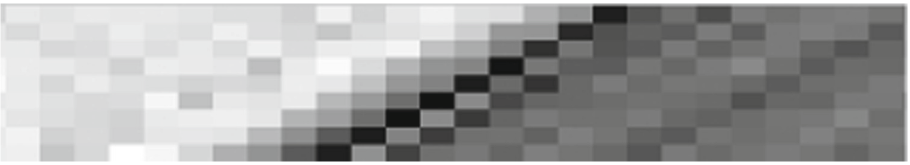


Fig. 1. The inclined lines on EPI with a large disparity.

To extract LF features for depth estimation, the main approach is to convert the 4D LF data into various 2D images, such as multi-view images, focal stacks, and epipolar plane images (EPIs). Methods based on EPI are the most commonly used methods in LF depth estimation [2–8]. EPI is the 2D slice of the 4D LF image, which presents 1D angular information and 1D spatial information from the same direction of LF. Due to the consistency of multi-view images, EPI shows patterns of oriented lines with constant colors, and the slope of each line represents the disparity of corresponding point in the scene. But as shown in Fig. 1, it can be hard to measure the lines since the insufficient angular resolution, especially when the scene has a large disparity. Besides, occlusions and noise can also make the extraction of the lines more difficult. Conventional methods try many complex optimization approaches to reduce the effect of unsatisfactory conditions. These years, deep learning has been widely used in LF depth estimation. The data-driven methods show stronger competitiveness when facing complex and diverse scenes.

In this paper, we propose a learning-based method for LF depth estimation, which shows strong robustness in large disparity pixels, occlusions, and noise areas. We use parallelograms with different inclination angles to match lines on EPI, and warp the parallelograms into rectangles to make the depth feature free from disturbance of large disparity pixels. Based on the disparity warping of EPI, we design an attention-based feature extraction module to extract the depth feature from warped EPI, and propose a global integration network based on this module to get the credible depth map, which can reduce the effect of occlusions and noise areas. Experiments on synthetic LF images [9,10] and real-world LF images [11] can demonstrate the effectiveness of our method.

In summary, the contributions of this paper are as follows:

- To reduce the effect of large disparity pixels, we introduce the disparity warping processing on EPI, which also make LF data more convenient to handle with the network.
- To extract the depth feature from warped EPI and reduce the effect of occlusions and noise areas, we design a feature extraction module based on the attention mechanism.
- Based on the attention-based feature extraction module, we propose the depth estimation network named LF-DWNet, and it realizes the state-of-the-art performance in LF depth estimation.

2 Related Work

2.1 Conventional Methods

Conventional methods of LF depth estimation based on EPI calculate the slope of the lines on EPI through some conventional measurement formulas to get the initial depth map, and refine the depth map through some optimization approaches. Wanner et al. [2] proposed a method based on the structure tensor to measure the slope of the lines on EPI and get the depth map of the scene. And they used a fast total variation denoising filter to refine the depth map and get more credible results. Johannsen et al. [3] used EPI patches to compose a dictionary with a corresponding known disparity, and query the EPI features in this original dictionary to get the depth map. Zhang et al. [4] designed a spinning parallelogram operator (SPO) to measure the slope of the lines on EPI. They spun the parallelogram to find the direction that is able to divide the parallelogram into two parts with the largest histogram differences. To reduce the effect of occlusions, they combined the results from the horizontal EPI and the vertical EPI based on confidence. And they used the guided filter to provide the global structure information for the more robust depth map. SPO also inspires our method, which will be introduced in Sect. 3.2 in detail.

Some methods use stereo matching or focal stacks instead of EPI to get the depth map of the scene. Chen et al. [12] introduced a method based on the surface camera (SCam). They used a bilateral consistency metric to tackle occlusions in LF depth estimation. Jeon et al. [13] proposed a multi-view stereo matching with a phase-based sub-pixel shift. And they adopted the weighted median filter and the multi-label optimization to reduce the effect of image noise and the weak texture regions. Tao et al. [14] combined both defocus and correspondence depth cues from LF images to get the depth map. They improved the approach by adding a shading-based refinement technique.

Most of the conventional methods analyze and model only for limited conditions. Even if a series of optimization methods are adopted, it is difficult to apply to a large number of different complex situations in the actual scene. Data-driven approaches based on convolutional neural networks (CNN) usually perform better than these optimization-based methods [15–19]. Although conventional methods are not very competitive today, they can also inspire learning-based methods.

2.2 Learning-Based Methods

In the past few years, machine learning techniques have been applied to a variety of LF imaging applications such as super-resolution [20], view synthesis [21], material recognition [22], and depth estimation [5–8,23–25].

For depth estimation, several methods are proposed to extract the linear feature of EPI through well-designed networks. Heber et al. [5] trained a CNN to predict the orientations of lines on EPI, and formulated a global optimization with a higher-order regularization to refine the predictions of their network.

Shin et al. [6] proposed an end-to-end network with the multi-stream of EPI to reduce the effect of occlusions. And they also introduced a specifically designed data augmentation approach to address the issue of the lack of training data for the network. Leistner et al. [7] proposed a technique to virtually shift the LF stack which can keep the slope of the lines on EPI in a small receptive field, independent of the disparity range. Li et al. [8] also designed an EPI-based oriented relation network to get the depth map.

Some networks do not utilize EPI. Tsai et al. [23] proposed an attention-based view selection network to adaptively incorporate LF images of all views for depth estimation. They used the attention mechanism to extract the feature hidden behind multi-views and it performed better than methods based on EPI at that time. Zhou et al. [24] trained a two-pathway CNN to predict the depth of each pixel from the LF focal stack. Their network learns the depth semantic features and the low-level structure information from the focal stack and the central view. Wang et al. [25] proposed the disentangling mechanism and designed three disentangling networks for LF spatial super-resolution, angular super-resolution, and disparity estimation. All these three methods based on the disentangling mechanism show strong competitiveness in their respective fields. Besides, Wang et al. [26] also proposed a matching cost constructor which is occlusion-aware and efficient in depth estimation.

These years, learning-based methods of LF depth estimation have attracted much attention. Large disparity pixels, occlusions, and noise of images prevent most of them from getting credible results in complex scenes. In this paper, we address the issue by proposing a new network based on disparity warping of EPI, and get more robust results through our methods.

3 Method

3.1 Overall Framework

As shown in Fig. 2, our method is to train a global integration network using an attention-based module to extract the depth feature from warped EPI. Our network has four different streams based on EPI of horizontal, vertical, left diagonal, and right diagonal directions, which can improve the robustness to occlusions and noise. For every stream, we warp the EPI as described in Sect. 3.2, and extract the depth feature as described in Sect. 3.3.

To make full use of the depth feature our attention-based module gets, we need to guide the depth estimation by the global structure information. The attention mechanism on EPI may cause some global structure information in 4D LF images to be ignored, and the warping will also lead to some tiny errors which should be corrected by the original images. We concatenate the feature obtained from our attention-based feature extraction module with the original sub-aperture images corresponding to the EPI used in this stream. Then, we use three residual blocks for deep feature extraction, which can be expressed as follows:

$$F_{out,i} = H_{res,3}(H_{res,2}(H_{res,1}(F_{in,i} \oplus I_i))) \tag{1}$$

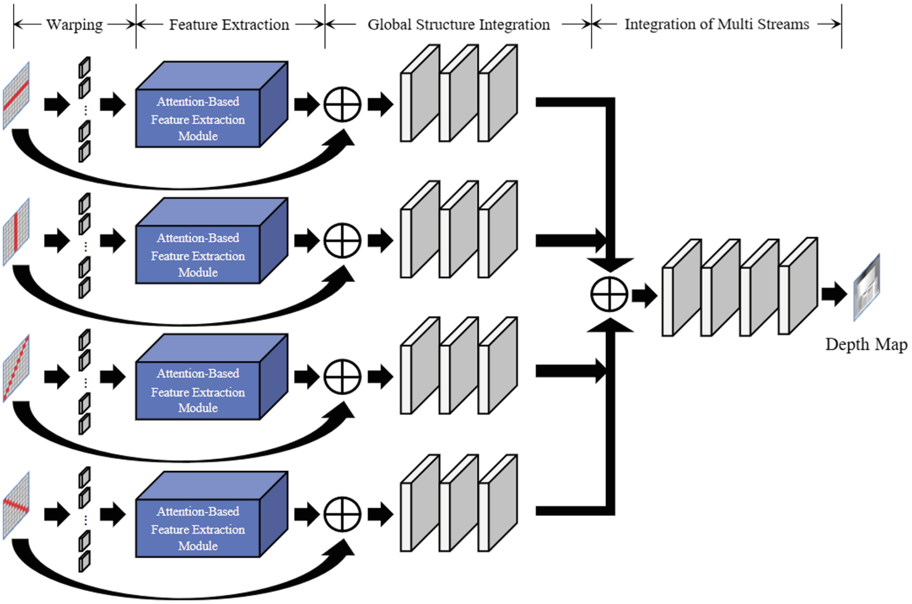

Warping
Feature Extraction
Global Structure Integration
Integration of Multi Streams
Attention-Based Feature Extraction Module
Attention-Based Feature Extraction Module
Attention-Based Feature Extraction Module
Attention-Based Feature Extraction Module
Depth Map

Fig. 2. The architecture of our network.

where F_{out}, i denotes the output of stream i ($i = 1, 2, 3, 4$). $H_{res,1}$, $H_{res,2}$ and $H_{res,3}$ are three residual blocks. $F_{in,i}$ denotes the input feature obtained from the module introduced in Sect. 3.3, and I_i represents the original sub-aperture images used in this stream.

Four more residual blocks are used to integrate the output from four different streams and get the credible depth map from integrating data. The residual structure can extract vital information from the depth feature and prevent our deep network from the vanishing gradient problem. By training this end-to-end network, we can get an effective method for LF depth estimation, which is robust in large disparity pixels, occlusions, and noise areas.

3.2 Disparity Warping of EPI

The 4D LF image can be represented as $L(x, y, u, v)$, where (x, y) is the spatial coordinate and (u, v) is the angular coordinate. Due to the consistency of multi-view images, the relationship of pixels between the center and the other views of the LF image can be expressed as follows:

$$L(x, y, u, v) = L(x + d_{x,y} \times (u' - u), y + d_{x,y} \times (v' - v), u', v') \tag{2}$$

where $d_{x,y}$ is the disparity of the pixel (x, y) between adjacent views. For the horizontal EPI, we have the relationship of $v = v'$ and $y = y'$, then we can reformulate the relationship of Eq. 2 as follows:

$$L_{y,v}(x, u) = L_{y,v}(x + d_{x,y} \times (u' - u), u') \tag{3}$$

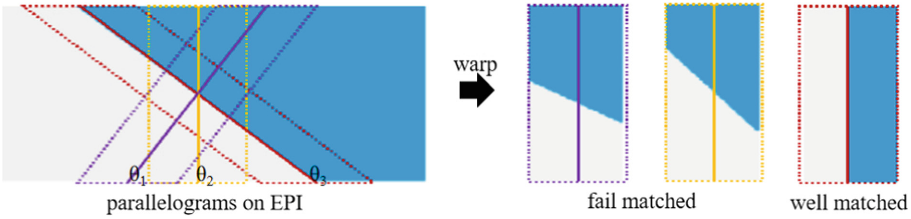


Fig. 3. The disparity warping of parallelograms on EPI.

This formula expresses the linear feature of EPI, which is widely used in LF depth estimation.

Inspired by SPO [4], we can use parallelograms with different inclination angles to match lines on EPI. But the feature is not so easy to extract when the angular resolution is not so sufficient and the scene has a large disparity. When we convert the parallelograms into rectangles, as shown in Fig. 3, the successfully matched rectangle shows a vertical central axis, while other rectangles have an inclined line passing through the center point. This transformation can make the feature independent of disparity, which is of great help reduce the effect of large disparity pixels. We can take a series of disparity labels equidistant according to the disparity range of scenes, and each label is corresponding to parallelograms with a certain inclination angle. We warp the pixels from view (u', v) to (u, v) based on the disparity label, which results in a spatial transformation from (x', y) to (x, y). This transformation is equivalent to warping the parallelograms with the certain inclination angle into rectangles. In practice, we can warp all the pixels at once, and recognize the vertical line from each rectangle on warped EPI. The disparity warping also makes it easier to extract the depth feature with the network.

To get the sub-pixel information of EPI in the warping, we need to adopt some interpolation methods. To reduce the errors, we process the warping on the frequency domain through a 2D Fourier transform, which can be expressed as follows:

$$I(x + \varDelta x) = F^{-1}(F(I(x + \varDelta x))) = F^{-1}(F(I(x))e^{2\pi i \varDelta x}) \quad (4)$$

where $F(\cdot)$ and $F^{-1}(\cdot)$ are the 2D Fourier transform and the 2D inverse Fourier transform.

3.3 Attention-Based Feature Extraction Module

Due to the influence of occlusion and noise, lines on the EPI may not be complete and easy to recognize, which makes it hard to apply some conventional mathematics measurement methods. Learning-based methods can often perform better on complex classification and identification problems. We propose an attention-based feature extraction module to find EPI warped based on the ground truth

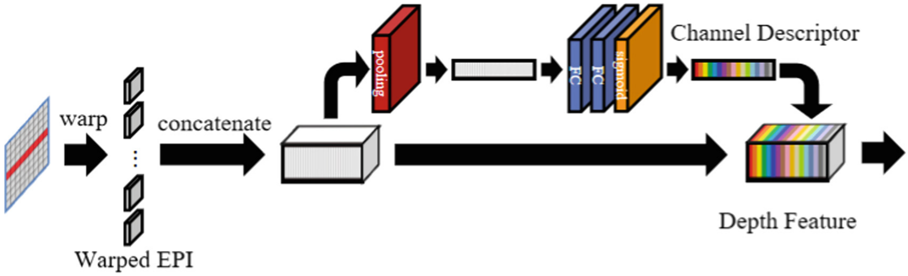


Fig. 4. The attention-based feature extraction module.

of disparity rather than some common network models. The attention mechanism has been proved to be able to extract the key feature from a large amount of information in many other computer vision tasks. In LFattNet [23], it is used to indicate the importance of every individual view in LF images. To get better performance in complex scenes, we design an attention-based feature extraction module and it shows the obvious advantage in extracting the depth feature from warped EPI mentioned in Sect. 3.2.

As shown in Fig. 4, the input of the attention-based feature extraction module is the EPIs warped based on different labels of disparity, which are arranged by channel. So the size of the input feature map is (H, W, kN), where H and W are the height and width of the LF sub-aperture image, that is, the spatial resolution of LF images. N denotes the angular resolution of LF images, and k denotes the number of disparity labels used for warping.

In the attention-based feature extraction module, the global spatial information of the feature map is squeezed into a channel descriptor by global average pooling. Then, we apply two fully-connected (FC) layers to the channel descriptor. The scale can be expressed as follows:

$$s = f(H_{FC2}(\delta(H_{FC1}(H_{pooling}(F_{in}))))) \tag{5}$$

where $f(\cdot)$ and $\delta(\cdot)$ denote the sigmoid and ReLU function. $H_{pooling}$ is the operation of global average pooling. H_{FC1} and H_{FC2} are two FC layers, while H_{FC1} downsamples the channels and H_{FC2} upsamples the channels. The output of the module is:

$$F_{out} = s \cdot F_{in} \tag{6}$$

Through the attention mechanism on the channel, we can adjust the weights of channels and find effective information from warped EPI, which is useful in the LF depth estimation network.

Table 1. Average MSE and BadPix achieved by different methods on synthetic datasets.

	HCI_new				SLFD			
	MSE×100↓	BP0.01↓	BP0.03↓	BP0.07↓	MSE↓	BP0.05↓	BP0.1↓	BP0.3↓
SPO [4]	3.572	61.82	19.23	8.231	1.138	35.81	20.52	6.574
EPINET [6]	1.753	27.33	7.823	3.580	1.597	83.02	67.21	34.96
EPI-Shift [7]	4.948	65.42	32.79	17.08	1.591	67.21	56.46	32.67
EPI_ORM [8]	3.155	46.31	15.61	8.738	1.004	63.15	43.83	22.99
LFattNet [23]	1.350	**15.07**	5.269	**2.839**	–	–	–	–
DistgDisp [25]	1.415	21.71	7.329	3.867	0.581	42.76	25.18	7.547
Ours	**1.243**	15.26	**5.233**	2.884	**0.390**	**35.14**	**16.65**	**5.603**

4 Experiments

4.1 Datasets and Implementation Details

We use three datasets for the experiment of LF depth estimation, which include synthetic LF images from HCI_new [9] and SLFD [10], and real-world images provided by Stanford Lytro Light Field Archive [11]. HCI_new is the most commonly used synthetic LF dataset which provides 24 scenes with ground truth depth released. SLFD contains 53 synthetic LF images with a much larger disparity range which can evaluate the robustness of methods on large disparity pixels. The real-world images provided by Stanford are captured by a Lytro Illum camera. The angular resolution of the Lytro images is 14 × 14, and we use the middle 9 × 9 views to estimate the depth.

To train our network, we use 16 images from HCI_new dataset and crop them into patches of size 32 × 32. We also apply some data augmentation methods including random horizontal flipping, vertical flipping, and rotation to prevent the overfitting problem of our network. We use Adam optimizer and set the batch size to 16. The learning rate is initially set to 10^{-5} and decreases to 10^{-6} after 100 epochs. Our network is implemented in Pytorch and is trained on an NVIDIA RTX 3090 GPU for about five days. Then we finetune our network and evaluate the robustness of the method on SLFD dataset.

4.2 Experimental Results

We use mean square error (MSE) and error rate (BadPix) for the quantitative evaluation of LF depth estimation. In the experiment, we compare our method with six state-of-the-art methods. Five of them are learning-based methods, and we train these networks on the same datasets for fairness.

As shown in Table 1, our method achieves the lowest MSE and BadPix on SLFD, and achieves the lowest MSE and BadPix 0.03 on HCI_new. For scenes with complex occlusions, such as ‘Boxes’, our method is more robust than other methods. And for the images with a large range of disparity, our method can perform much better than other methods. The detailed results of every scene

Table 2. MSE×100/BP0.01/BP0.03/BP0.07 achieved by different methods on HCI_new.

	Boxes	Cotton	Dino	Sideboard
SPO [4]	9.107/73.23/29.52/15.89	1.313/69.05/13.71/2.594	0.310/69.87/16.36/2.184	1.024/73.36/28.81/9.297
EPINET [6]	6.036/45.73/18.66/12.25	0.223/25.27/2.217/0.464	0.151/23.44/3.221/1.263	0.806/40.49/11.82/4.783
EPI-Shift [7]	9.790/74.36/44.14/25.95	0.475/46.86/10.68/2.176	0.392/64.16/22.14/5.964	1.261/73.42/36.64/11.80
EPI_ORM [8]	4.189/59.68/25.33/13.37	0.287/42.94/5.564/0.856	0.336/41.04/8.993/2.814	0.778/52.59/14.61/5.583
LFattNet [23]	3.996/37.04/18.97/11.04	0.209/**3.664**/0.697/0.271	**0.093/12.22/2.339/0.848**	**0.530/20.73/7.243/2.869**
DistgDisp [25]	3.325/41.62/21.13/13.31	0.184/7.594/1.478/0.489	0.099/20.46/4.018/1.414	0.713/28.28/9.575/4.051
Ours	**3.316/35.94/17.76/10.58**	**0.168**/3.670/**0.686/0.259**	0.106/13.46/2.786/1.038	0.586/21.06/7.468/3.104
	Backgammon	Dots	Pyramids	Stripes
SPO [4]	4.587/49.94/8.639/3.781	5.238/58.07/35.06/16.27	0.043/79.20/6.263/0.861	6.955/21.87/15.46/14.97
EPINET [6]	3.909/15.39/4.482/3.287	1.980/44.64/18.70/4.030	0.007/8.913/0.604/0.147	0.915/**14.75/2.876/2.413**
EPI-Shift [7]	12.79/70.58/40.53/22.89	13.15/74.55/53.18/43.92	0.037/40.48/7.315/1.242	1.686/78.95/47.70/22.72
EPI_ORM [8]	**3.411**/34.32/7.238/3.988	14.48/65.71/47.93/36.10	0.016/19.06/1.301/0.324	1.744/55.14/13.94/6.871
LFattNet [23]	3.648/11.58/3.985/3.126	1.425/15.05/3.012/1.432	**0.004/2.063/0.488**/0.195	**0.892**/18.21/5.417/2.933
DistgDisp [25]	4.712/26.17/10.54/5.824	1.367/25.37/4.464/1.826	**0.004**/4.953/0.539/**0.108**	0.917/19.25/6.885/3.913
Ours	3.582/**11.26/3.798/3.008**	**1.276/14.98/2.974/1.388**	0.005/3.012/0.496/0.202	0.906/18.68/5.894/3.496

Table 3. MSE/BP0.05/BP0.1/BP0.3 achieved by different methods on SLFD.

	Electro_devices	Furniture	Lion	Toy_bricks
SPO [4]	1.734/47.37/26.48/8.525	1.662/53.69/30.50/10.69	0.221/29.14/19.27/4.067	0.936/**13.05/5.835/3.013**
EPINET [6]	0.756/81.06/61.94/23.77	1.715/84.88/70.76/37.74	3.265/85.98/76.17/55.86	0.652/80.14/59.98/22.48
EPI-Shift [7]	0.742/68.91/58.86/27.94	1.854/76.33/65.48/38.69	2.768/53.41/39.98/28.76	0.998/70.19/61.52/35.28
EPI_ORM [8]	**0.405**/45.39/**15.96/5.493**	0.893/65.26/49.28/21.58	1.625/72.15/58.63/40.17	1.096/69.78/51.45/24.71
DistgDisp [25]	0.584/43.27/25.18/7.956	0.765/51.83/28.96/9.851	0.285/**26.99**/19.16/4.213	0.689/48.95/27.43/8.166
Ours	0.426/**39.68**/21.45/6.398	**0.578/42.17/24.99/7.806**	**0.084**/29.75/**6.734/2.103**	**0.472**/28.96/13.41/6.105

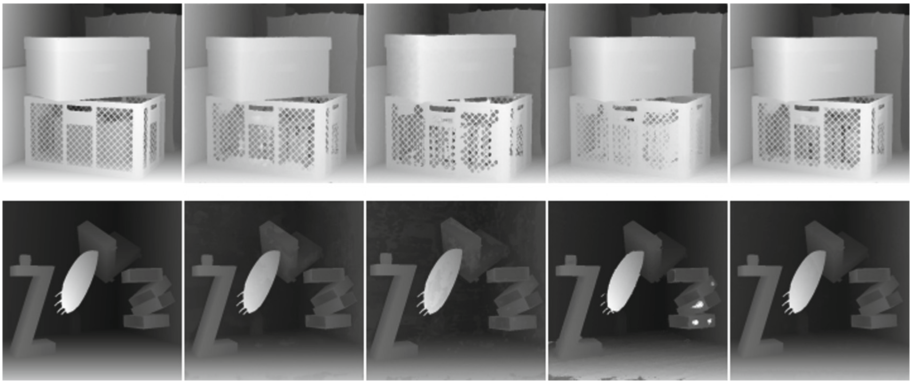


Fig. 5. Results achieved by different methods on synthetic LF images. The first column is the ground truth of the disparity map. From the second column to the fifth row: DistgDisp [25], EPINET [6], SPO [4], Ours.

are listed in Table 2 and Table 3. As a supplement, some results on 'Boxes' and 'Furniture' are shown in Fig. 5.

To fully demonstrate the robustness of our LF-DWNet, we compare our method with other methods on real-world images. As shown in Fig. 6, our method

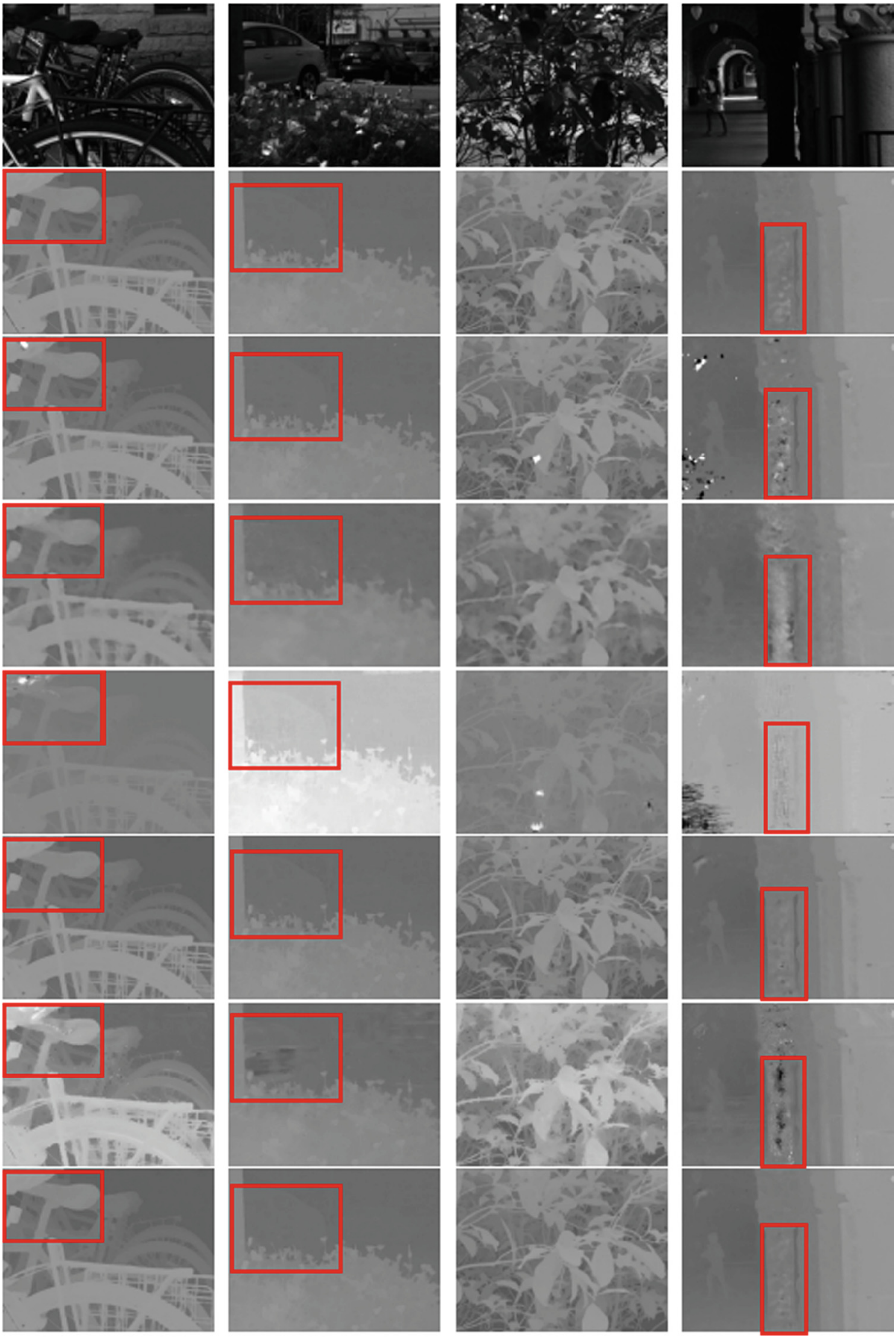


Fig. 6. Results achieved by different methods on real-world LF images. The first row is the image of the center view. From the second row to the eighth row: DistgDisp [25], EPINET [6], EPI-Shift [7], EPI_ORM [8], LFattNet [23], SPO [4], Ours.

is robust in large disparity pixels, such as the handle closest to the camera in the first scene. For the complex occlusions in the third scene, our method can also perform well. Compared with other methods, our method is less affected by the noise of real-world images, and can accurately estimate the depth of distant objects.

5 Conclusion

In this paper, we propose a learning-based method using disparity warping on EPI for LF depth estimation. We design an attention-based feature extraction module to extract the depth feature from warped EPI, and propose a global integration network based on this module to get the credible depth map. Experiments demonstrate the strong robustness of our network in large disparity pixels, occlusions, and noise areas. And our method can achieve state-of-the-art performance on both synthetic and real-world datasets.

Acknowledgements. This study is partially supported by the National Key R&D Program of China (No. 2018YFB2101100), the National Natural Science Foundation of China (No. 61872025), and the Science and Technology Development Fund, Macau SAR (File no. 0001/2018/AFJ) and the Open Fund of the State Key Laboratory of Software Development Environment (No. SKLSDE-2021ZX-03). Thank you for the support from HAWKEYE Group.

References

1. Wu, G., et al.: Light field image processing: an overview. IEEE J. Sel. Top. Signal Process. **11**(7), 926–954 (2017)
2. Wanner, S., Goldluecke, B.: Globally consistent depth labeling of 4D light fields. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (2012)
3. Johannsen, O., Sulc, A., Goldluecke, B.: What sparse light field coding reveals about scene structure. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (2016)
4. Zhang, S., Sheng, H., Li, C., Zhang, J., Xiong, Z.: Robust depth estimation for light field via spinning parallelogram operator. Comput. Vis. Image Underst. CVIU **145**, 148–159 (2016)
5. Heber, S., Pock, T.: Convolutional networks for shape from light field. In: Computer Vision and Pattern Recognition (2016)
6. Shin, C., Jeon, H.G., Yoon, Y., Kweon, I.S., Kim, S.J.: EPINET: a fully-convolutional neural network using epipolar geometry for depth from light field images. IEEE (2018)
7. Leistner, T., Schilling, H., Mackowiak, R., Gumhold, S., Rother, C.: Learning to think outside the box: wide-baseline light field depth estimation with EPI-shift (2019)
8. Li, K., Zhang, J., Sun, R., Zhang, X., Gao, J.: EPI-based oriented relation networks for light field depth estimation (2020)
9. Honauer, K., Johannsen, O., Kondermann, D., Goldluecke, B.: A dataset and evaluation methodology for depth estimation on 4D light fields. In: Asian Conference on Computer Vision (2016)

10. Shi, J., Jiang, X., Guillemot, C.: A framework for learning depth from a flexible subset of dense and sparse light field views. IEEE Trans. Image Process. **28**(12), 5867–5880 (2019)
11. The Stanford light field archive. http://lightfields.stanford.edu/LF2016.html. Accessed 15 Apr 2022
12. Chen, C., Lin, H., Zhan, Y., Kang, S.B., Yu, J.: Light field stereo matching using bilateral statistics of surface cameras. In: Computer Vision and Pattern Recognition (2014)
13. Jeon, H.G., et al.: Accurate depth map estimation from a lenslet light field camera. In: Computer Vision and Pattern Recognition, pp. 1547–1555 (2015)
14. Tao, M.W., Hadap, S., Malik, J., Ramamoorthi, R.: Depth from combining defocus and correspondence using light-field cameras. In: Proceedings of the 2013 IEEE International Conference on Computer Vision (2013)
15. Wang, Y., Cai, Z., Zhan, Z.H., Zhao, B., Qi, L.: Walrasian equilibrium-based multiobjective optimization for task allocation in mobile crowdsourcing. IEEE Trans. Comput. Soc. Syst. **7**(4), 1033–1046 (2020)
16. Tang, W., Hui, B., Tian, L., Luo, G., Cai, Z.: Learning disentangled user representation with multi-view information fusion on social networks. Inf. Fusion **74**(4), 77–86 (2021)
17. Xiong, Z., Cai, Z., Han, Q., Alrawais, A., Li, W.: ADGAN: protect your location privacy in camera data of auto-driving vehicles. IEEE Trans. Industr. Inform. **17**(9), 6200–6210 (2020)
18. Cai, Z., Zheng, X., Yu, J.: A differential-private framework for urban traffic flows estimation via taxi companies. IEEE Trans. Industr. Inform. **15**(12), 6492–6499 (2019)
19. Xiong, Z., Xu, H., Li, W., Cai, Z.: Multi-source adversarial sample attack on autonomous vehicles. IEEE Trans. Veh. Technol. **70**(3), 2822–2835 (2021)
20. Li, D., Yang, D., Wang, S., Sheng, H.: Light field super-resolution based on spatial and angular attention. In: International Conference on Wireless Algorithms, Systems, and Applications (2021)
21. Zhang, S., Sheng, H., Yang, D., Zhang, J., Xiong, Z.: Micro-lens-based matching for scene recovery in lenslet cameras. IEEE Trans. Image Process. **27**(3), 1060–1075 (2017)
22. Wang, T.-C., Zhu, J.-Y., Hiroaki, E., Chandraker, M., Efros, A.A., Ramamoorthi, R.: A 4D light-field dataset and CNN architectures for material recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 121–138. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_8
23. Tsai, Y.J., Liu, Y.L., Ming, O., Chuang, Y.Y.: Attention-based view selection networks for light-field disparity estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 7, pp. 12095–12103 (2020)
24. Zhou, W., Zhou, E., Yan, Y., Lin, L., Lumsdaine, A.: Learning depth cues from focal stack for light field depth estimation. In: 2019 IEEE International Conference on Image Processing (2019)
25. Wang, Y., et al.: Disentangling light fields for super-resolution and disparity estimation. IEEE Trans. Pattern Anal. Mach. Intell. (2022)
26. Wang, Y., Wang, L., Liang, Z., Yang, J., An, W., Guo, Y.: Occlusion-aware cost constructor for light field depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19809–19818 (2022)

Toward Multi-sided Fairness: A Fairness-Aware Order Dispatch System for Instant Delivery Service

Zouying Cao[1], Lin Jiang[2], Xiaolei Zhou[3](✉), Shilin Zhu[1], Hai Wang[1], and Shuai Wang[1]

[1] School of Computer Science and Technology, Southeast University, Nanjing, China
{zouyingcao,shilinzhu,hai,shuaiwang}@seu.edu.cn

[2] School of Cyber Science and Engineering, Southeast University, Nanjing, China
linjiang@seu.edu.cn

[3] The Sixty-Third Research Institute, National University of Defense Technology, Changsha, China
zhouxiaolei@nudt.edu.cn

Abstract. Instant delivery platforms, equipped with professional couriers to provide convenient delivery services, have emerged rapidly in many cities. For the benefit of platforms, many researchers focus more on maximizing overall efficiency but ignore individual fairness. Current fairness research in mobile systems mainly concentrates on one-sided or two-sided relationships, such as drivers and customers. However, instant delivery services have two new characteristics in fairness: (i) **multi-stakeholder involvement**, namely couriers, merchants and users should be considered comprehensively; (ii) more complicated matching relationship because of **the concurrent dispatch mode**, meaning one courier will handle multiple orders simultaneously. To handle this multi-sided fairness problem, our paper proposes a novel order dispatch system to balance the platform revenue and multi-stakeholder fairness. Motivated by the analysis of real-world datasets, we formulate the order dispatch problem as a sequential decision-making problem and incorporate multi-sided fairness into the decision criteria. Then, we design a multi-sided fairness-aware deep reinforcement learning algorithm to solve large-scale decision problem, with the fairness relying on Least Misery Fairness definition for users and Variance Fairness definition for couriers and merchants. Finally, extensive experiments show the effectiveness of our model in balancing multi-sided fairness among stakeholders and long-term profits of the whole platform.

Keywords: Order dispatch · Multi-sided fairness · Instant delivery · Reinforcement learning

1 Introduction

With the rapid development of O2O (online to offline) and New Retailing, instant delivery services have gained much popularity and facilitate people's daily lives

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 303–316, 2022.
https://doi.org/10.1007/978-3-031-19214-2_25

enormously. Driven by the growing demand, many popular platforms (e.g., DoorDash, UberEats, Instacart and MeiTuan) provide fast delivery services to help their customers acquire food, medicine, and groceries quickly. In 2020, mainland China had over 22.85 billion instant delivery orders, marking a year-on-year increase of 25%. Although the rapid growth of instant delivery generates huge economical profit, it leads to multiple challenges in social governance which deserves to be studied to solve.

Most of the research efforts [7,20,21] for instant delivery services concentrate on maximizing the efficiency for order dispatch and improving the experiences at the users' side. However, the issue of fairness in sharing economy attracts widely attentions from the whole society. As the number of registered couriers grows, it is crucial to guarantee the fairness among their incomes, and the same is true for merchants. Besides, violating user fairness is not only ethically fraught, but also unfavorable for securing the platform performance in the long term.

The current fair matching mechanism has several drawbacks. Firstly, some matching models [3,6,9] only pay attention to one-sided fairness but ignore the overall fairness among other stakeholders. Secondly, instant delivery service has concurrent dispatch mode, which means one courier can process multiple orders simultaneously. However, most algorithms only focus on the sequence dispatch mode which is common on ride-hailing platforms [14,19]. Therefore, these latest algorithms are not suitable for instant delivery platforms, and we are interested in investigating a novel order dispatch system to ensure the multi-sided fairness in instant delivery platforms.

Since order dispatch decisions are ordered by time, we can explore the use of Reinforcement Learning (RL) [8,10,22] in the instant delivery serving multi-sided fairness. In addition, with massive historical dispatch and route records, we can amortize equality in multi-sided systems over longer periods (e.g., weeks or months) and then extend the concept of fairness to multiple stakeholders based on an empirical data analysis. Similar ideas of fairness amortization [14] have been utilized in the context of ranking [2,13] and recommender systems [1,12].

However, seeking an optimal fairness-aware order dispatch algorithm applying to this new multi-stakeholder commercial platform is not an easy task due to two challenges: (i) **uncertain multi-sided fairness notions** as different stakeholders may have different perceptions of fairness; (ii) **potential conflicting relationships** within the same stakeholder group and between different stakeholder groups. For example, reducing income inequality among couriers may result in inefficient service as well as loss and disparity in customer utilities.

To tackle the above challenges, we propose an Advantage Actor-Critic-based deep reinforcement learning approach to learn the Multi-sided Fairness-aware order dispatch policy called **A2CMF**. Then, we establish two notions of fairness based on the "variance" fairness semantics for the couriers and merchants to maintain equality, and utilize the "least misery" [18] to guarantee the user waiting time within reasonable bounds. Specifically, we design a policy network in A2CMF which integrates state & action embedding features and two fairness metrics into an accumulated reward. And different from traditional actor-critic algorithm, the action space as input is designed variable to handle the uncertain number of optional couriers considering user fairness constraints.

In summary, the salient contributions of this paper are as follows.

- To the best of our knowledge, we perform the first work on multi-sided (tripartite or more) fairness-aware order dispatch policy in an instant delivery platform. Our approach is proposed with 1,159,371 order records in one month relevant to 595 merchants and about 4,000 couriers. We believe our proposal would contribute to further explore the fair matching issue when the new commercial pattern brings the complicated multiple service suppliers model.
- We consider multi-sided notions of fairness which not only relate to the fair income distribution for couriers, the fair service experience among customers and merchants but also the long-term profitability of multi-stakeholder platforms. Such an idea would be helpful to address the fairness concerned issues in similar sharing economy scenarios.
- More importantly, to better train the A2CMF Network, we design a data-driven simulator to model the real-time instant delivery environment with dynamic demand & supply, spatial-temporal features and complicated courier behaviors. Then we evaluate the performance of A2CMF through extensive experimentation with data from Eleme (one of the largest instant delivery companies in China). The evaluation results show that our A2CMF achieves a 21.3% increase in total revenue, improves the profit fairness of couriers by 9.7%, and reduces customers' waiting time and the benefit gap among merchants by 6.9% and 6.2%, simultaneously.

2 Background and Motivation

2.1 Instant Delivery Scenario

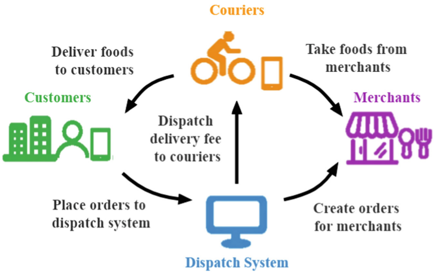

Couriers
Deliver foods to customers
Take foods from merchants
Customers
Merchants
Dispatch delivery fee to couriers
Place orders to dispatch system
Create orders for merchants
Dispatch System

Fig. 1. Four stakeholders in instant delivery

Table 1. Order progress record

Field	Value
User/Courier/Merchant ID	U001/C001/M001
Food Amt/Delivery Fee	32.99/3.8
Promise Delivery Time	3300
Merchant Location	121.45916,31.25554
User Location	121.46889,31.25317
Order Create Time	2020/10/1 11:59:00
Accept Order Time	2020/10/1 12:03:00
Arrive Restaurant Time	2020/10/1 12:07:00
Pickup Time	2020/10/1 12:09:00
Delivery Time	2020/10/1 12:17:00

As illustrated in Fig. 1, a typical instant delivery service involves four stakeholders: **couriers**, **merchants**, **customers** and the **platform**. And their corresponding information combined with five critical timestamps will be recorded during the order dispatching process (e.g., listed in Table 1). Then, the roles of the four stakeholders in instant delivery will be briefly introduced.

(i) **Couriers** are assigned order tasks by the platform, and need to pick up orders in merchants and deliver to customers in time. The fairness appeal of couriers is that they can get the same labour efficiency when having the same working hours. (ii) **Merchants** receive orders from customers and are arranged couriers by platform. From the perspective of merchants, they want to get couriers to pick up prepared orders as soon as possible. (iii) When **customers** place orders through the platform and look forward to acquiring them on schedule, it's better to have early delivery. (iv) As the principal of dispatch algorithm, **platform** has the primary aim to obtain more benefit, but it also has the responsibility to consider the fair requirements of the other three stakeholders. Only achieve a trade-off among the above four aspects, can the instant delivery platform realize a stable operation in the long run.

2.2 Characteristics of Fairness in Order Dispatch

Given the historical delivery order data, we conduct a data-driven order dispatch pattern analysis and obtain the following observations:

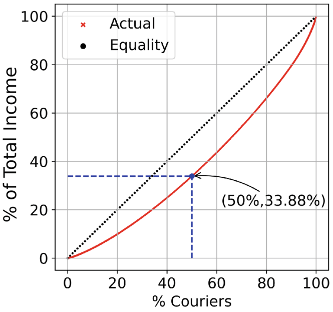


Fig. 2. Lorenz curve of courier income

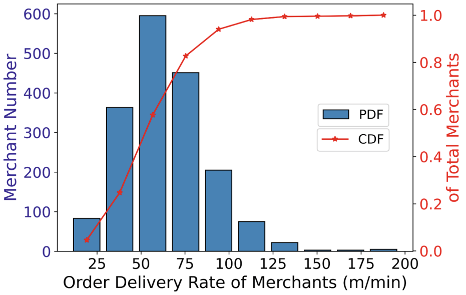


Fig. 3. Merchant fairness index

1. **Income Inequality among Couriers.** Figure 2 shows that after one day, 50% couriers only earned 33% of total income, while 20% most successful couriers get 35%. Couriers in the bottom ten percent of income almost made little money, which represents unequal income distribution among couriers.
2. **Unfair User Experience.** From user comments on the platform, we observe that positive and negative comments coexist and the comments are even polarized, showing unfair service experience issues among customers.
3. **Inequality in Merchant Benefit.** We further analyze the order delivery rate of each merchant and find the inequality in merchant benefit. It is demonstrated in Fig. 3 that 20% of merchants are severely lower than the average level while another 20% are significantly higher than the average.

3 System and Formulation

3.1 System Overview

We present the overview of our system design in Fig. 4 which is composed of **three** modules.

1. **An Environment Simulator for Instant Delivery.** We introduce a simulator design that models the events of order generation, order assignments and key stakeholder behaviors such as distributions across the city, along with changes in weather and traffic conditions in the real world.
2. **A State & Action Feature Extraction Module.** This part serves as a feature extractor to characterize multiple attributes important for order dispatching decisions, including order features, spatial features, temporal features and environmental features.
3. **A2CMF-Dispatch Model.** This model aims to learn an optimal fairness-aware order dispatch policy by calculating the long-term value for each candidate dispatch action via the Actor Network and then achieves a more stable and efficient model learning process via the Critic Network.

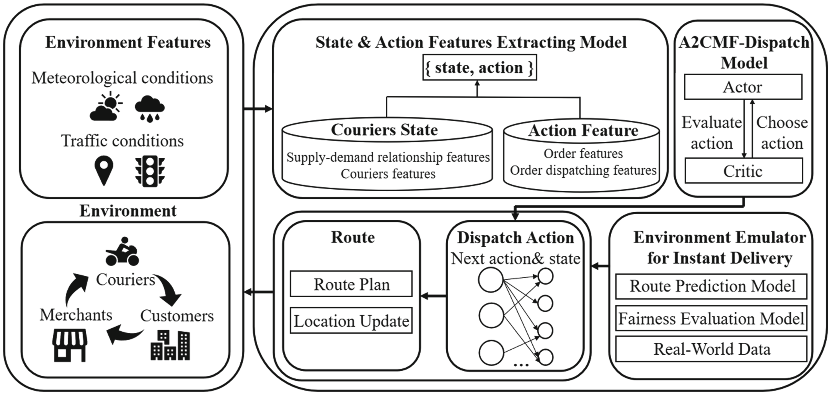


Fig. 4. System architecture of A2CMF

3.2 Problem Formulation

In this paper, we formulate the multi-sided fairness-aware order dispatch problem as *a multi−agent Markov decision process*, which is characterized by five components: $\{\mathcal{S}, \mathcal{A}, \mathcal{R}, \gamma, \pi\}$, i.e., the integration of states $\mathcal{S}$, the courier action space $\mathcal{A}$, the multi-sided fairness-oriented reward $\mathcal{R}$, a discount factor γ and the policy π to make fair matching decisions. Formally, we present the training process as finding a target policy π(a|s) so that dispatching actions τ according to π(a|s), would lead to the maximum expected cumulative reward:

$$max_{\pi_\theta}\mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)], \tag{1}$$

where $R(\tau) = \sum_{t=0}^{|\tau|} r(s_t, a_t)$ and θ denotes policy parameter. The specific definitions of the *multi−agent Markov decision process* in our **A2CMF** are listed below and policy π is introduced detailed in Sect. 4.2.

- **Agent Set.** We consider each candidate courier as an agent, and all couriers share the same dispatch strategy. In our system, the dispatch strategy is under centralized training, but does a decentralised execution based on every individual agent (courier) [16].
- **State $\mathcal{S}$.** We divide one day into $\mathcal{T}$ time slots and represent the state at time slot t as $s_{t \sim \mathcal{T}} = \{\mathcal{P}_t, \mathcal{ST}_t, \mathcal{D}_t, \mathcal{C}_t\}$, where $\mathcal{P}$ is the personal attribute of courier, $\mathcal{ST}$ is the set of spatiotemporal features, $\mathcal{D}$ is the global information about the distribution of stakeholders, and $\mathcal{C}$ are some contextual features.
  - $\mathcal{P}_t$: The personal state of a courier is defined as $\mathcal{P}_t = [loc, n_o, t_o, f, route_p]$, where loc is courier real-time location, n_o is the number of existing orders, t_o is his/her on-duty time and f marks whether this courier can deliver the order without disturbing the customer fairness index. Last, we use the idea designed by Zhang et al. [21] to predict $route_p$.
  - $\mathcal{ST}_t$: Note that couriers' locations are continuously changing, which will affect future order receive rate. We define a local-view state $\mathcal{ST}_t = \frac{N_o}{N_c}$ capturing the income opportunity where N_o and N_c are the total number of orders and rival couriers along the planning route, respectively.
  - $\mathcal{D}_t$: Shared by all couriers, the global-view state $\mathcal{D}$(consists of $\mathcal{O}, \mathcal{CO}, \mathcal{M}$ and $\mathcal{U}$) depicts the demand and supply distribution. The four matrices record the online number of four parts(i.e., orders, couriers, merchants, users) in each grid, representing a fine-grained distribution.
  - $\mathcal{C}_t$: In instant delivery, customers are tolerant of delayed packages [21] because of factors such as bad weather, rush hours and so on. So, we take into account those contextual information via one-hot encoding.
- **Action $\mathcal{A}$.** The agent action in our proposal is to recommend the optimal order to the courier waiting to be allocated. Thus, action features involve: (1) **order features** including price p, the create time t_c, the merchant location l_m and the customer location l_c; (2) **order dispatching attributes** including the delivery distance, the increased route time if the courier takes this order. Eventually, after choosing an optimal courier to take action, all related states need a proper transition $P(s_{t+1}|s_t, a_t) : S \times A \rightarrow S$.
- **Reward $\mathcal{R}$.** As the reward function, $\mathcal{R} = r(s_t, a_t)$ denotes the immediate reward of the action a_t at specific state s_t. It is designed to reach a balance between the overall platform revenue and the fairness among stakeholders:

$$r^{(i)}(s_t, a_t) = (1 - \alpha - \beta)PE(i, t) + \alpha \cdot (-CF(t)) + \beta \cdot (-MF(t)) \quad (2)$$

  where $\alpha, \beta \in [0,\ 1]$ balance the profit efficiency and two-sided fairness.
  - $PE(i, t)$ is the profit efficiency of the order i in the time slot t and set as

$$PE(i, t) = \gamma^{\Delta t} \times fee_i \quad (3)$$

    where Δt is the actual delivery time, γ is the discount factor in regard to the influence of time cost and fee_i is the delivery fee of order i.

- $CF(t)$ is a metric of profit fairness among couriers. For Variance fairness semantic, the fairness index is formulated as:

$$CF(t) = \frac{1}{N_c} \sum_{k=1}^{N_c} (CE(k,t) - \overline{CE(t)})^2, \tag{4}$$

the variance of profit efficiency CE of all N_c online couriers. And CE consists of two components: profit efficiency PE of each $order^i$ delivered by the $courier^k$ and working hours T_{work} measured in one time slot.

$$CE(k,t) = \frac{\sum_{i=1}^{m} PE(i,t)}{T_{work}(k,t)}, \overline{CE(t)} = \frac{1}{N_c} \sum_{k=1}^{N_c} CE(k,t) \tag{5}$$

- $MF(t)$ describes the profit fairness among merchants and borrows idea from Meituan that they think it is fair when merchants' products can be picked up and delivered to customers in time. Based on this, we define MF as a variance of the mean product value PV of all N_m merchants.

$$MF(t) = \frac{1}{N_m} \sum_{m=1}^{N_m} (PV(m,t) - \overline{PV(t)})^2, PV(m,t) = \frac{1}{N_o} \sum_{i=1}^{N_o} \frac{dist}{T_d^{(i)} - T_m^{(i)}} \tag{6}$$

where $N_o, dist, T_m^{(i)}, T_d^{(i)}$ denote the number of orders produced, the delivery distance, the time when $order^i$ is ready and delivered to the user.

Therefore, the Eq. 2 can be converted to Eq. 7.

$$r^{(i)}(s_t, a_t) = (1 - \alpha - \beta) \cdot (\gamma^{\Delta t} \cdot fee_i)$$
$$+\alpha \cdot (-\frac{1}{N_c} \sum_{k=1}^{N_c} (CE(k,t) - \overline{CE(t)})^2) + \beta \cdot (-\frac{1}{N_m} \sum_{m=1}^{N_m} (PV(m,t) - \overline{PV(t)})^2) \tag{7}$$

- **Discount factor** γ. γ selected from [0,1] discusses the time-based penalization for the rewards agent achieved in the past, present, and future.

4 Order Dispatch Model Design

In this section, we show how we solve the above formulated fairness-aware order dispatch problem with our advantage actor-critic(A2C)-based model **A2CMF**.

4.1 Environment Simulator Design

As real-world features give crucial content about dispatch decisions and actions can also affect the environment, an environment simulator for instant delivery plays a functional role in the performance of **A2CMF**. To simulate the real order dispatch environment better, we include the following features, generally can be classified into **five** categories:

- **Order Features.** Order features provide the basic information(e.g., price, create time, the corresponding merchant, and customer location).
- **Couriers Features.** Couriers distinguish from each other by their positions, capacity, working hours, number of existing orders, and route planning.
- **Supply-demand Relationship Features.** By capturing the real-time distribution of couriers and orders at the grid level, this kind of features describe the fine-grained supply-demand relationship in the instant delivery platform.
- **Order Dispatching Features (i.e., Action Features).** An order dispatching action is depicted by the planned route, the distance between merchant and courier, and the increased delivery time when the order is added.
- **Environmental Features.** Like meteorological conditions and traffic fleet, environmental features give contextual content about dispatch decisions.

4.2 Advantage Actor-Critic Network

The basic idea of A2C algorithm is that there are two networks, a policy network(i.e., Actor, utilized to calculate the possible long-term reward of the courier-order matching and learn a policy) and a value network (i.e., Critic, a state-value function and leveraged to evaluate the performance of the actor).

In our problem, we collect nearby couriers under customer fairness constraints when a new order is created and extract the pair $\langle state, action \rangle$ as the input of Actor. After feeding them into feature embedding layers respectively, we concatenate two features and feed the result vectors into hidden layers to calculate the long-term matching reward Q. Finally, given all possible $courier^k$-$order^i$ matching value $Q(s_t, a_t)$, policy π is parameterized as

$$\pi(a_t^{k=c}|s_t^i;\theta) = \frac{exp(Q(s_t^i, a_t^{k=c}))}{\sum_{c'=c_1}^{C} exp(Q(s_t^i, a_t^{k=c'}))} \tag{8}$$

where $a_t^{k=c}$ means dispatching $courier^c$ to deliver $order^i$ and θ is weight of Actor.

The second network called Critic judges whether the action selected by policy π is optimal or not and predicts the state-value function defined in Eq. 9.

$$V(s_t; w) = E[\sum_{k=0}^{\infty} \gamma^k r_{t+k}|s_t] \tag{9}$$

where w denote the parameters of the Critic.

Then, Actor are updated in the direction of $\nabla_\theta log\pi(a_t|s_t;\theta)A(a_t, s_t)$ where $A(a_t, s_t)$ is an advantage function ($k = 1$ in our experiment) which estimates the relative benefit of taking action a_t in state s_t and computed as Eq. 10.

$$A(a_t, s_t) = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k}) - V(s_t) \tag{10}$$

We update the parameters w of the value function $V(s; w)$ by minimizing the square loss of actual state value and estimated state value:

$$\arg\min_{w} \frac{1}{2}[r_{t+1} + \gamma V(s_{t+1}; w) - V(s_t; w)]^2 \tag{11}$$

4.3 Order Dispatch Based on A2CMF Model

Lastly, the training process of the **A2CMF** model will be introduced in detail.

1. **Get order information in one small period.** At each period, environment generate orders from real datasets containing information such as merchant location, customer location, price, promising delivery time and so on.
2. **Determine the dispatch range.** For each order, couriers in nearby areas have the chance to take this order. Our A2CMF model selects a proper number of candidate couriers within severe constraints of user satisfaction to avoid improper matches disturbing user fairness.
3. **Extract order and pending couriers' features.** After determining the pending couriers, using the environment simulator and feature extraction module, we extract features including spatial-temporal information, route planning information, existing order information and weather information.
4. **Find optimum courier and dispatch order to him/her.** The model receives each order-courier feature and sends them into the A2C network. Then Actor network calculates the order-courier matching value and recommends the optimum courier with Softmax function and Critic network works for advantage function by receiving reward and generating state value.
5. **Simulate and execute couriers' route plan.** After all orders in this small period have been arranged optimum couriers, our system would update the couriers' future route plans based on the environment simulator.
6. **Record feedback reward and update the A2C network.** Our system would record the reward from environment upon couriers finishing one order. And using the reward and state and action information, we can optimize the A2C network making its decisions closer and closer to the final fairness goals.

In a word, based on the environment simulator and feature extraction module, the A2CMF model can reasonably simulate the operation of couriers' movement and order dispatch in instant delivery. Meanwhile, the A2C network and reward based on multi-sided fairness can effectively guide the dispatch system to make a reasonable trade-off between system efficiency and multi-sided fairness.

5 Evaluation

5.1 Evaluation Methodology

Parameter Setting. We implement A2CMF and consider order dispatch in a map of 10 × 10 spatial grids with 167 time steps (i.e., 5 min as a time slot). At each time step, orders can only be dispatched to the courier whose customers don't wait over 8 min in peak hours and 5 min otherwise. And to guarantee the convergence, we set $\alpha = \beta = 0.3$ to balance the profit and fairness.

Baselines. To show the effectiveness of our system, we compare A2CMF with

- **GT(the ground truth)** is the order dispatch strategy extracted from the data simulated by the simulator in Eleme;
- **RD(random dispatch)** is the algorithm which always selects the courier with random strategy without considering muti-sided fairness;
- **SD2** is the shortest distance based dispatching method [11]. When one new order is created, it will be dispatched to the nearest courier in line with the customer is always right philosophy;
- **IDT** takes into account the influence of the new order added to a courier's route plan, using the increased delivery time based policy;
- **DDQN-as** utilizes a Double-DQN network to learn a order dispatch method in ride-sharing, with additional capability of carrying out action search [17];
- **XgD** is a Xgboost-based dispatch method in instant delivery [21]. Xgboost does ranking considering couriers' income, delivery distance and the increased journey time, and orders are dispatched to the courier ranking first.

Metrics. The evaluation metrics for capturing the fairness and efficiency are:

- **Total revenue (R_p):** From the platform's perspective, we investigate the efficiency of different order dispatch algorithms which is defined as the sum of each order's profit efficiency (Eq. 3).
- **Courier-side profit fairness ($Gini_c$):** We investigate income distributions among n couriers which is given by Gini Coefficient. The lower $Gini_c = \frac{\sum_{i=1}^{n}(2i-n-1)CE_i}{n\sum_{i=1}^{n}CE_i}$ (CE defined in Eq. 5), the better is courier fairness.
- **Merchant-side benefit gap (G_m):** To capture minimum benefit gap guarantee for all merchants, we compute the variance G_m of the merchants' mean product value (Eq. 6). The lower G_m, the smaller is merchant benefit gap.
- **Customer-side metrics:**
  - **Mean average waiting time (M_w):** Although A2CMF ensures Least Misery Fairness guarantee for customers, here we capture how effectively this reduces waiting time M_w on average in comparison to the baselines.
  - **Disparity in waiting time (D_w):** We also calculate the standard deviation of customer waiting time, that is, $D_w = \sqrt{\frac{1}{N_m}\sum_{k=1}^{N_m}(T(k) - M_w)^2}$. The lower the D_w, lesser is the disparity in waiting time.

5.2 Main Performance

Table 2 reports the overall results of our A2CMF model and all the compared baselines concerning our four metrics. As can be seen, A2CMF achieves the most well-rounded performance among all the baselines.

Table 2. Performance comparison

Method	R_p	$Gini_c$	G_m	M_w	D_w
GT	100%	100%	100%	100%	100%
RD	40.2%	66.4%	85.4%	–	–
SD2 [11]	67.4%	131.5%	130.4%	–	–
IDT	92.3%	78.2%	92.8%	81.6%	77.6%
DDQN-as [17]	113.1%	56.3%	111.4%	95.0%	82.3%
XgD [21]	42.8%	76.1%	119.4%	–	–
A2CMF	**137.2%**	**50.8%**	**80.1%**	**93.1%**	**81.8%**

Specifically, our method increases 21.3% of total revenue than DDQN-as which has the second-best performance. Figure 5 gives a visual confirmation that the performance is better in enhancing the total revenue when we choose A2CMF.

In Fig. 6, A2CMF outperforms other baselines in reducing the benefits gap among merchants, with a 19.9% decrease compared with GT. From Fig. 7 and Fig. 8, A2CMF's smallest radian of the Lorenz and the smallest $Gini_c$ illustrate its performance in helping couriers achieve more equitable income distribution.

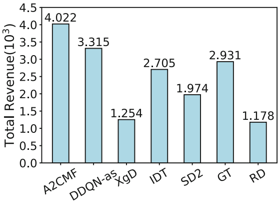


Fig. 5. Total revenue

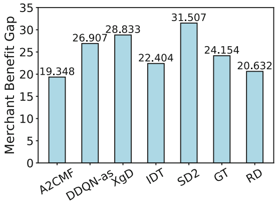


Fig. 6. Benefit gap G_m

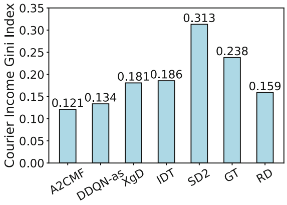


Fig. 7. Courier income *Gini*

In addition, we present the comparison in terms of customer-side metrics with GT, IDT, and DDQN-as. Figure 9 and Fig. 10 show that through our dispatch algorithm, we help users save 6.9% customer's mean waiting time than GT. Besides, the variance between the customers becomes smaller since we consider the fairness among them. Although IDT has a slight advantage over our A2CMF in minimizing the waiting time, it only focuses on the customer benefit without considering the potential revenue loss and unfair experience among merchants and couriers. As can be seen, A2CMF achieves the best overall performance by improving R_p by 21.3% and reducing ($Gini_c$, G_m) by (9.7%, 6.2%) compared to the second-best baselines.

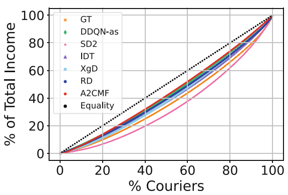


Fig. 8. Rider income lorenz

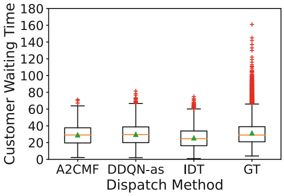


Fig. 9. User waiting time

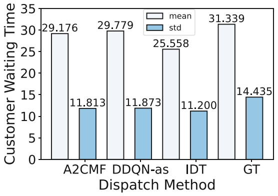


Fig. 10. M_w, D_w comparison

6 Related Works

6.1 Fairness in the Matching Mechanisms

Recently, instant delivery service plays an important role in online ordering and the potential unfairness problem comes into focus. Based on this scenario, researchers seek for matching mechanisms to guarantee fairness. Tom Sühr et al. propose a novel framework to think about not requiring every match to be fair, but rather distributing fairness over time, so they can achieve better overall benefit for all stakeholders [14]. On the other hand, Wang Guang et al. consider fairness as an optimization objective by improving overall efficiency and fairness [16]. And they once leverage greedy algorithm with Pareto improvement to solve multi-objective optimization [15].

6.2 Order Dispatch Mechanisms Based on Reinforcement Learning

Reinforcement learning is widely applied for sequential decision problems and particularly has been adopted for order dispatching in recent years. Ding, Yi et al. build a reinforcement learning model to learn the optimal order dispatching strategies, together with a profit model as the reward function [4]. Considering that instant delivery imposes a strict time deadline, Guo Baoshen et al. propose a Time-Constrained Actor-Critic Reinforcement learning based concurrent dispatch system to enhance long-term overall revenue and reduce overdue rate [5].

7 Conclusion

In this paper, we propose the first multi-sided fairness-aware dispatch system called A2CMF to improve the overall platform revenue and benefit fairness of all stakeholders. We first conduct a data-driven order dispatch pattern analysis, which shows the unfairness of dispatching problem and provides us insights into different notions of fairness among stakeholders. We then formulate the order dispatch as a Markov decision process and use the Advantage Actor-Critic (A2C) algorithm to tackle this problem. The performance of A2CMF is evaluated through a real-world dataset obtained from Eleme including over 1.15 million orders. Experimental results show that our fairness-aware A2CMF effectively increases the total platform revenue, improves customer service experience, and reduces the benefit gap between couriers and merchants by 9.7% and 6.2%.

Acknowledgements. The authors would like to thank the anonymous reviewers for their constructive and helpful feedback. This work was supported in part by Science and Technology Innovation 2030 - Major Project 2021ZD0114202.

References

1. Abdollahpouri, H., Burke, R.: Multi-stakeholder recommendation and its connection to multi-sided fairness. CoRR abs/1907.13158 (2019)
2. Biega, A.J., Gummadi, K.P., Weikum, G.: Equity of attention: amortizing individual fairness in rankings. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, pp. 405–414. Association for Computing Machinery, New York (2018)
3. Chakraborty, A., Patro, G.K., Ganguly, N., Gummadi, K.P., Loiseau, P.: Equality of voice: towards fair representation in crowdsourced top-k recommendations, FAT* 2019, pp. 129–138. Association for Computing Machinery, New York (2019)
4. Ding, Y., et al.: A city-wide crowdsourcing delivery system with reinforcement learning. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **5**(3), 1–22 (2021)
5. Guo, B., et al.: Concurrent order dispatch for instant delivery with time-constrained actor-critic reinforcement learning. In: 2021 IEEE Real-Time Systems Symposium (RTSS), pp. 176–187 (2021)
6. Lei, H., Zhao, Y., Cai, L.: Multi-objective optimization for guaranteed delivery in video service platform. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2020, pp. 3017–3025 (2020)
7. Li, M., et al.: Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In: The World Wide Web Conference, WWW 2019, pp. 983–994 (2019)
8. Li, Y., Zheng, Y., Yang, Q.: Efficient and effective express via contextual cooperative reinforcement learning. In: KDD 2019 (2019)
9. Li, Y., Chen, H., Fu, Z., Ge, Y., Zhang, Y.: User-oriented fairness in recommendation. In: Proceedings of the Web Conference 2021, WWW 2021, pp. 624–632 (2021)
10. Lin, K., Zhao, R., Xu, Z., Zhou, J.: Efficient large-scale fleet management via multi-agent deep reinforcement learning. In: KDD 2018 (2018)
11. McCann, J., Chatley, R.: Fleet management in on-demand transportation networks: using a greedy approach (2018)
12. Patro, G.K., Biswas, A., Ganguly, N., Gummadi, K.P., Chakraborty, A.: Fairrec: two-sided fairness for personalized recommendations in two-sided platforms. In: Proceedings of the Web Conference 2020, WWW 2020, pp. 1194–1204 (2020)
13. Singh, A., Joachims, T.: Fairness of exposure in rankings. In: KDD 2018 (2018)
14. Sühr, T., Biega, A.J., Zehlike, M., Gummadi, K.P., Chakraborty, A.: Two-sided fairness for repeated matchings in two-sided markets: a case study of a ride-hailing platform. In: The 25th ACM SIGKDD International Conference, KDD 2019 (2019)
15. Wang, G., Zhang, Y., Fang, Z., Wang, S., Zhang, F., Zhang, D.: Faircharge: a data-driven fairness-aware charging recommendation system for large-scale electric taxi fleets. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **4**(1), 1–25 (2020)

16. Wang, G., Zhong, S., Wang, S., Miao, F., Dong, Z., Zhang, D.: Data-driven fairness-aware vehicle displacement for large-scale electric taxi fleets. In: ICDE 2021 (2021)
17. Wang, Z., Qin, Z., Tang, X., Ye, J., Zhu, H.: Deep reinforcement learning with knowledge transfer for online rides order dispatching. In: ICDM 2018 (2018)
18. Xiao, L., Min, Z., Yongfeng, Z., Zhaoquan, G., Yiqun, L., Shaoping, M.: Fairness-aware group recommendation with pareto-efficiency. In: RecSys 2017 (2017)
19. Xu, Z., et al.: Large-scale order dispatch in on-demand ride-hailing platforms: a learning and planning approach. In: KDD 2018 (2018)
20. Zhang, L., et al.: A taxi order dispatch model based on combinatorial optimization. In: The 23rd ACM SIGKDD International Conference, KDD 2017, pp. 2151–2159 (2017)
21. Zhang, Y., et al.: Route prediction for instant delivery. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **3**(3), 1–25 (2019)
22. Zhou, M., et al.: Multi-agent reinforcement learning for order-dispatching via order-vehicle distribution matching. In: CIKM 2019 (2019)

HeadTracker: Fine-Grained Head Orientation Tracking System Based on Headphones

Jinpeng Song[1], Haipeng Dai[1](✉), Shuyu Shi[1], Lei Wang[2], Haoran Wan[1], Zhizheng Yang[1], Fu Xiao[3](✉), and Guihai Chen[1](✉)

[1] Department of Computer Science and Technology,
Nanjing University, Nanjing, China
{jinpengsong,wanhr,yzz}@smail.nju.edu.cn,
{haipengdai,ssy,gchen}@nju.edu.cn

[2] Department of Computer Science and Technology, Peking University, Bejing, China
wang_l@pku.edu.can

[3] School of Computer Science, Nanjing University of Posts and Telecommunications,
Nanjing, China
xiaof@njupt.edu.cn

Abstract. Head orientation tracking has many potential applications in various fields, *e.g.*, online courses, online meetings, and somatosensory games. Undoubtedly, with the information of the user's head orientation, these applications will have more opportunities to enhance performance and provide better user experience. However, reviewing existing works regarding head tracking, the CV-based solutions have limited tracking angle range and privacy issues and the IMU-based solutions have accumulated errors. None of these methods provide accurate and stable user head orientation. In this paper, we propose HeadTracker, a fine-grained 3D head orientation tracking system based on a single headphone. Headtracker achieves high-precision head orientation tracking by installing ultrasonic transmitters on an ordinary headphone and deploying ultrasonic receivers in the environment. We conducted experiments to evaluate the performance of HeadTracker in the real use environment, and the experimental results show that the system can achieve an average error of 6° in the 3D head orientation tracking. To the best of our knowledge, HeadTracker is the first system to use head-mounted ultrasound device to achieve 3D head orientation tracking and achieves the state-of-the-art in this category.

Keywords: Head orientation · Wireless sensing · Ultrasonic signal · Wearable devices

This work was supported in part by the National Natural Science Foundation of China under Grant 61872178, 61832005, 62102006, in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University, and in part by the Jiangsu High-level Innovation and Entrepreneurship (Shuangchuang) Program.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 317–329, 2022.
https://doi.org/10.1007/978-3-031-19214-2_26

1 Introduction

User tracking, which refers to locating users in real time, has become the focus of many research work in recent years [2,5,6,9,12]. Nevertheless, most user tracking systems only focus on the user's location but ignore the user's head orientation, which can reveal important and valuable information, such as the user's attention and intent. If the user's accurate 3D head orientation can be obtained in real time, we can envision and expect its wide usage in many scenarios. For example, in online courses scenarios, we can know where the students' attention is through their head orientation. In addition, it also has promising usage in motion-sensing games as an alternative to mouse and keyboard. Besides, in driving scenarios, we can implement many intelligent driving applications such as estimating the driver's intention based on his head orientation.

According to our survey, most of the existing head orientation tracking work is based on computer vision [1,10,13]. These CV-based solutions can only achieve a small range of head tracking due to the narrow angle of view of camera, and they are severely affected by environmental factors such as light. Moreover, the use of cameras will bring certain privacy risks. There are also some IMU-based head tracking solutions [6], but such solutions are limited by the cumulative error of the six-axis IMU and need to be continuously calibrated in use. Although the nine-axis IMU addresses the cumulative error problem to a certain extent [4,8], it is seriously affected by the external magnetic field [3]. Most importantly, both of the CV-based solutions and the IMU-based solutions obtain head orientation in their own internal coordinate system, which is difficult to be converted to the world coordinate system for interaction with other devices. Besides, there are some solutions based on microphone arrays [15,16], but the accuracy of these solutions are relatively low.

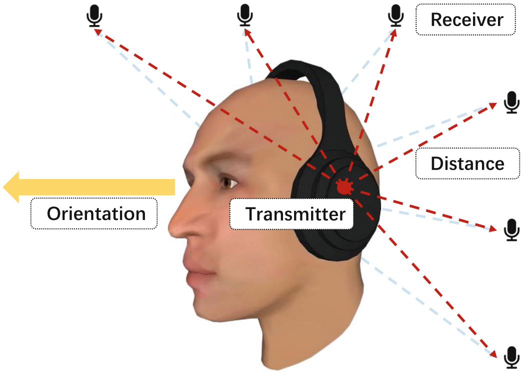


Fig. 1. HeadTracker

In this paper, we propose HeadTracker (Fig. 1), a fine-grained 3D head orientation tracking system based on headphones. Compared with other existing work, HeadTracker significantly improves the accuracy of head orientation tracking. On the hardware side, we add two ultrasonic transmitters to both sides of an ordinary headphone and deploy some ultrasonic receivers in the environment to complete the positioning of the headphone. On the software side, we design several algorithms to calculate the head orientation and keep the system running smoothly. Specifically, the contributions of our paper are as follows:

1) We use the Zadoff-Chu sequence as the baseband signal and modulate it to the ultrasound band as our transmitting signal. We demodulate it on the receiving side, and decompose different paths from the accurate CIR. On this basis, we design a frequency division multiplexing method to realize the simultaneous positioning of two transmitters.
2) To solve the problem that signal direct path is easily blocked, we borrow the idea of GPS satellite positioning systems [7]. That is, we deploy multiple receivers in the environment and propose a receiver selection algorithm based on signal quality to accomplish positioning.
3) We use neural network to design a special head orientation tracking algorithm based on head movement recognition, which enables approximately 6-DoF head orientation tracking using only two coordinates on the head.

The remaining of this paper is organized as follows. In Sect. 2, we describe the system design and processing flow of our proposed approach HeadTracker in detail. Then, we introduce the deployment of the system and conduct a large number of experiments to evaluate the effectiveness of it in Sect. 3. Finally, we conclude this paper in Sect. 4.

2 System Design

In this section, we introduce the technical details of the HeadTracker. The system mainly consists of four modules: *signal process, headphone positioning, movement recognition, and orientation calculation*, as depicted in Fig. 2.

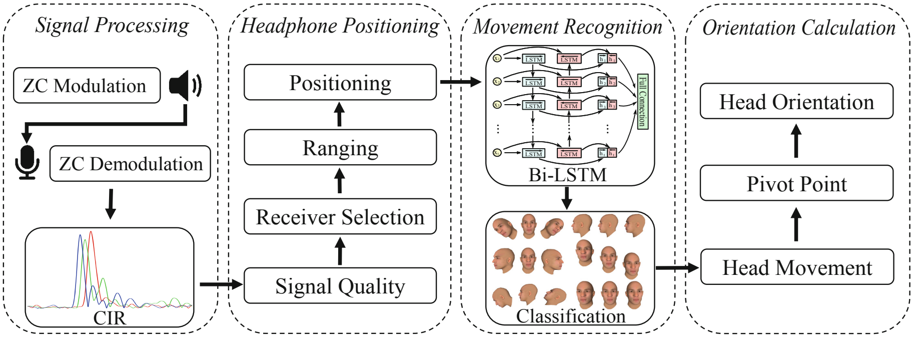


Fig. 2. Overview of HeadTracker

2.1 Signal Progressing

Signal Design. We use ZC sequences as baseband signal as ZC is a kind of CAZAC (Constant Amplitude Zero AutoCorrelation waveform), which means ZC sequences have ideal auto-correlation properties [11,14]. Compared with the common CW (Continuous Wave) signal, ZC signal can separate paths at different distances and reduce the influence of multipath. And compared with the FMCW (Frequency Modulated Continuous Wave) signal, ZC signal has better range resolution. We modulate the ZC sequence by a sinusoid carrier at the transmitter, and the mathematical form of the ZC sequence is

$$ZC[n] = e^{-j\frac{\pi u n\left(n+c_f+2q\right)}{N_{ZC}}}, \tag{1}$$

where N_{ZC} is the length of ZC sequence, the value range of n is $0 \leq n < N_{ZC}$, and c_f takes 0 or 1 as the remainder of N_{ZC} modulo 2. The ZC sequence contains two integer parameters q and u. Generally, q is set to 0, and the ZC sequence degenerates into Chu sequence. Moreover, u is in the range $[0, N_{ZC}]$, and it is relatively prime to N_{ZC}.

ZC Modulation and Demodulation. In the process of signal modulation and demodulation, we use an OFDM-based interval interpolation method, which makes it possible to modulate two different ZC sequences to the same center frequency. Similarly, in the demodulation process of the received signal, we use the frequency domain interval sampling method to separate the two ZC sequences from the same received signal.

We know that according to the characteristics of the ZC sequence, the auto-correlation result of the ZC sequence is non-zero only at $t = 0$, which ideally will be a Dirac impulse function $\delta(t)$ and is a *sinc* function practically due to limited bandwidth. Because the received signal is composed of multiple transmitted signals with different time delay versions through multiple different paths, the result of cross-correlation between the transmitted signal and the received signal is $h(t)$, which is a combination of $\delta(t - \tau_i)$ signals with different time delays τ_i:

$$h(t) = \sum_{i=1}^{P} A_i e^{-j\phi_i(t)} \delta(t - \tau_i), \tag{2}$$

where P is the number of paths, A_i is the signal strength of signal path i, and ϕ_i is the phase offset of the signal on path i. And we use Dirac function here for convenience. We can express the channel impulse response (CIR) as $h(t)$, as shown in Fig. 3.

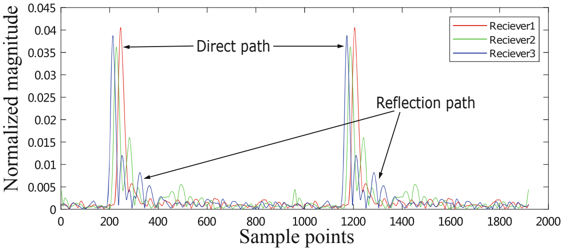


Fig. 3. CIR

Ranging. The abscissa of the CIR corresponds to the delay, while the ordinate corresponds to the cross-correlation value between transmitted signal $ZC\left[t\right]$ and $ZC\left[t-\tau_i\right]$, which is transmitted signal after a certain delay τ_i. The larger the cross-correlation value, the stronger the delayed signal. Generally, the path corresponding to the highest peak of the CIR is the direct path in the case that it is not blocked. So we can calculate the direct path using the following equation:

$$d = \underset{1<=i<=\frac{L}{2}}{\arg\max}\ CIR\left[i\right]\frac{c}{f_s}, \tag{3}$$

where L is the length of CIR. Therefore, after ranging, we can obtain the straight-line distance between each transmitter and each receiver, which makes preparations for our subsequent positioning work.

2.2 Headphone Positioning

Receiver Selection. As is well known, a major challenge for ultrasonic positioning in practice is that the direct path of sound waves between transmitter and receiver can be blocked frequently. To solve this challenge, we refer to the idea of satellite positioning systems like GPS, which is to deploy multiple satellites in orbit to achieve full coverage of the ground. Similarly, we can deploy multiple ultrasonic receivers in the environment so that no matter how the user's head rotates and moves, the direct path between each transmitter and at least three receivers is not blocked. To this end, we propose a receiver selection algorithm by which the system will select the most suitable three receivers to positioning the transmitter each time. Firstly, we propose an indicator named SNR_{los}, which is used to evaluate the signal quality between receivers and transmitters. Formally, SNR_{los} is defined as the ratio of the amplitude of the highest peak to the average of all other peaks' amplitude in the CIR.

$$SNR_{los} = \frac{\max CIR\left[i\right]}{\sum_{i=1}^{\frac{L}{2}} CIR\left[i\right] - \max CIR\left[i\right]}. \tag{4}$$

Positioning. After the receiver selection, each transmitter has found the three most suitable receivers. According to the triangulation method, knowing the distance between a certain point and three known anchor points, a ternary quadratic equation can be established to calculate this point's coordinates:

$$f = \begin{cases} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 - d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 - d_2^2 \\ (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 - d_3^2, \end{cases} \tag{5}$$

where (x, y, z) is the position of the transmitter, (x_i, y_i, z_i) is the position of selected receiver, and d_i is the distance between the transmitter and receiver measured by ultrasonic ranging. We use Newton's iterative method to solve this ternary quadratic equation system. To improve the calculation speed, we set the initial iteration value of each positioning as the result of the last positioning, which can greatly reduce the number of iterations. Generally, each positioning can be completed only after three or four iterations in this way. After positioning, we can obtain the trajectory data of the headphone, which can be used to identify the current movement of the head.

2.3 Movement Recognition

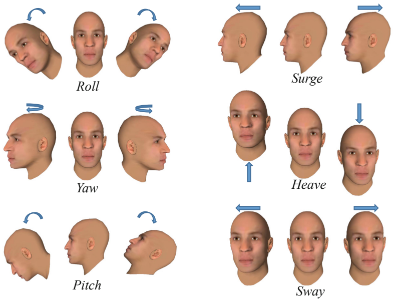


Fig. 4. Head movements

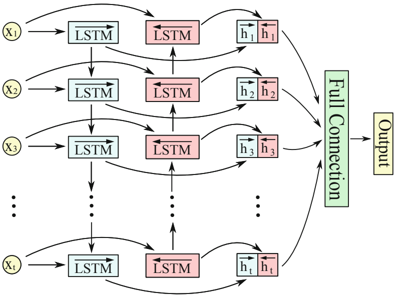


Fig. 5. Bi-LSTM

Movement Definition. We know that a rigid body has six degrees of freedom in three-dimensional space. We use the definition in the field of navigation to describe these movements, which are the three translational movements (surge, sway, and heave) and the three rotational movements (roll, pitch, and yaw), as shown in Fig. 4. Specifically, surge, heave, and sway are the translation movements along the x-axis, z-axis, and y-axis, respectively; roll, yaw, and pitch are the rotation movements around the x-axis, z-axis, and y-axis, respectively.

We ignore overly complicated head movements here as we believe that the six basic movements account for the vast majority in our daily life, while other complex movements are relatively rare. Besides, adding other uncommon movements will increase the complexity of the classification model and reduce the overall classification performance, which we think is not worth the gain.

Classification Model. Head movements recognition is a classification task with the data of headphone trajectory. Since trajectory is a kind of time series data, we adopt Bi-LSTM as the classification model to complete this classification task. Bi-LSTM is a special kind of recurrent neural network that has a good representation ability for the time series data. Its excellent performance has been proven in many fields such as speech recognition. Bi-LSTM combines the forward LSTM with the backward LSTM as shown in Fig. 5. Therefore, Bi-LSTM can make better use from the information of the subsequent data compared with traditional LSTM.

In this step, we use the headphone trajectory as training data to train a classification model, which can be used to identify the ongoing head movement. After obtaining the head movement, we can calculate the head orientation according to some head movement rules, which is the next step of our system.

2.4 Orientation Calculation

Clearly, to determine the posture of an object in three dimensions, at least the coordinates of three different points need to be known. The posture of the rigid body is not unique with only two coordinates, because it can rotate around the axis formed by the two points. But after headphone positioning we can only obtain two points on the head, now the problem is ***how to estimate the head posture based on the positions of only two points?***

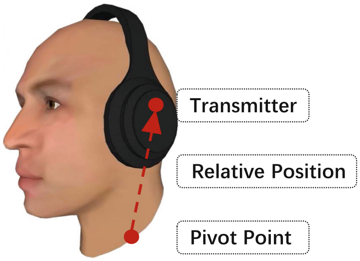


Fig. 6. Pivot point

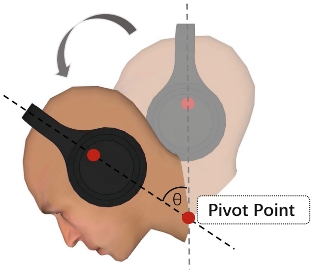


Fig. 7. Rotation

In fact, the head is not an object that can move freely in three-dimensional space, which is limited by its connection to the body. By observing and analyzing, we find some rules of head movement. That is, the movement of the human head are carried out around a point in the neck, we call it the **pivot point** (Fig. 6). The position change of the pivot point is closely related to the head's movements.

When the head is only rotating without moving, the absolute position of the pivot point is almost unchanged (Fig. 7). When the head is only moving without rotating, the relative position of the pivot point and the headphone remains almost unchanged. Therefore, these rules give us the possibility to determine the position of the pivot point by the head's movement. The relative position between the pivot point and the headphone is unchanged for a person, which is determined by the bones of the head and neck. So we only need to initialize the pivot point once at the beginning and then we can update it in real time according to the movement of the head.

The pivot point position updating formula is expressed as follows:

$$P_{pivot} = \begin{cases} \frac{P_{left}+P_{right}}{2} - V_{relative} & M \in \{surge, sway, heave\} \\ P_{pre} & M \in \{roll, pitch, yaw, static\} \end{cases}, \quad (6)$$

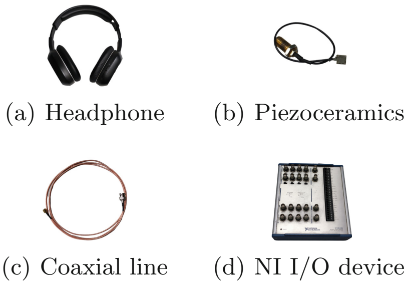
where P_{pivot} is the position of the current required pivot point, P_{left} and P_{right} are the positions of two transmitters, $V_{relative}$ is the vector between the midpoint of the two transmitters and the pivot point, P_{pre} is the position of the pivot point in the previous frame, and M is the ongoing movement of the head.

We now have the coordinates of the three points on the head in total, *i.e.*, the pivot point and two transmitters. Sequentially, we can calculate the head orientation according to the following formula:

$$V_{orientation} = (P_{right} - P_{pivot}) \times (P_{left} - P_{pivot}) . \quad (7)$$

3 Implementation and Evaluation

3.1 Implementation


(a) Headphone (b) Piezoceramics

(c) Coaxial line (d) NI I/O device

Fig. 8. Hardware

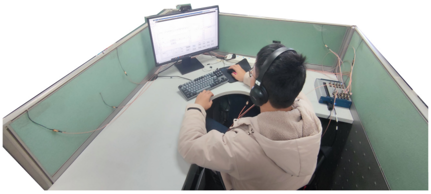
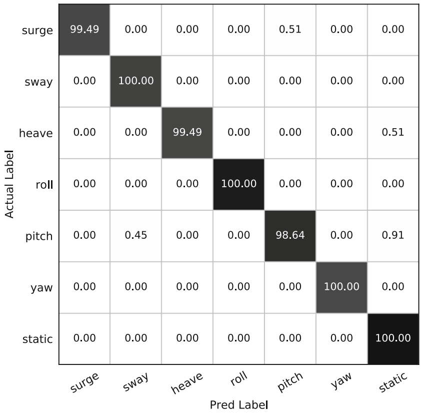


Fig. 9. Experimental scene

Figure 8 shows the devices used in our experiment. We choose piezoelectric ceramics as the transmitter and receiver of ultrasonic waves. We install the two receivers on both sides of the headphone and install the receivers in the environment. We use Murata MA40H1 piezoelectric ceramics as sound sources for transmitting and receiving ultrasonic waves. The I/O device we use is USB-6356 produced by National Instruments, which can support up to 2 analog signal outputs and 8 analog signal inputs. These piezoelectric ceramics are connected to I/O device through coaxial cables and the experimental scene is shown in Fig. 9. In the part of software, we use MATLAB to drive the device for signal acquisition and data processing. In the experiment, we set the center frequency of ZC signal to 40 KHz with 96 KHz sampling rate, which is far beyond the hearing range of human ears. As for the cost of this system, we admit that the price will be higher than other solutions, such as CV and Bluetooth. We are studying how to complete this task with the help of loudspeakers and microphones commonly used in life to reduce costs.

3.2 Performance of Head Movement Recognition

We collect a data set of more than 7000 trajectories information to evaluate the performance of the head movement recognition module. There are about 1000 trajectories for each type of movements, each of which is a two-second coordinate sequence of two transmitters. Specifically, 80% of the data in the dataset is used to train the model while the remaining 20% is used for testing. According to the confusion matrix in Fig. 10, the average classification accuracy on the dataset is more than 99%. For a single head movement, the one with the lowest accuracy is pitch, which achieves the accuracy of 98.64%, and the one with the highest accuracy is sway, roll, yaw and static movements, which reach 100%. It can be seen that the classification accuracy of the two movements (*i.e.*, surge and pitch) are relatively low compared with others. This is also in line with our intuition, because surge is the forward and backward translation of the head, and pitch is the forward and backward rotation of the head. The two movements are very similar when the movement range is not large, so they are easy to be confused.



	surge	sway	heave	roll	pitch	yaw	static
surge	99.49	0.00	0.00	0.00	0.51	0.00	0.00
sway	0.00	100.00	0.00	0.00	0.00	0.00	0.00
heave	0.00	0.00	99.49	0.00	0.00	0.00	0.51
roll	0.00	0.00	0.00	100.00	0.00	0.00	0.00
pitch	0.00	0.45	0.00	0.00	98.64	0.00	0.91
yaw	0.00	0.00	0.00	0.00	0.00	100.00	0.00
static	0.00	0.00	0.00	0.00	0.00	0.00	100.00

Fig. 10. Confusion matrix of head movement recognition

As for the reason why the classification result is so accurate, we believe that it can be attributed to the strong representation ability of Bi-LSTM for time series data, the sufficient training data and the simple classification task.

3.3 Performance of Head Orientation Tracking

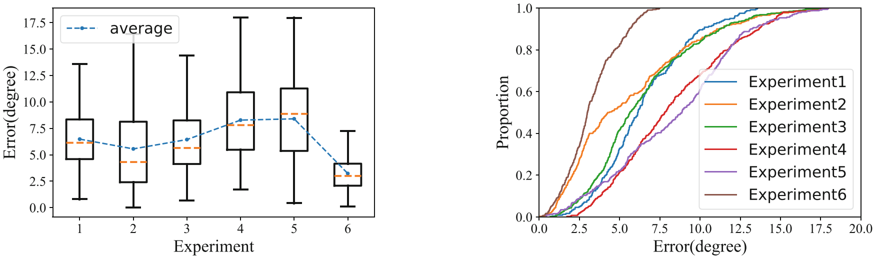


Fig. 11. Results of head orientation tracking

To get the groundtruth the head is facing, we use a nine-axis IMU (including accelerometer, gyroscope, and magnetometer). We attach this IMU to the same headphone together with the HeadTracker system. This IMU can feed back the three-axis angle changes of the head to us in real time. Based on this, we can calculate the orientation of the head as the groundtruth. We then compare the groundtruth with the HeadTracker measurements to evaluate the performance of the system. However, as we mentioned earlier, the IMU has the problem of cumulative error, which can adversely affect the experimental results. To reduce the influence of the cumulative error of the IMU, we try to shorten the duration of each experiment, which is about 20 s to 60 s. During each experiment, the participants are first told what to do and then put on the equipment with the help of the experimenter. The participants will repeat the following actions during the experiment: surge, sway, heave, roll, pitch, and yaw. During the experiment, we record the groundtruth of the IMU and the measurements of the HeadTracker in real time at ten frames per second.

We conduct a total of 6 groups of experiments. The system samples the head's orientation at a frequency 10 Hz during the volunteer's rotation. Figure 11(a) shows the error of the 6 groups. From the figure, it can be seen that the median error of each experiment is between 3° and 7°, and the maximum error is about 17.5°. According to the calculation, the average error of these 6 groups is about 6°. Figure 11(b) shows the CDF of the errors of all groups, where the 50% error of data is less than 7° and the 90% error of data is less than 12°.

3.4 Impact of Speed

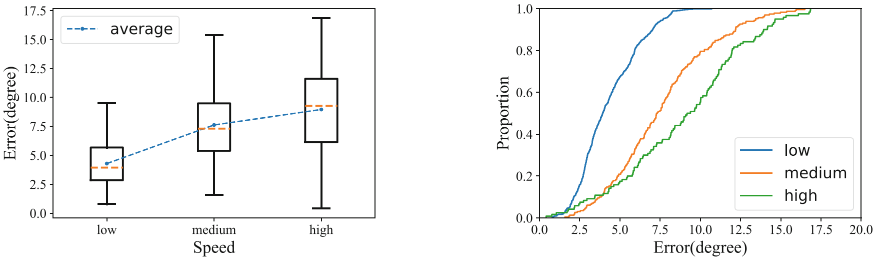


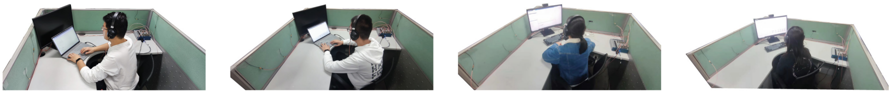
Fig. 12. Results for different rotation speeds

We also conduct experiments at different rotation speeds. First of all, according to the habit of human head rotation, we divide head's rotation speed into low speed, medium speed, and high speed. Low speed means that the rotation speed is about 1.5 degrees per second, medium speed is about 3 degrees per second, and high speed is about 9 degrees per second. We let the volunteer rotate the head at different speeds, and then evaluate the head orientation tracking performance. Figure 12 shows the experimental results at different rotation speeds.

It can be seen from Fig. 12(a) that the error of low-speed rotation is smaller than that of medium-speed rotation, and the error of medium-speed rotation is smaller than that of high-speed rotation. Regardless of the average value, median, maximum value, and other indicators for comparison, the result of low-speed rotation is almost always the best. Figure 12(b) also shows that the error of low-speed rotation is the smallest, achieving a result that the 50% error of the data is less than 5° and the 90% error of the data is less than 12°. The results are in line with our intuition, because the lower the rotation speed, the more stable the head is, the easier it is to control the head orientation.

3.5 Impact of Participants

Considering that the performance of our HeadTracker system is closely related to the user's physiological characteristics, especially the size and shape of the bones in the head and neck, different users may bring different experimental results. To evaluate the robustness of our system to users, we invite 10 participants (7 males, 3 females) to conduct the experiment (Fig. 13). These volunteers range in age from 20 to 25. We let each participant wear the equipment to conduct the same experiment and evaluate the results of these experiments. The results of all experiments are shown in Fig. 14. Because of the different physiological structure and head movement habits among people, there are some differences between the results from different participants as shown in Fig. 14(a). Among them, participant M5 has the largest average error (about 5°), while participant



(a) Male1 (b) Male2 (c) Female1 (d) Female2

Fig. 13. Different participants

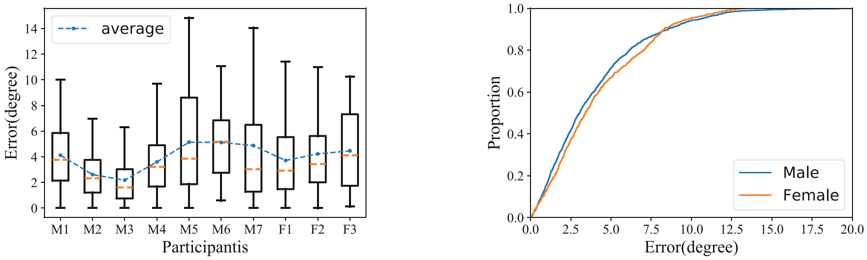


Fig. 14. Results for different participants

M3 has the smallest average error (about 2.5°). Moreover, the average error of all the participants' data is about 4°, which is basically the same as the error we measured above. These experiments show that HeadTracker is robust to different participants. In addition, we count the head orientation errors of male and female participants and draw the CDF of them in Fig. 14(b). We can see that the two curves basically overlap, which prove that the results are almost not affected by gender.

4 Conclusion

Users' head orientation provides valuable information to various fields such as online courses, online conferences, and somatosensory games. To effectively obtain and utilize this information, we propose HeadTracker in this paper, which is a fine-grained 3D head orientation tracking system based on a headphone. To achieve high-precision tracking, we first install the ultrasonic transmitters on the headphone and deploy the ultrasonic receivers in the environment to realize the positioning of the user's headphone. Then, we use the trajectory of the headphone and Bi-LSTM to complete the recognition of the user's head movement. Finally, we calculate the real-time position of the pivot point based on the head movement and then calculate the head orientation. Our experimental results show that the average error of head orientation tracking is about 6° in real environment, which is the best performance known at present and indicates that our system has great development potential and application prospects.

References

1. Cordea, M.D., Petriu, E.M., Georganos, N., Petriu, D.C., Whalen, T.E.: Real-time 2(1/2)-D head pose recovery for model-based video-coding. IEEE Trans. Instrum. Meas. **50**(4), 1007–1013 (2001)
2. Correa, A., Munoz Diaz, E., Bousdar Ahmed, D., Morell, A., Lopez Vicario, J.: Advanced pedestrian positioning system to smartphones and smartwatches. Sensors **16**(11), 1903 (2016)
3. Das, S.S.: Simple, inexpensive, accurate calibration of 9 axis inertial motion unit. In: 2019 28th IEEE International Conference on Robot and Human Interactive Communication, pp. 1–6. IEEE (2019)
4. Euston, M., Coote, P., Mahony, R., Kim, J., Hamel, T.: A complementary filter for attitude estimation of a fixed-wing UAV. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 340–345. IEEE (2008)
5. Jimenez, A.R., Seco, F., Prieto, C., Guevara, J.: A comparison of pedestrian dead-reckoning algorithms using a low-cost mems IMU. In: 2009 IEEE International Symposium on Intelligent Signal Processing, pp. 37–42. IEEE (2009)
6. Kang, W., Han, Y.: SmartPDR: smartphone-based pedestrian dead reckoning for indoor localization. Sensors **15**(5), 2906–2916 (2014)
7. Kaplan, E.D., Hegarty, C.: Understanding GPS/GNSS: Principles and Applications. Artech House (2017)
8. Madgwick, S.O., Harrison, A.J., Vaidyanathan, R.: Estimation of imu and marg orientation using a gradient descent algorithm. In: 2011 IEEE International Conference on Rehabilitation Robotics, pp. 1–7. IEEE (2011)
9. Roy, N., Wang, H., Roy Choudhury, R.: I am a smartphone and i can tell my user's walking direction. In: Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, pp. 329–342 (2014)
10. Tong, Y., Wang, Y., Zhu, Z., Ji, Q.: Robust facial feature tracking under varying face pose and facial expression. Pattern Recogn. **40**(11), 3195–3208 (2007)
11. Wan, H., Shi, S., Cao, W., Wang, W., Chen, G.: Resptracker: multi-user room-scale respiration tracking with commercial acoustic devices. In: Proceedings of IEEE INFOCOM Conference on Computer Communications, pp. 1–10. IEEE (2021)
12. Wang, H., Sen, S., Elgohary, A., Farid, M., Youssef, M., Choudhury, R.R.: No need to war-drive: unsupervised indoor localization. In: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, pp. 197–210 (2012)
13. Wang, J.G., Sung, E.: EM enhancement of 3D head pose estimated by point at infinity. Image Vis. Comput. **25**(12), 1864–1874 (2007)
14. Wang, L., et al.: Watching your phone's back: gesture recognition by sensing acoustical structure-borne propagation. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **5**(2), 1–26 (2021)
15. Yang, J.J., Banerjee, G., Gupta, V., Lam, M.S., Landay, J.A.: Soundr: head position and orientation prediction using a microphone array. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–12. Association for Computing Machinery (2020)
16. Yang, Q., Zheng, Y.: Model-based head orientation estimation for smart devices. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **5**(3), 1–24 (2021)

Kalman Filter Algorithm Based on Sheep Herding Optimization

Peng Wang[1], Junyi Zhang[2], Yuqi Zheng[3](✉), Xiaohu Li[3], and Yixin Li[3]

[1] The 54th Research Institute of China Electronics Technology Group Corporation,
Hebei 050050, China

[2] Hebei Key Laboratory of Electromagnetic Spectrum Cognition and Control,
Hebei 050050, China

[3] Beijing University of Posts and Telecommunications, Beijing 100876, China
yqzheng@bupt.edu.cn

Abstract. When dealing with the real track, the environment is often an unpredictable factor, so filtering is very important. We can use the filter to eliminate the influence of noise as much as possible. Kalman filter is one of them. In this work, we proposed a new Particle Swarm Optimization algorithm, called the Sheep Herding Optimization algorithm, which can obtain higher quality solutions with faster convergence speed and better stability. Besides, in order to improve the performance of Kalman filter, we apply the Sheep Herding Optimization algorithm to the filter. The improved Kalman filter can fuse and predict the track, and has higher computational performance and smaller error.

Keywords: Kalman filter · PSO · KSP · Filtering · Track fusion

1 Introduction

Facing the difficulty of multi-source electromagnetic sensing information estimation and fusion in multi-sensor networks, aiming at typical application scenarios, research contents such as target track generation and tracking under low positioning accuracy, target track fusion under multi-precision hybrid conditions, multi-source information fusion based on evidence theory is carried out. New technologies and methods for multi-source electromagnetic sensing data estimation and fusion are established, the prototype software is developed to provide basic support for the intelligent processing of multi-source electromagnetic sensing data. In real life, the environment is very complex and unpredictable, the data we obtained are often accompanied by noise interference, so signal processing is very important, and filtering is one of the key technologies. Kalman filter stands out for its excellent estimation ability, it is different from ordinary filters. Kalman filter algorithm reduces the storage and calculation to a great extent. However, the traditional Kalman filter algorithm needs to predict the statistical characteristics of noise in advance, but this is not realistic in general. Kalman filter algorithm is limited, so we need to introduce a new particle algorithm to optimize and improve it. Particle swarm optimization algorithm has always been an optimal method in optimization. The

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 330–342, 2022.
https://doi.org/10.1007/978-3-031-19214-2_27

individual function of particles is very simple and easy to realize, and the information exchange between particles is not complex.

Therefore, in our work, we adopt the Sheep Herding optimization Algorithm, which is proved to be more efficient than ordinary PSO algorithms. Besides, the optimization method is introduced into Kalman filter algorithm, making it perform better. Using particle algorithm to improve Kalman filter algorithm is of great research significance and research value.

2 Related Work

2.1 Kalman Filter Algorithm

In practice, it is often impossible to directly obtain the real value of the required state and variable, because the radar system will have the problem of random jam when detecting the target, resulting in the random noise in the observed signal; It is very difficult to separate the motion state of the target, and Kalman filter is a powerful tool to reduce these noises. Kalman filter needs current measurements and predictions from previous sampling cycles to estimate state. It is important to note that when the filter is used to solve the problem of state estimation for a moving target or entity, its measurement equations are linear. Scientists have done a lot of research, hoping to modify the KF algorithm to fit to the actual situation. Dr. Schmidt [1] proposed EKF Kalman Schmidt filter, this filter was developed to reduce the dimensionality of the state estimation without overlooking the effects of the additional state on the calculation of the covariance matrix and Kalman gains. One of the main advantages offered by this modification is reducing computational requirements. Bonnabel et al. [2] proposed a new version of the Extended Kalman Filter, called IEKF, which is proposed for nonlinear systems possessing symmetries. Instead of using a linear correction term based on a linear output error, it uses a geometrically adapted correction term based on an invariant output error. Pi et al. [3] proposed an adaptive extended Kalman filter (AEKF) algorithm to resolve the problem of the error accumulation. It takes the Taylor series of sampling time in AEKF and use the Sage-Husa time-varying noise estimator to estimate observation noise in real time. Hesch et al. [4] proposed OC-EKF (Observability Constrained Extended Kalman Filter), it removes false information along unobservable directions of the estimator. The algorithm improves the precision and consistency of the inertial navigation system.

The first application of the Kalman filter and its extension was conducted in the guided navigation field. The KF and its variants have been used in a wide range of tasks [5]. The paper proposed a pedestrian navigation algorithm based on colored noise improved Kalman filter, and the experimental result showed that the proposed algorithm would have higher positioning accuracy than the pedestrian navigation algorithm using the white noise model. Li et al. [6] researched and simulated how to apply KF to the radar target tracking system. The algorithm can be used in some single-target tracking systems directly, or combined with other algorithms for multi-target tracking systems. What's more, Assaf et al. [7] studied Kalman Filter (KF) based techniques for tracking ships using Global Positioning System (GPS) data. The absence and presence problem of a ship is handled by an applying KF theory to analyze GPS coordinates and compare current marine vessel routes to previously recorded ones.

2.2 Particle Swarm Optimization

For the complex optimization problems in various fields in real life, many researchers choose to seek solutions from the models of nature. The intelligent behavior of social animals through simple cooperation without centralized control is called swarm intelligence. The agents in a Swarm Intelligence system follow very simple rules [8]. There is no centralized control structure dictating how individual agents should behave. The agents' real behaviors are local, and to a certain degree random; however, interactions between such agents lead to the emergence of "intelligent" global behavior, which is unknown to the individual agents. Well- known examples of Swarm Intelligence include ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling. It has the characteristics of self-organization, robustness, flexibility and low consumption. Besides, it can still effectively process a large amount of data due to its characteristics of distribution and parallelism. So far, particle swarm optimization algorithm is one of the typical algorithms.

Particle swarm optimization is a part of evolutionary computing. It finds the optimal solution by cooperating and sharing information among each individual in the group. It has few parameters need to be adjusted, and is widely used. PSO performs searching via a swarm of particles that updates from iteration to iteration [8]. To seek the optimal solution, each particle moves in the direction to its previously best (pbest) position and the global best (gbest) position in the swarm, so that most particles can gather near the global optimal solution to solve the problem. Based on this, many researchers made a lot of efforts to improve the performance of particle swarm optimization algorithm. Yang et al. [9] proposed an improved PSO algorithm called Elitist promotion strategy, a stored information recombination method. When criteria are met, the personal best solutions of particles are used to reconstruct the new individuals through specified operators. In order to improve the global search ability of the algorithm, the new generated individuals with better fitness values are selected as the new personal best solutions and global best solution. Dong et al. [10] proposed a new approach, introduced the adaptive elite mutation and nonlinear inertia weight (OPSO-AEM&NIW) to overcome the drawbacks, such as falling into local optimization, slow convergence speed of opposition-based particle swarm optimization. Davoodi et al. [11] proposes a new approach, based on a hybrid algorithm combining of Improved Quantum-behaved Particle Swarm Optimization (IQPSO) and simplex algorithms. It can give a good direction to the optimal global region. IQPSOS has the robustness and better convergence under normal and critical conditions, when conventional load flow methods fail. Zhang and Wu [12] proposed adaptive CPSO (ACPSO) to train the weights/biases of two-hidden-layer forward neural network in order to develop a hybrid crop classifier for polarimetric synthetic aperture radar images.

PSO has been applied in various academic and industrial fields so far. The hottest application categories are "electrical and electronic engineering," "automation control systems," "communication theory," "operations research," "mechanical engineering," etc. Larrea et al. [13] proposed the weighted averaging method, where the parameters (weights) are tuned with the Particle Swarm Optimization algorithm. And the algorithm helps forecast the short-term consumption reliably, which improves the operations plan management of the supply companies. Djemama et al. [14] take cellular automata as a modeling tool, an evolutionary process carried out by the QPSO algorithm attempts to extract the rules resulting in satisfactory image denoising and edge detection. Experimental results demonstrate the feasibility, the convergence and robustness of the QPSO algorithm for solving reverse emergence in the specific application of image processing. Cai and Yang [15] proposed an improved PSO-based approach for a team of mobile robots to cooperatively search for targets in complex unknown environments. The improved cooperation rules for a multirobot system were applied in the potential field function, which acted as the fitness function of the PSO.

3 Kalman-Sheep Algorithm

Except PSO, there are a lot of other swarm intelligence algorithms, such as Ant colony optimization (ACO) algorithm, artificial bee colony (ABC) algorithm, cuckoo search and so on. These algorithms realize intelligent optimization by simulating the behavior of a group in reality. Because of this, they can't be completely consistent with the core of the algorithm. We hope the algorithm, at the beginning, can reasonably carry out global exploration and local optimization to quickly converge to the feasible solution, timely judge whether they fall into local optimization and execute the corresponding jump out mechanism. At the same time, the algorithm should have as few parameters as possible to maintain simplicity.

In the Kalman filter, the error covariance matrix P of the initial filter can't be taken as zero, because this may make Kalman fully believe that the given initial state is the optimal of the system, so that the algorithm can't converge. In the progress of Kalman filter, the matrix P will constantly update itself, but a good initial value can improve the convergence of the filter and speed up the convergence, so as to make the system enter a stable state as soon as possible.

Given a value range of P, in which the sheep herding algorithm is used to search the best value of P, so that when p reaches this value, the effect will be more accurate and the denoising performance will be better. In this way, the evaluation of Kalman filtering is equivalent to the fitness function of PSO algorithm. The fitness function's input variable is the error covariance matrix P of initial filtering, and its output is the evaluation of filtering effect, and optimization P is a process in which PSO algorithm constantly calls filtering algorithm. Therefore, we propose a new Kalman-sheep algorithm (KSP), which is simple, efficient and easy to implement.

3.1 Architecture of the Algorithm

The inputs of the Kalman filter algorithm in the figure are the measured value Z_k, the estimated value X_k and the state covariance matrix P_i at time K. The complete architecture of the system is illustrated in Fig. 1.

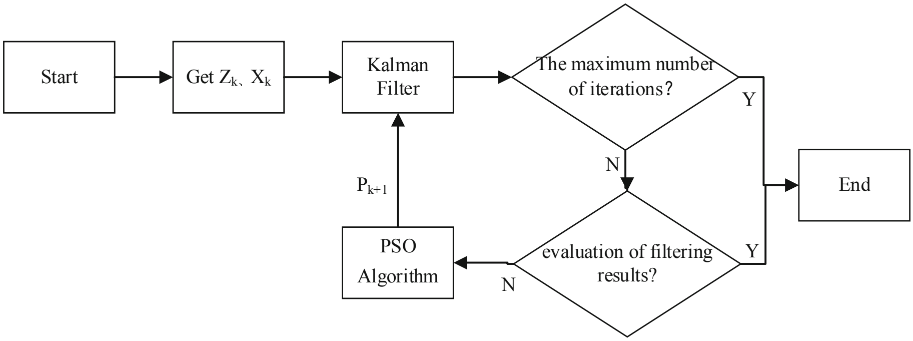

Start
Get Zk、Xk
Kalman Filter
The maximum number of iterations?
Y
N
End
Pk+1
PSO Algorithm
evaluation of filtering results?
Y
N

Fig. 1. Flowchart of the algorithm

a) After Kalman filtering according to Z_k, X_k and corresponding parameters P_k, an optimal estimation value can be obtained, which is recorded as $X_k(P_k)$.
b) Calculate the adaptive function of $X_k(P_k)$ to judge whether the Kalman filter algorithm has reached an efficient operation state. If the system state is not good, enter the PSO algorithm; If the parameter p reaches a good value, the algorithm ends.
c) Optimize the input parameter P, and finally get a better parameter P_{k+1} for the next round of Kalman filter algorithm.
d) Iteratively update P to obtain an optimal input parameter P.

3.2 Sheep Herding Optimization Algorithm

PSO algorithm is the core part of the algorithm in this paper. In the SHO, the value of P represents the position of each sheep, which is the parameter. Z_k is the measured value, and X_k is the estimated value at time k.

A fitness function is a particular type of objective function that is used to summarize, as a single figure of merit, how close a given design solution is to achieving the set aims. The fitness function to judge the position of each sheep can use the mean square error between the observed value and the optimal estimate at time k, where M is the number of iterations.

$$\mathrm{f}(\mathrm{P}_k) = \frac{1}{M}\sum_{k=1}^{M} abs(Z_k - X_k(P_k))^2$$

The flowchart of SHO is shown in Fig. 2. The algorithm mainly consists of the following three parts.

Sheep Lead

In the algorithm, each sheep in the group will move a certain distance towards the head sheep. In this stage, the input is X_i^{old} of each non head sheep, and the output is the latest position X_i^{new} of each sheep. X_{leader} indicates the position of the leader at the moment.

$$\mathrm{X}_i^{new} = \mathrm{X}_i^{old} + rand(0,1) * \left(X_{leader} - \mathrm{X}_i^{old}\right) \tag{2}$$

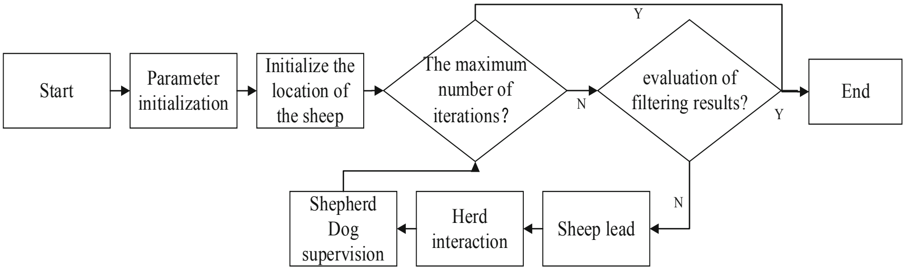

Start
Parameter initialization
Initialize the location of the sheep
The maximum number of iterations?
evaluation of filtering results?
End
Y
N
Y
N
Shepherd Dog supervision
Herd interaction
Sheep lead

Fig. 2. Flowchart of SHO

The position of each sheep can be updated according to the formula, where rand (0,1) indicates that the sheep in the group approach the leader randomly in varying degrees. However, if the fitness function value is not improved after updating, the optimization will be abandoned.

Herd Interaction

In the algorithm, each sheep will randomly find a sheep in the flock for adjustment. At this time, the input objects are X_i^{old} of the sheep and X_j^{old} of the sheep who is randomly chosen in the flock. Compare the fitness function values of the two. If the value of the latter is greater than the former, X_i moves a random distance towards X_j, and X_j moves away from X_i to a certain extent.

$$X_i^{new} = X_i^{old} + rand(0,1) * \left(X_j^{old} - X_i^{old}\right) \tag{3}$$

$$X_j^{new} = X_j^{old} + rand(0,1) * \left(X_j^{old} - X_i^{old}\right) \tag{4}$$

On the contrary, if the fitness function value of the former is large, it will be updated according to the following formula.

$$X_i^{new} = X_i^{old} + rand(0,1) * \left(X_i^{old} - X_j^{old}\right) \tag{5}$$

$$X_j^{new} = X_j^{old} + rand(0,1) * \left(X_i^{old} - X_j^{old}\right) \tag{6}$$

As in the previous stage, if the fitness function values of X_i and X_j are not improved after the update, the update will be cancelled.

Shepherd Dog Supervision

Once the difference between the old and old fitness function values of the leader is less than a threshold K, the shepherd dog mechanism will be triggered. In this mechanism, the number of sheep reset is described by probability P. If the shepherd mechanism is triggered and a sheep is chosen, it will be reset. If a sheep is not driven by the shepherd dog, that is, the sheep is not reset, it will randomly select a sheep that has been reset and move towards it.

$$X_i^{new} = X_i^{old} + rand(0,1) * \left(X_i^{old} - X_j^{old}\right) \tag{7}$$

As in the previous stage, if the fitness function values of X_i and X_j are not improved after the update, the update will be cancelled.

3.3 Kalman Filter Algorithm

Computationally, the multiplication of these probability density functions relates to the discrete KF equation designed for stochastic systems, which is similar to the state observer for deterministic systems:

$$x_{k+1} = Ax_k + B\mu_k + \omega_k \tag{8}$$

The state vector x_k contains information about the position, direction, and speed, and these variables should be estimated. It represents the a priori state, while the vector x_{k+1} represents the posteriori state. The variables to be estimated are given by matrices A(state transition matrix) and B(control matrix). The variable μ_k represents the input from which the estimate is derived, and ω_k takes into account the noise. KF is a two-step algorithm. It consists of a prediction part (time update equations) and an estimation part (measurement update equations).

Firstly, after measurements are taken, the prediction part is done. The linear system model, without dynamic noise taken into account, is used to calculate the a priori state estimate $\hat{x}$ and the error covariance P:

$$\hat{\mathrm{x}} = Ax_{k-1} + B\mu_k \tag{9}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{10}$$

The a priori error covariance matrix P is based on knowledge about the difference between measured states and previously estimated states. Matrix Q is the system process noise covariance matrix, defined in time intervals. It collects data about unmeasured dynamics and sensor noise.

The second part of the algorithm uses the a priori estimates calculated in the prediction step and updates (correct) them to find the posteriori estimates of the state and to minimize error covariance. The state estimate of correction is given by:

$$\hat{\mathrm{x}} = \hat{x}_k^- + K_k\left(y_k - H\hat{x}_k\right) \tag{11}$$

$$P_k = (I - HK)P_k^- \tag{12}$$

where H is a measurement matrix related to the connection between the current state and measurement. The expression $\left(y_k - H\hat{x}_k\right)$ represents the deviation of the actual y_k measurement from the predicted measurement. The Kalman gain K_k is calculated so that it minimizes the posteriori error covariance:

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R} \tag{13}$$

where R is the measurement noise matrix. Once the update equations are calculated, in the next time step, the posterior estimates are used to predict the new a priori estimates, and the previous steps are repeated. To estimate the current state, the algorithm does not need all past measurements. Only estimated states, the error covariance matrix from the previous time step, and the current measurement are needed.

4 Experiment

To evaluate the performance of Sheep Herding optimization algorithm, a comprehensive set of benchmarks are adopted. These functions had local optima and/or saddles in their solution spaces where the number of local minima increases exponentially with the problem dimension. The formulation of each function, feasible solution space, and $f_{\min}$ are listed in Table 1.

Sphere and Shifted Sphere function are typical representatives of unimodal function, as Rastrigrin and Ackley function are typical representatives of multimodal function. Take Sphere function as an example, it is a continuous, convex, symmetrical and unimodal function. It is mainly used to test the local search ability of the algorithm. On the other hand, Rastrigrin function is multi-modal and usually employed for evaluating the global search ability of the algorithm. Multimodal functions have many local minima and are difficult to be optimized. For multimodal functions, the final results are more important to be obtained since they reflect the ability of the algorithm in escaping from poor local optima and locating a near-global optimum.

Table 1. Benchmark functions.

Functions	Formulations	Feasible solution space	$F_{\min}$
Sphere function	$f_1(x) = \sum_{i=1}^{n} x_i^2$	(-100,100)	0
Shifted Sphere function	$f_2(x) = \sum_{i=1}^{n} z^2(i)$	(-100,100)	0
Rastrigrin function	$f_3(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	(-5.12,5.12)	0
Ackley function	$f_4(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z^2(i)}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi z(i))\right) + 20 + e+1$	(-32,32)	0

All of simulations in this paper are executed on a PC with a 2.66 GHz Intel Processor and 4.0 GB RAM. All of programs are written and executed in MATLAB 7.6.0. For all these algorithms, a population of 40 individuals corresponding to the dimensions 30 is used, the maximum iteration times of unimodal function (f1, f2) and multimodal function (f3, f4) are 10000 and 100000 respectively. We set P = 0.2, and ε is the error threshold of the test function.

In our work, statistical measures are used to assess performance of these algorithms. The root-mean-square error (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. The RMSE represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences. These deviations are called residuals when the calculations are performed over the data sample that was used for estimation and are called errors (or prediction errors) when computed out-of-sample. The RMSE serves to aggregate the magnitudes of the errors in predictions for various data points into a single measure of predictive power.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \widehat{x_i}\right)^2} \tag{14}$$

4.1 SHO

Table 2 gives the comparison of the processes of SHO and PSO in well-known four benchmark functions averaged on 100 trial runs. The mean and standard deviation of solutions found by these algorithms are also listed in Table 2. For unimodal functions, the performance of so algorithm is significantly better than that of standard PSO algorithm; For multimodal functions, so algorithm obtains the global optimal value on function F4, the solution quality is much better than PSO, the solution quality on functions F5 and F8 is much better than PSO, and the solution quality on functions F6 and F7 is similar to PSO, The main reason why so algorithm can obtain higher quality solutions than PSO algorithm is that its head sheep leading mechanism can quickly lead all sheep to actively approach the currently known optimal solution, strengthen the global exploration ability, and the sheep interaction mechanism can accelerate the sheep to approach the surrounding optimal solution, actively judge whether they fall into the local optimal solution and execute the corresponding jump out mechanism while converging rapidly.

Table 2. Comparison of SHO and PSO algorithm

Function	Algorithm	Mean	RMSE	Optimal value	Worst Value
f1	SHO	1.56E−19	1.35E−19	1.83E−20	5.82E−19
	PSO	1.54E−18	2.99E−18	3.05E−21	1.62E−17
f2	SHO	1.50E−13	3.59E−14	1.14E−13	2.27E−13
	PSO	1.70E + 02	2.87E + 02	5.68E−14	1.50E + 03
f3	SHO	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
	PSO	3.24E + 01	1.07E + 01	1.69E + 01	5.77E + 01
f4	SHO	7.25E−15	1.00E−15	7.11E−15	1.42E−14
	PSO	7.39E−15	1.41E−15	7.11E−15	1.42E−14

4.2 KSP

Figure 3 shows the trajectory of robot. In Fig. 3, there is a deviation between the actual trajectory and the designed trajectory of the robot because of the noise and mileage meter error in the process of robot motion. It also shows deviation curve between EKF and AEKF.

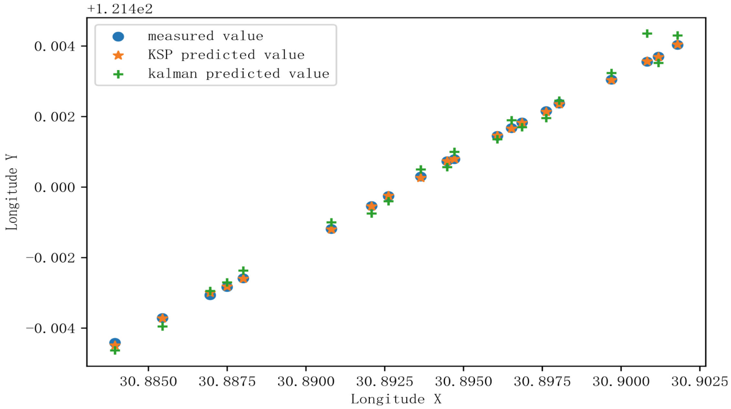


Fig. 3. The trajectory of boat in KSP and Kalman

In Fig. 4, we introduce a new algorithm, GO-PSO, to compare with KSP. GO-PSO introduces global optimization method to improve the search efficiency of particle swarm optimization algorithm and overcome the shortcomings of its easy to fall into local optimal solution. Figure 4 shows that the MSE of the Kalman algorithm based on sheep optimization tends to be constant after about 14 iterations, while the optimal solution of the GO-PSO algorithm tends to be stable after 24 iterations; According to the curve in Fig. 4, the performance of KSP algorithm is better than GO-PSO algorithm, so the Kalman optimized by KSP can obtain the global optimal solution in a more accurate way.

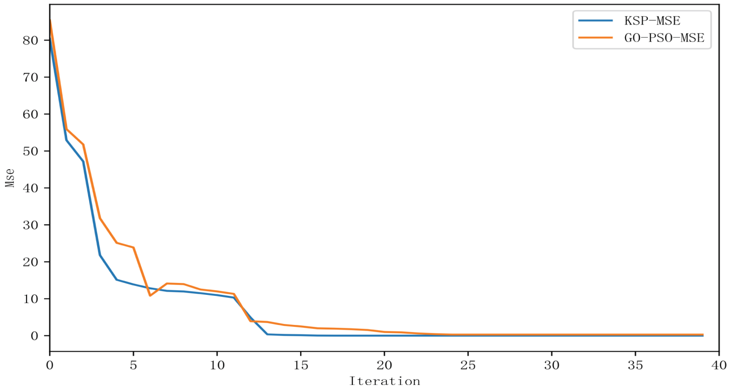


Fig. 4. The relationship between iterations and MSE

Figure 5 shows the comparison of trace error before and after filtering. Before filtering, the trace error (shown in blue) is at a relatively high level, while after filtering, the trace error (shown in orange) fluctuates in a relatively low range. It can be seen that the filter is effective and it can reduce the error to a certain extent and control the error within an acceptable range.

The error fluctuates with time in the experiment, because it is carried out before and after the algorithm training, not the real-time error in the training process. Therefore, the error does not gradually decrease with time, but after training, the average error of the algorithm has decreased significantly.

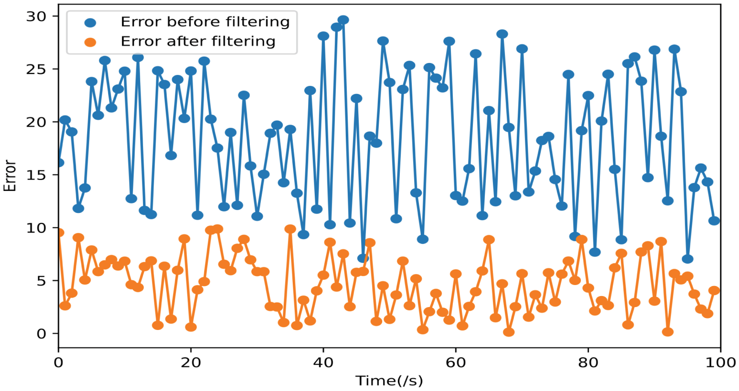


Fig. 5. The error before and after filtering

5 Conclusion

In this paper, an optimized Kalman filter method is proposed, which can not only maintain the advantages of particle swarm optimization algorithm, but also effectively avoid particles falling into local optimal solution in the search process; Through dynamic adjustment, the particles gradually converge to the global optimal value. The KSP algorithm has good convergence effect, and the filtering effect of the optimal parameters is obviously better than the traditional particle swarm optimization algorithm.

References

1. Mcgee, L.A., Schmidt, S.F.: Discovery of the Kalman Filter as a Practical Tool for .Aerospace and Industry. National Aeronautics & Space Administration Ames Research, Moffett Field, pp. 1–13 (1985)
2. Bonnabel, S, Martin, P., Salaun, E.: Invariant extended Kalman filter: theory and application to a velocity-aided attitude estimation problem. In: IEEE Conference on Decision & Control. IEEE, Shanghai, pp. 1297–1304 (2009)
3. Pi, Y., Yuan, Q., Zhang, B.: The application of adaptive extended Kalman filter in mobile robot localization. In: 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China, pp. 5337–5342 (2006)

4. Hesch, J.A., Kottas, D.G., Bowman, S.L., Roumeliotis, S.I.: Observability-Constrained Vision-aided Inertial Navigation, p. 24 (2016)
5. Weisheng, X.J., et al.: Pedestrian navigation algorithm based on improved Kalman filtering. J. Navig. Position. **9**(2), 28–34 (2021)
6. Ke, L., Rui, W., et al.: The research of rader single target tracking algorithm based on Kalman filter. Space Electr. Technol. **16**(1), 16–20 (2019)
7. Assaf, M.H., Petriu, E.M., Groza, V.: Ship track estimation using GPS data and Kalman filter. 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 1–6 (2018)
8. Lalwani, S., et al.: A Comprehensive Survey: Multi-objective Particle Swarm Optimization (MOPSO) Algorithm: Variants and Applications, no. 1, p. 64 (2013)
9. Zhenlun, Y.: Stored Information recombination based particle swarm optimization algorithm and its applications. South China University of Technology, Guangzhou, China (2016)
10. Wen-yong, D., Lan-lan, K., et al.: Opposition-based particle swarm optimization with adaptive elite mutation and nonlinear inertia weight. J. Commun. **37**(12), 10 (2016)
11. Davoodi, E., Hagh, M.T., Zadeh, S.G.: A hybrid improved quantum-behaved particle swarm optimization–simplex method (IQPSOS) to solve power system load flow problems. Appl. Soft Comput. **21**, 171–179 (2014)
12. Chuang, L.Y., Tsai, S.W., Yang, C.H.: Chaotic catfish particle swarm optimization for solving global numerical optimization problems. Appl. Math. Comput. **217**(16), 6900–6916 (2011)
13. Ml, A., Ap, B., Ei, A., et al.: Extreme learning machine ensemble model for time series forecasting boosted by PSO: application to an electric consumption problem. Neurocomputing **452**, 465–472 (2021)
14. Djemame, S., Batouche, M., Oulhadj, H., Siarry, P.: Solving reverse emergence with quantum PSO application to image processing. Soft. Comput. **23**(16), 6921–6935 (2018). https://doi.org/10.1007/s00500-018-3331-6
15. Cai, Y., Yang, S.X.: An improved PSO-based approach with dynamic parameter tuning for cooperative multi-robot target searching in complex unknown environments. Int. J. Control **86**(10), 1720–1732 (2013)
16. Wei, T., et al.: Track fusion based on particle swarm optimization algorithm with genetic operator. J. Chongqing Univ.(Natural Science), **29**(5), 4 (2010)
17. Daqing, Y.: Application of particle swarm optimization algorithm in improved aircraft track fusion based on Kalman filter. Softw. Guide **12**(10), 3 (2013)
18. Akca, A., Efe, M.Ö.: Multiple model Kalman and particle filters and applications: a survey. IFAC-PapersOnLine **52**(3), 73–78 (2019)
19. Selimovi, D., et al.: Improving the performance of dynamic ship positioning systems: a review of filtering and estimation techniques. J. Marine Sci. Eng. **8**(4), 234 (2020)
20. Djemame, S., et al.: Solving reverse emergence with quantum PSO application to image processing. Soft Comput. **23**, 1–15 (2018)
21. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artif. Intell. Rev. **42**(1), 21–57 (2012). https://doi.org/10.1007/s10462-012-9328-0
22. Kivi, M.E., Majidnezhad, V.: a novel swarm intelligence algorithm inspired by the grazing of sheep. J. Ambient Intell. Hum. Comput. **13**, 1201–1213 (2021)
23. Krause, J., Cordeiro, J., Parpinelli, R.S., et al.: A survey of swarm algorithms applied to discrete optimization problems. Swarm Intell. Bio-Inspired Comput. **4**(9), 169–191 (2013)
24. Wu, Y., Liu, G., Guo, X., Shi, Y., Xie, L.: A self-adaptive chaos and Kalman filter-based particle swarm optimization for economic dispatch problem. Soft. Comput. **21**(12), 3353–3365 (2016). https://doi.org/10.1007/s00500-015-2013-x
25. Parpinelli, R.S., Lopes, H.S.: New inspirations in swarm intelligence: a survey. Int. J. Bio-lnspired Comput. **3**(1), 1–16 (2011)

26. Pellegrini, P., Stuitzle, T., Birattari, M.: A critical analysis of parameter adaptation in ant colony optimization. Swarm Intell. **6**(1), 23–48 (2012)
27. Dorigo, M., Blum, C.: Ant colony optimization theory: a survey. Theoret. Comput. Sci. **344**(2–3), 243–278 (2005)
28. Yang, X.S., Deb, S.: Engineering optimization by cuckoo search. Int. J. Math. Model. Num. Optim. **1**(4), 330–343 (2010)
29. Meuret, M., Provenza, F.D.: When art and science meet: integrating know ledge of French herders with science of foraging behavior. Rangel. Ecol. Manage. **68**(1), 1–17 (2015)
30. Cai, X., Cui, Z., Zeng, J., Tan, Y.: Particle swarn optimization with self-adjusting cognitive selection strategy. Int. J. Innov. Comput. Inf. Control **4**(4), 943–952 (2008)

Target Detection Algorithm Based on Feature Optimization and Sample Equalization

Chao Li[1], Fangzheng Huang[1](✉), Zhaoxian Yang[2], Zhou Wang[1], and Dayan Ban[1]

[1] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo N2L 3G1, Canada
{c638li,f34huang,zhou.wang,dban}@uwaterloo.ca

[2] University of Electronic Science and Technology of China, Chengdu 611731, China
202022011927@std.uestc.edu.cn

Abstract. In the field of intelligent security, due to the negative effect of complicated influence factors, such as low video quality, different size of the target and occlusion, target detection is hard to be well-applicated in real life. Based on the above problems, this paper proposes a multi-layer feature cascade aggregation pyramid network (MCA-FPN) on the basis of Faster RCNN, which can fully combine the different level of semantic information to generate optimized feature maps and improve the expression ability of different size of features finally. Besides, to remove the negative effect from the imbalance distribution of samples, this paper discusses a new sample balanced weighted loss function SB-Loss to increase convergence speed and make the training process more efficiency. Finally, the method proposed by this paper has been experimented on the Pascal VOC dataset, with a maximum accuracy of 86.0%, which is highly competitive in this research area.

Keywords: Target detection · FPN · Sample balanced

1 Introduction

The target detection has raised a lot of attention from both academic and engineering fields, and it has also been applied to people's real life. In fact, the process of target detection achieved by algorithm is very similar with human's approach, both of which need to confirm the category and location of target from specific image, so as to complete the two main tasks of target classification and identification [1]. However, in realistic application, there will be a lot of factors affecting the accuracy and speed of detection algorithm, such as the occlusion and frame lost for specific image. These factors may lead to an information lost phenomenon and make it hard for traditional methods to analyze images accurately.

In recent years, the emergence of convolution neural network has greatly helped to increase the accuracy of target detection and has attracted a lot of attention from the academic field [2]. Neural network method can process wider levels of features

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 343–355, 2022.
https://doi.org/10.1007/978-3-031-19214-2_28

and extract more detailed information from original image, which solves the problems mentioned above to some extends. Most of recent research focus on changing or fine-tuning the existing convolution neural network structures. As shown by the result of ILSVRC, which is one of the most influential competitions in the computer vision field [3], it is common to see that the convolution neural network-based method performed better or even the best in the past few years, which also illustrates the importance of convolution neural network on target detection.

There are different kinds of classical image detection algorithm. The one-step target detection algorithm is relatively direct and simple. It usually operates directly on the feature map and then send the extracted information to the final result, one of the main advantages of such network structure is to increase the processing speed of network which leads to a relatively fast detection speed than other methods. The step-by-step target detection algorithm on the other hand, divide the network into multiple subnets and process the feature map in different stages. Because most of the two-stage detection algorithms have more detailed network structures and divisions, they tend to perform better with higher accuracy than the one-step method. For example, a recent detection algorithm fast RCNN is a typical representative. Therefore, if there is a way to increase the accuracy of the one-step target detection method, the algorithm would benefit from both advantages of fast speed and high accuracy.

One of the main reasons that one-step detection network perform badly can be attributed to the imbalance division between positive and negative samples. By solving this disadvantage, the performance of the one-step detection method increases to a similar level with the step-by-step method [4]. Meanwhile, the feature pyramid structure proposed by FPN [5] enables hierarchical processing on the output generated by ResNet backbone network, so that the shallow and deep features can be effectively utilized, which enhances the utilization of feature information by the model and improves the performance of the algorithm.

To solve the practical problems in security monitoring application, this paper proposes d a series of method based on the deep learning target detection model to improve the performance of detection. The main innovations are as follows:

(1) A new multi-layer feature cascade aggregation pyramid network (MCA-FPN) is proposed based on the structure of feature pyramid. This module can fully combine the deep and shallow feature semantic information and improve the representation ability of each feature.
(2) Optimize the positive and negative sample matching mechanism, and design the sample equilibrium loss function by asymmetrically weighting the positive and negative samples to solve the imbalance existing in the training process. The experiment result shows that the optimized network model can converge quickly.
(3) The designed multi-layer feature cascaded aggregation pyramid network (MCAFPN) can be easily applied to other detection algorithms and help to achieve the algorithm migration on the basis of better performance.

2 Related Work

2.1 Target Detection

The two-stage target detector is developed based on the RCNN [6] network. RCNN firstly determines the potential location of the target through the selective search method and extracts the corresponding features by models. Then, it carries out linear classification and prediction of the target in each region through supervised learning method to identify the target categories. On the basis of RCNN, the deep learning detector of fast RCNN [4] and regional recommendation network (RPN) are proposed, which reduce the computational complexity compared to the previous selective search methods. Besides, as an improvement of the isolated RCNN modules, modules from Faster RCNN complement with each other very well, which not only reduces the training loss among different modules but also greatly meet the requirement of real-time detection. What is more, some improved research based on the above network are proposed to further decrease the computational complexity and increase the detection speed. Examples include the fully convolution RFCN [7] and Lightweight backbone Light-RCNN [8].

For the single-stage target detector, Yolo detection network [9] is the typical representative which cuts the input picture into different regions and completes the detection task based on the pixels of each region. The subsequent improved version of Yolo [10–12] made further breakthrough in detection speed. SSD [13] detection model optimizes the resolution and spatial information of the input image, which greatly makes up for the disadvantages of the single-stage target detector in scale problem. Inspired by the idea of SSD layered detection of different scale feature maps, RetinaNet [14] is proposed. In order to solve the problem of uneven foreground and background of the training samples in the single-stage detection model, RetinaNet designed a new loss function based on the theory of standard cross semantic entropy loss, to increase the attention towards ‘hard example’ during the process of network training.

2.2 Feature Pyramid Networks

For the better use of hierarchical feature output extracted by backbone network, people proposed a pyramid-structured network to process the output from different layers, which is called feature pyramid model in research field. SPPNet [5] is the model to study the pyramid structure.

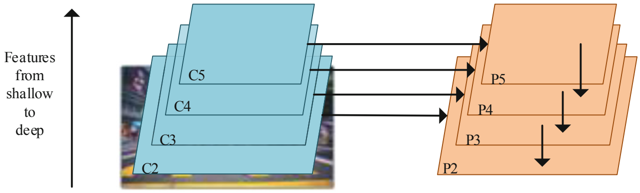


Fig. 1. Feature pyramid networks

Based on SPPNet, the feature pyramid structure FPN is proposed to allow the detection network to adapt to the targets with various sizes in the image and solve the problem of low utilization rate of training samples. The basic construction method of FPN network is shown in Fig. 1.

On the basis of the original FPN structure, many people carried out more detailed research. Panet [17] realizes the effective utilization of underlying information by adding cross layer connections on FPN. Stdn [18] makes efficient use of the cross-scale feature through the scale transmission module. G-frnet [19] utilizes the feature information with different sizes, which fuse the information from different receptive fields well. NAS-FPN [6] and auto FPN [4] use e-learning [7] to obtain the best construction method. And finally, EfficientDet [8] suggests the design of duplicate BiFPN layers. However, the above methods either add other structures to the original FPN structure to improve the performance, or improve only one of the two disadvantages. There is no one method that focuses on solving both of the disadvantages.

2.3 Imbalance of Sample Distribution

The problem of sample distribution imbalance refers to the extremely imbalances among different kinds of samples during the training process. A recent study shows that it is difficult for the model to learn useful features continuously from simple samples, by contrast, sometimes the difficult samples are also helpful for training [9]. For example, it is easy for model to train when simple sample occupies a large proportion of the whole sample and the network will be converged quickly. However, such trained network will perform badly when the distribution of samples changes.

In machine learning, in order to deal with the problem of sample imbalance, data preprocessing and network fine-tuning are usually selected [10–16]. The preprocessing method includes data superposition training, category balance selection etc. From the perspective of network fine-tuning, such as the fast RCNN, the number of positive and negative samples will be set by super parameters to ensure the balance of samples in the RPN and ROI pooling stage. For the screening of candidate areas, it firstly determines a fixed number of positive samples, then manually set the sample selection proportion to determine other samples. In fact, such empirical setting method is not suitable for different tasks and data. Moreover, this kind of method only optimizes the number of positive and negative samples and does not optimize the imbalanced weights of different samples occupied in loss function.

3 Methodology

3.1 Framework

The whole framework of the proposed model consists of the backbone network Faster RCNN and two new modules: a multi-layer feature cascade aggregation pyramid network (MCA-FPN) and a sample balanced loss function SB-Loss. The detail of the process are shown in Fig. 2.

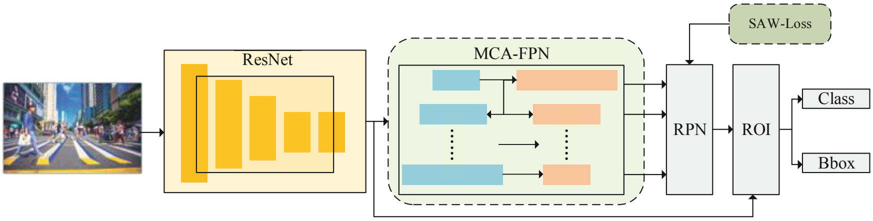


Fig. 2. Overall framework flow

To start with, the input image passes through the backbone network and multi-layer feature cascade aggregation module to generate multi-scale deep features. Then these features are sent to regional proposed network to produce anchor boxes with different length-width ratios and sizes. Finally, the output of MCA-FPN module passes through the ROI pooling layer to obtain down-sampled features, which is used for final multiclassification and regression operation. The improved methods are introduced in detail below.

3.2 Multilayer Feature Cascaded Aggregation Pyramid Network (MCA-FPN)

This paper proposes an improved structure based on traditional feature pyramid, as shown in Fig. 3(1).

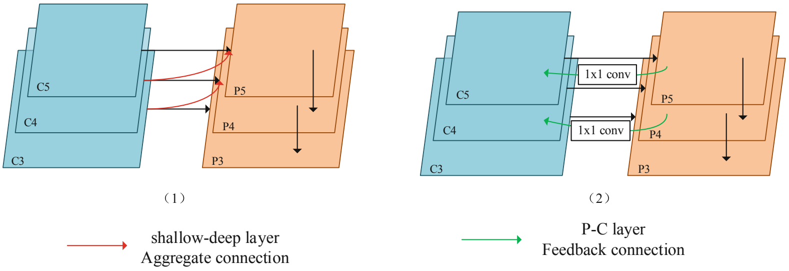


Fig. 3. Improved feature pyramid network

Take the P4 layer as example: the original P4 layer is produced by the feature aggregation from P5 layer and C4 layer, which only contains the information from the current layer (C4) and deep layer (P5).

Our improvement is to add an extra connection from C3 layer to P4 layer, which helps P4 layer to learn the information from shallow layer (C3). Such design will not change the scale of original feature map but will make the network more sensitive towards the location and category information of target.

Meanwhile, inspired by the improvement on ResNet backbone from [20], another feedback connection from feature pyramid P to C is added (as shown in Fig. 3(2)). The ResNet network will allow original input x and feedback input R(f) to be calculated in the regional proposed network. Significantly, the feedback connection here is only operated on the first residual blocks, which has limited negative effect on the whole network structure.

The input of traditional FPN module is shown below:

$$F_{BB}^{i} = C_i(x_{i-1}) \tag{1}$$

$$F_{FPN}^{i} = P_i(f_{i+1},\, x_i) \tag{2}$$

where C_i represents the i^{th} operation from the bottom-up pyramid (left side), P_i represents the i^{th} operation from the up-bottom pyramid (right side), F_{BB}^{i} represents the input feature from backbone network to FPN module $\left\{F_{BB}^{i} | i = 1, \ldots, S\right\}$, F_{FPN}^{i} represents the output feature from FPN module $\left\{F_{FPN}^{i} | i = 1, \ldots, S\right\}$ and the S represents the number of stages.

After improving the shallow to deep aggregation link and P-C feedback link, the input and output of the model are expressed as follows:

$$F_{FPN}^{i} = P_i\left(F_{FPN}^{i+1},\, F_{BB}^{i},\, F_{BB}^{i-1}\right) \tag{3}$$

$$F_{BB}^{i} = C_i\left(F_{BB}^{i-1},\, R_i\left(F_{FPNi}^{i}\right)\right) \tag{4}$$

$$R_i\left(F_{FPNi}^{i}\right) = \mathrm{Conv}\left(F_{FPNi}^{i}\right) \tag{5}$$

3.3 Sample Balanced Loss Function (SB-Loss)

The positive and negative samples participated in the target detection model should be firstly ensured that their quantity and distribution are appropriate for training; Secondly, the proportion of positive samples which have limited continuously positive effect on model training should be reduced. Thirdly, the importance of difficult examples it should be emphasized. In other words, the proportion of difficult examples should be increased during the training process.

The fast RCNN loss function is composed of two parts, in which the classification branch of RPN network selects BCE cross entropy loss:

$$L = -y \log y' - (1 - y)\log\left(1 - y'\right) \tag{6}$$

where y represents the label of target which takes a value of 1 when the sample is positive, and 0 when sample is negative, $y^{'}$ represents the predicted probability of network for target y, and the range of y and $y^{'}$ are both from 0 to 1. Obviously, for positive samples, a higher prediction probability the $y^{'}$ will lead a lower value of loss function. However, for negative samples, the lower the predicted probability, the lower the $y^{'}$ will be in the loss function. This design is more likely to lead to a slow training literation phenomenon when the model trained on the dataset with extreme distribution (too many easy or difficult samples), sometime it is even hard to be optimal.

To solve the above problems, this paper adds controlling factors into BCE cross entropy loss function to control the distribution of samples for an appropriate ratio. The details are as follows:

Firstly, it introduces k as the controlling factor which balances the negative effect of unbalanced distribution of positive and negative samples, the value of k ranges from 0 to 1. By this step, the order of training has changed from positive samples to negative samples, which means that the network will focus on the positive samples first, then the Eq. (6) changed to:

$$L = \begin{cases} -klog\, y' & y = 1 \\ -(1-k)log\left(1-y'\right) & y = 0 \end{cases} \tag{7}$$

Secondly, it introduces b as the controlling factors which balances the negative effect of unbalanced distribution of easy and difficult samples, the value of b ranges from 0 to 1. Based on the design of Focal Loss [13], by introducing positive exponential factor b, it decreases the value of $\left(1-y^{'}\right)^{b}$, which allows the model to focus more on difficult and incorrect classification samples. Combined with Eq. (7), the new loss function becomes:

$$L = \begin{cases} -k\left(1-y'\right)^{b} log\, y' & y = 1 \\ -(1-k)y'^{b} log\left(1-y'\right) & y = 0 \end{cases} \tag{8}$$

Lastly, based on the Singh's [21] idea of unbalanced weighted mechanism and Cao's [22] idea of importance-based dynamic weight mechanism, this paper proposes a sample asymmetric weighted loss function SB-Loss, which is shown as Eq. (9):

$$SAWLoss = \begin{cases} -\frac{1}{K_{\mathrm{IoU}}}\left(1-y'\right)^{b} \log y' & y = 1 \\ -(1-a)y'^{b} \log\left(1-y'\right) & y = 0 \end{cases} \tag{9}$$

On the basis of Eq. (8), it uses the IoU value as the value of controlling factor k when the sample is positive. By doing this, the samples that have high IoU value are more likely to be classified as positive easy samples and the value of loss function obtained from these samples will be smaller. Therefore, the loss function will focus less on such samples. In comparison, the attention of loss function will focus more on positive difficult samples which is more important than positive easy samples for model training. Moreover, because the IoU value calculation is compulsory for Faster RCNN backbone network, such design will not generate any extra calculations. As a result, the samples concerned in the training process will become positive-difficult, negative-difficult, positive-easy and negative easy.

The final loss function for network training is shown as Eq. (10), where the λ is used to control the loss difference between two branches.

$$L(p_i,\ t_i) = \frac{1}{N_{cls}} \sum\nolimits_i N\mathrm{Loss}\left(p_i,\ p_i^*\right) + \lambda \frac{1}{N_{\mathrm{reg}}} \sum\nolimits_i L_{reg} \tag{10}$$

4 Results

4.1 Training Configuration

In this paper, the experimental setting of Faster RCNN is strictly followed. The size of the image input to the model is reset to 600 * 800, and the training data are expanded by means of flipping and splicing. At the same time, ResNet network adopts ResNet-101 pre-training model in order to ensure the comparative significance of the training results. The momentum and weight attenuation parameters are set as 0.9 and 0.0001 respectively. The change of learning rate adopts preheating strategy. And the IoU crossover ratio threshold is set to 0.7. For parameter setting, refer to the setting of b and λ in Focal Loss, where $b = 2,\ \lambda = 2$.

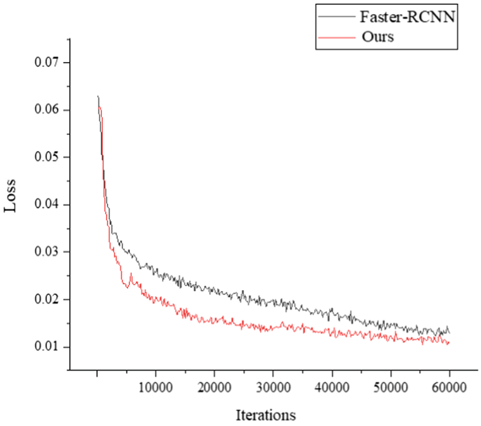


Fig. 4. Training process diagram

The total loss curve in the training process of the network model is shown in Fig. 4. Compared to the loss function in the Faster RCNN, the improved classified SB-Loss loss can converge to a stable state faster, which significantly speeds up the training process and reduces the loss. This effectively improves the learning ability of the network model for positive and difficult samples.

4.2 Ablation Experiment

In this paper, the improved multi-layer feature cascade polymerization pyramid network MCA-FPN and the improved sample equalization loss function SB-Loss are introduced into the original Faster RCNN. Through training and testing the Pascal VOC dataset, the accuracy and the number of parameters of different improved models were obtained, and the effectiveness of the proposed scheme in this paper is demonstrated (Table 1).

We used ResNet-101 as the backbone network to extract image features, and the mAP of Faster RCNN on Pascal VOC dataset reached 79.8%. After adding feature pyramid FPN, the mAP was improved to 82.1%. When FPN is not added, the loss function changed to SB-Loss function, the mAP improved to 83.6%. Without modifying the original loss function, the FPN of feature pyramid was changed to the MCA-FPN of multi-stage feature cascade aggregation pyramid module, and the mAP reached 84.5%. Finally, when we used both the MCA-FPN module and the sample equalizing loss function SB-Loss, the mAP improved to 86.0%. As the network complexity is smaller, the model

Table 1. Experimental results on Pascal VOC dataset

Steps	Backbone network	Added scheme	mAP (%)	Parameters (M)
1	ResNet-101	/	79.8	160.2
2	ResNet-101	FPN	82.1	163.5
3	ResNet-101	SB-Loss	83.6	163.6
4	ResNet-101	MCA-FPN	84.5	165.4
5	ResNet-101	MCA-FPN + SB-Loss	86.0	165.5

is more suitable for industrial landing and application. Compared with Faster RCNN + ResNet-101, the accuracy of the proposed improvement increased by 6.2%, and the model parameters only increased by 5.3M, indicating that the proposed improvement did not introduce too much extra calculation.

4.3 Plug and Play Experiment

MCA-FPN, a multi-layer feature cascade aggregation pyramid module, optimizes features after feature extraction from the backbone network and improves the ability of target detection model to deal with scale problems. In this paper, four typical target detection networks are selected for the multi-layer feature cascade polymerization pyramid network MCA-FPN, and MCA-FPN is embedded into the target detection network. Experiments are carried out in Pascal VOC dataset, and the experimental results are compared with the original structure. The specific results are shown in Table 2.

Table 2. Plug and play performance comparison table

Model structure	Original mAP (%)	MCA-FPN (%)	Promote (%)
DarkNet-Yolov3 [10]	60.6	68.9	13.6
ResNet-FOCS [24]	78.7	85.2	8.2
ResNet-RetinaNet [13]	80.7	84.7	4.9
ResNet-Faster RCNN [5]	79.8	84.5	5.8

As shown in Table 2, the MCA-FPN proposed in this paper can be introduced into other target detection models as a plug and play module, and all of them have certain performance improvement and universality. At the same time, the detection effect of the single-stage target is more significant, indicating that the structure has obvious improvement effect on multi-scale problems.

4.4 Algorithm Comparison Experiment

As shown in Table 3, this paper lists the accuracy results of target detection related models on VOC datasets in the last five years. It can be seen that the mAP of network model proposed in this paper is 1.9% higher than PFPNet, but 0.5% lower than NAS Yolo, the competition model. The effectiveness and superiority of the proposed method are fully proved.

Table 3. Comparison table of algorithm accuracy

Algorithm	mAP (%)	Algorithm	mAP (%)	Algorithm	mAP (%)
Pelee [24]	70.9	MLKP [26]	80.6	RefineDet [29]	83.8
FCOS [23]	78.7	R-DAD [27]	81.2	PFPNet [30]	84.1
HKRM [25]	78.8	RFBNet [28]	82.2	NAS Yolo	86.5
Ours	**86.0**				

4.5 Algorithm Comparison Experiment

Pascal VOC dataset contains 20 detection categories. This paper compares the accuracy of the improved detection model with that of the original baseline model, and the accuracy of each category is shown in Fig. 5. By introducing the MCA-FPN module, it can effectively deal with the multi-scale problems of similar targets, and is more friendly to the detection of small size targets. The part of chairs and tables of the picture often occupies most of the space, which will be hard for network to distinguish the difference between foreground and background, positive and negative samples. This paper designed a SB-Loss function which gives different loss function structure to positive and negative samples and uses the IoU value as the asymmetric weight of loss function for easy samples, such method effectively improves the network learning ability for positive samples and difficult samples.

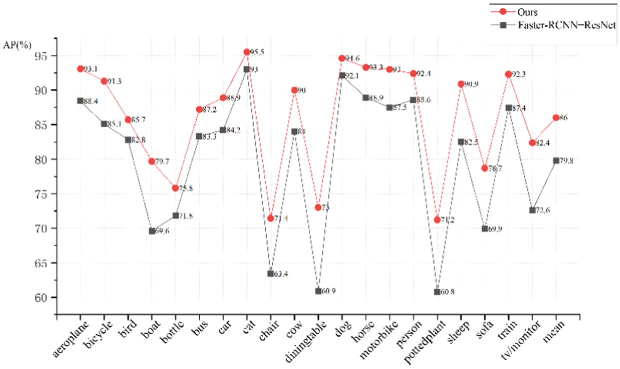


Fig. 5. Accuracy of catagories

The method designed in this paper is intended to improve the ability of the object detection model to deal with size and sample division problems. Combined with the above experiments, various accuracy diagrams clearly showed the effectiveness of the improved method. Among them, the improvement of the detection accuracy of some typical categories truly reflect that the method can better solve the problems faced by the existing detection model. The detection effect diagram of some categories is shown in Fig. 6.

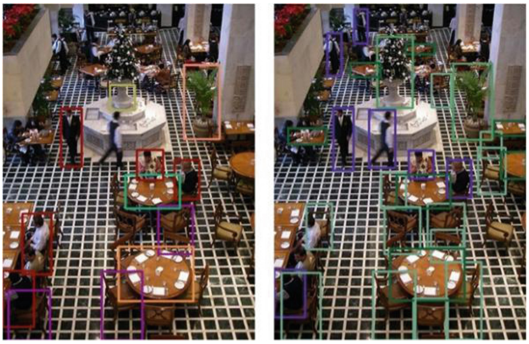


Fig. 6. Partial category inspection effect picture

The above pictures show the difference between the result of Faster RCNN+ResNet structure (left side) and our proposed method (right side). This original image contains some difficult detection tasks such as multi-scaled targets detection and similar background and targets detection, so the result on this picture could illustrate clearly about the how proposed method improve the performance. Firstly, our proposed method decreases the missing rate, such as the successful detection of the waiters in the left top of image, which fails to be detected by Faster RCNN. Secondly, our proposed method increases the detection accuracy, such as the accurate detection of the tree in the middle of image. Moreover, our proposed method could even detect the waiter in the middle of the image even if the target is blur.

5 Results

To solve the problem of fast RCNN being insensitive to feature scale and the problem of imbalance in sample distribution, this paper proposes a multi-level feature cascade aggregation pyramid module MCA-FPN and sample equalization loss function SBLoss. The MCA-FPN module adds two different direction connections between the feature maps, which helps model to extract feature more efficiently and analyze information more accurately. The new sample equalization loss function SB-Loss helps network to use data with different distribution more efficiently and converge faster by adding different types of controlling factors. As shown by the results, the improved model proposed has higher accuracy and faster convergence speed. Besides, from the result of comparative experiments and module plug experiments on Pascal VOC dataset, it is proved that the proposed method can be easily added to different backbone networks, and generate positive effects on their performance.

References

1. Fu, Z., Chen, Y., Yong, H.: Foreground gating and background refining network for surveillance object detection. IEEE Trans. Image Process. **28**(12), 6077–6090 (2019)
2. Fan, Q., Brown, L., Smith, J.: A closer look at faster R-CNN for vehicle detection. In: 2016 IEEE Intelligent Vehicles Symposium (IV), Gotenburg, pp. 124–129. IEEE (2016)
3. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. Adv. Neural. Inf. Process. Syst. **6**(60), 84–90 (2017)
4. Ren, S., He, K., Girshick, R.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
5. Russakovsky, O., Deng, J., Su, H.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vision **15**(3), 211–252 (2015)
6. Girshick, R., Donahue, J., Darrell, T.: Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE Comput. Soc. 580–587 (2014)
7. Dai, J., Li, Y., He, K.: R-FCN: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, pp. 379–387 (2016)
8. Li, Z., Peng, C., Yu, G.: Light-head R-CNN: in defense of two-stage object detector. arXiv preprint arXiv:1711.07264 (2017)
9. Redmon, J., Divvala, S., Girshick, R.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, pp. 779–788. IEEE (2016)
10. Bochkovskiy, A., Wang, C.Y., Liao, H.: Yolov4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
11. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
12. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,Honolulu, pp. 7263–7271. IEEE (2017)
13. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
14. Lin, T.Y., Goyal, P., Girshick, R.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, vol. 42, no. 2, pp. 318–327 (2020)
15. Zhipeng, C., Zaobo, H., Xin, G., Yingshu, L.: Collective data-sanitization for preventing sensitive information inference attacks in social networks. IEEE Trans. Depend. Secure Comput. **15**(4), 577–590 (2018)
16. Zhipeng, C., Xu, Z.: A private and efficient mechanism for data uploading in smart cyber-physical systems. IEEE Trans. Netw. Sci. Eng. (TNSE) **7**(2), 766–775 (2020)
17. LeCun, Y., Bottou, L., Bengio, Y.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
18. Simonyan. K.: Very deep convolutional networks for large-scale image recognition. Computer Science (2014)
19. He, K., Zhang, X., Ren, S.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, pp. 770–778. IEEE (2016)
20. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: British Machine Vision Conference 2016, BMCV (2016)
21. Singh, B., Davis. L. S.: An analysis of scale invariance in object detection snip. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake, pp. 3578–3587. IEEE (2018)

22. Cao, Y., Chen, K., Loy, C.: Prime sample attention in object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, pp. 11580–11588. IEEE (2020)
23. Tian, Z., Shen, C., Chen, H.: FCOS: fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, pp. 9627–9636. IEEE (2019)
24. Wang, R.J., Li. X., Ling, C.X.: Pelee: a real-time object detection system on mobile devices. In: Advances in Neural Information Processing Systems (2018)
25. Jiang, C., Xu, H., Liang, X.: Hybrid knowledge routed modules for large-scale object detection. In: Advances in Neural Information Processing Systems, pp. 1559–1570 (2018)
26. Wang, H., Wang, Q., Gao, M.: Multi-scale location-aware kernel representation for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake, pp. 1248–1257. IEEE (2018)
27. Bae, S.H.: Object detection based on region decomposition and assembly. In: Proceedings of the AAAI Conference on Artificial Intelligence, Hawaii, pp. 8094–8101. AAAI (2019)
28. Deng, L., Yang, M., Li, T.: RFBNet: deep multimodal networks with residual fusion blocks for RGB-D semantic segmentation. arXiv preprint arXiv:1907.00135 (2019)
29. Zhang, S., Wen, L., Bian, X.: Single-shot refinement neural network for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4203–4212 (2018)
30. Kim, S.-W., Kook, H.-K., Sun, J.-Y., Kang, M.-C., Ko, S.-J.: Parallel feature pyramid network for object detection. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11209, pp. 239–256. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01228-1_15

SBA-GT: A Secure Bandwidth Allocation Scheme with Game Theory for UAV-Assisted VANET Scenarios

Yuyang Cheng[1,2], Shiyuan Xu[1,3], Yibo Cao[1,4], Yunhua He[1,5](✉), and Ke Xiao[1]

[1] School of Information Engineering, North China University of Technology, Beijing, China
heyunhua@ncut.edu.cn

[2] Department of Electrical Engineering, The University of Sydney, Sydney, Australia

[3] Department of Computer Science, The University of Hong Kong, Hong Kong, Hong Kong

[4] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China

[5] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

Abstract. Unmanned Aerial Vehicles (UAVs) are widely utilized for wireless communication services, promoting the emergence of promising UAV-assisted vehicle networks. However, due to the ever-increasing traffic data and diversified wireless service requirements of vehicles, there are also privacy issues caused by fraud, which challenges the effective allocation of limited security bandwidth for secure communications. In this article, to solve these two problems, we firstly propose a secure bandwidth allocation scheme based on the game theory on the Internet of Vehicles assisted by UAVs. Secondly, the proposed blockchain-based system introduces an emerging consensus mechanism that can significantly reduce the delay in exchanging information and protect data privacy. Furthermore, to allocate the limited safe bandwidth, based on the real-time feedback of each UAV, we design an optimal decision search algorithm based on gradient descent to achieve Stackelberg equilibrium. Finally, the simulation results show the superiority of improving the utility's security bandwidth allocation scheme.

Keywords: Vehicular Ad-Hoc Network (VANET) · Bandwidth allocation · UAV · Game theory · Information security · Privacy protection

1 Introduction

With the rapid popularity of UAVs equipped with wireless transceivers, a promising UAV-assisted vehicle network has been advocated to provide vehicles with ubiquitous wireless communications [1, 2, 6, 7 14], . The traditional mobile network uses ground base stations to adapt to the wireless access of vehicles [13, 15–18]. High data rate wireless communication for drones. Therefore, the Internet of Vehicles composed of base stations and drones has become a new paradigm for the next generation of communication networks.

Specifically, most current data management systems use a Proof-of-Work (PoW) mechanism in blockchain systems [3]. However, the bandwidth allocation of secure

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 356–364, 2022.
https://doi.org/10.1007/978-3-031-19214-2_29

spectrum resources is essential for providing vehicles with satisfactory Quality-of-Experience (QoE) wireless data services [4].

The Proof of Stake (PoS) mechanism has many advantages [5]. In a PoS-based network, vehicles are essential for maintaining the operation and safety of the network. Therefore, we propose a model based on the Stackelberg game to jointly maximize utility and secure bandwidth allocation, and then design an optimal decision search algorithm based on gradient descent to find Stackelberg equilibrium. Finally, simulations verify the feasibility and effectiveness of the proposed scheme.

We propose a blockchain-based network and game-theoretic security bandwidth allocation scheme. The contribution of this paper can be summarized as follows:

- We propose a secure bandwidth allocation scheme with game theory to provide secure data services to vehicles in VANET, which jointly considers the cooperation and competition between drones and vehicles.
- We develop a novel PoS-based framework for service management, including smart contracts, enabling all vehicles to send feedback messages without any privacy leaks and get safe bandwidth allocation.
- We design utility functions for vehicles and drones, and we utilize the Stackelberg game to study the complex interaction between drones and vehicles.
- An optimal decision search algorithm based on gradient descent is designed to find Stackelberg equilibrium. Finally, through simulation performance, we compare with other methods to prove the superiority and effectiveness of our method.

2 Our Proposed Scheme

2.1 Network Model

As shown in Fig. 1, we consider a UAV-based VANET, consisting of a single ground base station, a roadside unit, multiple drones, and vehicles.

The set of UAVs is represented as $U = \{1, 2, \cdots, U\}$ and the bandwidth of the UAV is B_0. We apply the UAV-to-X communication protocol in the connections between UAVs and the RSU. During the provision of wireless service, the serving UAV hovers over the vehicles. In the time slot t, the location of the UAV u is denoted as $l_u(t) = \{x_u(t), y_u(t), z_u(t)\}$, where z is the height of the UAV. Due to vehicles' mobility, the number of vehicles under each UAV varies over time. At the time slot t, the set of vehicles under UAV u is denoted as $N_u^t = \{1, 2, \cdots, n_u^t, \cdots, N_u^t\}$. The location of the vehicle n_u^t is $l_{n_u^t} = \{x_{n_u^t}, y_{n_u^t}, 0\}$. As such, the distance between UAV u and vehicles n_u^t at a time slot t is given by

$$d_{u,n_u^t}(t) = \sqrt{(x_u(t) - x_{n_u^t}(t))^2 + (y_u(t) - y_{n_u^t}(t))^2 + z_u^2} \tag{1}$$

We assume that the Line-of-Sight (LOS) chain-link dominates the channel between the drone and each vehicle. Therefore, the channel gain from UAV u to vehicle n_u^t is $g_{u,n_u^t(t)}(t) = g_0(d_{u,n_u^t}(t))^{-\mu}$ where g_0 is the UAV-to-ground channel gain with the unit distance and μ is the path loss parameter of the LOS link. Then, the signal to interference plus noise ratio (SINR) at the vehicle n_u^t is

$$\beta_{n_u^t}(t) = \frac{P_u g_{u,n_u^t(t)}(t)}{\sigma^2 + \sum\limits_{u'=1,u' \neq u}^{U} P_{u'} g_{u',n_u^t(t)}(t) + P_0 g_{0,n_u^t}(t)} \qquad (2)$$

where P_u represents the transmission power from the UAV u to each vehicle and σ^2 denotes the white Gaussian noise power. P_0 and $g_{0,n_u^t}(t)$ represents the power of each vehicle and power gain from the base station to the vehicle. At the time slot t, the bandwidth data rate from UAV is $R_{n_u^t} = \log_2(1 + \beta_{n_u^t}(t))$.

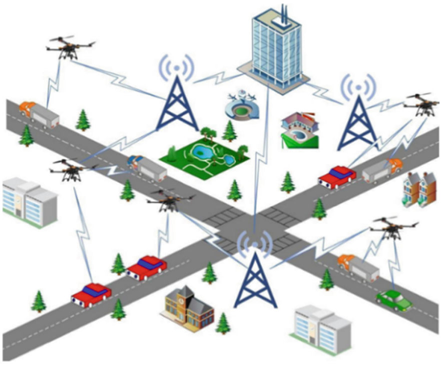


Fig. 1. Network model

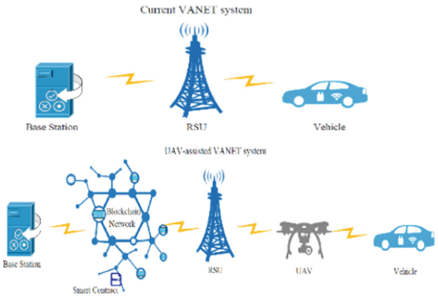


Fig. 2. VANET system and UAV-assisted system

2.2 Our Proposed UAV-Based VANET System

Our proposed UAV-assisted VANET system maximizes the efficiency of secure bandwidth allocation and provides smart contracts in the storage server for protection. The current VANET [12] system and the UAV assistance system elaborates in Fig. 2. The process of management includes the following steps:

Step 1. The communication protocol between the base station and the RSU is stored in the storage server as a smart contract, the program that defines the vehicle. The communication protocol will be automatically executed when the conditions specified by the smart contract are met. **Step 2.** When one vehicle desires the RSU for road traffic information services, the vehicle inquires the RSU through UAV to receive information about the service. **Step 3.** If the vehicle decides to enable the service, the request information and verification information will hear the address sent to the smart contract by the RSU. **Step 4.** Once successfully sent to the smart contract, the RSU will grant the vehicles access to traffic information directly or through the UAV. **Step 5.** When one vehicle finishes using the information service, RSU will send a data packet containing the provided service to the smart contract address. **Step 6.** The smart contract will be automatically calculated and sent to the base station. It also triggers the information transmitted from the base station to the RSU.

2.3 The Secure Bandwidth Allocation System

To find the best bandwidth allocation for vehicles and drones. The utility of vehicles and drones should be designed separately. Their utilities consist of the revenue from the sale of secure bandwidth and the cost of providing wireless services. Therefore, the utility function of UAV in the time slot is expressed as

$$U_u(P_u(t)) = \sum_{n_u^t}^{N_u^t} P_u(t) b_{n_u^t}(t) - \sum_{n_u^t=1}^{N_u^t} c_u(t) b_{n_u^t}(t) \tag{3}$$

Among them, $P_u(t)$ is the security bandwidth price of the drone u $c_u(t)$ is the cost of providing unit security bandwidth for the drone u and $b_{n_u^t}(t)$ is the security bandwidth obtained by the vehicle n_u^t in a time slot t. The utility of each vehicle consists of the satisfaction of obtaining safe bandwidth and the cost of purchasing safe bandwidth from the associated drone. $\gamma_{n_u^t}$ is the satisfaction parameter of the vehicle n_u^t, and $R'_{n_u^t}(t)$ is the data rate requirement of a vehicle n_u^t in the time slot t. Formally, the utility function of a vehicle n_u^t in the time slot t is

$$U_{n_u^t}(b_{n_u^t}(t)) = \gamma_{n_u^t} \log(1 + \frac{R_{n_u^t}(t) b_{n_u^t}(t)}{R'_{n_u^t}(t)}) - p_u(t) b_{n_u^t}(t) \tag{4}$$

2.4 Security Analysis of the Proposed Roaming System

By adopting the Ouroboros consensus mechanism, our roaming management system can achieve a minimal data exchange delay compared with the current roaming system. In particular, it takes about 20 s to add a block to the chain and 3 min to confirm the transaction. Therefore, compared with traditional roaming fraud protection systems, fraud attacks can be detected approximately 4 h earlier. In addition, the Ouroboros consensus mechanism has been proven to resist multiple types of attacks, such as double-spending attacks and grinding attacks [6].

Data authentication and unforgeability: Attackers cannot be used as a legitimate tool to destroy the trust evaluation storage server because it cannot forge any vehicle to apply for verification, and maliciously authorized vehicles are also problematic to destroy the storage server since it is almost impossible to control most entities due to high costs. Malicious attack: A group of malicious vehicles may produce unfair feedback to deal with the security bandwidth allocation of the target vehicle. In our system, the UAV iterative algorithm obtains the allocation, and thus the continuous feedback sent by the malicious vehicle will not be effective.

3 Game Theory Analysis

3.1 Game Modeling

In the Stackelberg game, the goal is to maximize their utility. Therefore, we define the following two problems:

Problem 1.

$$\max \ \{ \ U_u(p_u(t)), p_u(t)\} \geq 0, B_0 \geq \sum_{n_u^t=1}^{N_u^t} b_{n_u^t}(t) \tag{5}$$

The local condition means that the bandwidth price should be greater than or equal to zero, and the amount of bandwidth allocated should be less than the owned by the drone.

Problem 2.

$$\max\{U_{n_m^t}(b_{n_m^t}(t)), b_{n_m^t}(t)\} \geq 0 \tag{6}$$

Among them, the condition means that the amount of bandwidth obtained by vehicles should be greater than zero.

Assumption 1: $b_{n_u^t}(t)'$ and $p_u(t)$ are respectively the solutions of UAV n_u^t **problem 2** and **problem 1** in time slot t. Let $b_u(t)$ be the bandwidth demand vector of vehicles in the coverage area of UAV u, and $b_{-n_u^t,u}(t)'$ be the vehicle bandwidth demand vector except for vehicle n_u^t. We have these two inequalities $U_u(p_u(t)', b_u(t)') \geq U_u(p_u(t), b_u(t)')$ and $U_{n_u^t}(b_{n_u^t}(t)', b_{-n_u^t,u}(t)', p_u(t)') \geq U_u(b_{n_u^t}(t), b_{-n_u^t,u}(t)', p_u(t)')$.

3.2 Follower Strategy

By solving **problem 2**, we obtain the optimal bandwidth purchase strategy for vehicles according to the following theorem.

Theorem 1: Given the bandwidth price, the optimal bandwidth purchase strategy for the UAV covered by the vehicle n_u^t at time t is

$$b_{n_u^t}(t)' = \max(\frac{\alpha_{n_u^t}}{p_u(t)} - \frac{R'_{n_u^t}(t)}{\log(1+\gamma_{n_u^t}(t))}, 0) \tag{7}$$

Proof: We need to know if the utility function has the extremum. The processing of proof elaborates in Table 1.

Case 1: Low Bandwidth Price Regime. The low price situation corresponds to the situation where the bandwidth price provided by UAV u is not greater than $\frac{\alpha_{n_u^t} R_{n_u^t}(t)}{R'_{n_u^t}(t)}$. Therefore, the utility function $U_{n_u^t}(b_{n_u^t}(t))$ initially increases and then declines $b_{n_m^t}(t)$. The optimal bandwidth requirement at the time slot t can be obtained by solving $\frac{\partial U_{n_u^t}(b_{n_u^t}(t))}{\partial(b_{n_u^t}(t))} = 0$. Therefore, within the coverage of UAV u in a time slot t, the optimal bandwidth requirement of a vehicle n_u^t is $b_{n_u^t}(t)' = \frac{\alpha_{n_u^t}}{p_u(t)} - \frac{R_{n_u^t}(t)}{R'_{n_u^t}(t)}$.

Case 2: High Bandwidth Price Regime. The high bandwidth price system means that the bandwidth price provided by UAV u is greater than . So $\lim_{b_{n_m^t}(t)\to 0} \frac{\partial U_{n_m^t}(b_{n_m^t}(t))}{\partial(b_{n_m^t}(t))} < 0$, the first derivative of the utility remains negative as the bandwidth demand enhances. The optimal bandwidth requirement of the vehicle n_u^t in the UAV u coverage of time slot t is $b'_{n_m^t}(t) = 0$.

Table 1. Proof of Theorem 1.

Proof: The max-min value judgment
1: **For** $\frac{\partial U_{n_u^t}(b_{n_u^t}(t))}{\partial(b_{n_u^t}(t))} = \frac{\alpha_{n_u^t} R_{n_u^t}(t)}{R'_{n_u^t}(t)+R_{n_u^t}(t)b_{n_u^t}(t)} - p_u(t)$
2: And $\frac{\partial^2 U_{n_u^t}(b_{n_u^t}(t))}{\partial(b_{n_u^t}(t))^2} = -\frac{\alpha_{n_u^t}(R_{n_u^t}(t))^2}{(R'_{n_u^t}(t)+R_{n_u^t}(t)b_{n_u^t}(t))^2}$ is less than 0
3: **Therefore**, the utility function is concave
4: **For** $\lim_{b_{n_u^t}(t)\to\infty} \frac{\partial U_{n_u^t}(b_{n_u^t}(t))}{\partial(b_{n_u^t}(t))} = -p_u(t)$ is less than 0
5: And $\lim_{b_{n_u^t}(t)\to 0} \frac{\partial U_{n_u^t}(b_{n_u^t}(t))}{\partial(b_{n_u^t}(t))} = \frac{\alpha_{n_u^t} R_{n_u^t}(t)}{R'_{n_u^t}(t)} - p_u(t)$
6: **Therefore**, it has a max or min value

Then, we further analyze the optimal bandwidth price strategy of each UAV is

$$\begin{aligned} U_u(p_u(t)) &= (p_u(t) - c_u(t)) \sum_{n_u^t=1}^{N_u^t} \max(\frac{\alpha_{n_u^t}}{p_u(t)} - \frac{R'_{n_u^t}(t)}{\log(1+\beta_{n_u^t}(t))}, 0) \\ &= (p_u(t) - c_u(t)) \sum_{n_m^t=1}^{N_m^t} (\frac{\alpha_{n_u^t}}{p_u(t)} - \frac{R'_{n_u^t}(t)}{\log(1+\beta_{n_u^t}(t))}) \end{aligned} \tag{8}$$

The second derivative of the UAV utility function relative to the bandwidth price $p_u(t)$ can be expressed as $\frac{\partial^2 U_u(p_u(t))}{\partial(p_u(t))^2} = -2\sum_{n_u^t=1}^{N_u^t}(\frac{c_u\alpha_{n_u^t}}{(p_u(t))^3}) < 0$.

We propose an optimal decision search algorithm based on gradient descent to find the optimal bandwidth pricing strategy for each UAV. By adjusting the policy to improve the utility, the price of the drone u is updated to $p_u(t)[\tau+1] = p_u(t)[\tau] + \varepsilon\nabla U_u(p_u(t)[\tau])$, where $p_u(t)[\tau]$ is the bandwidth price of the UAV m in the $\tau-th$ iteration, ε is the number of iterations of bandwidth price, and $\nabla U_u(p_u(t)[\tau])$ is the gradient value. The iterative process of optimal bandwidth pricing strategy shows in **Algorithm 1**.

Algorithm 1: Iterative algorithm based on gradient descent

```
Initialization: Each UAV determines its initial bandwidth price $p_u(t)$ and bandwidth capacity $B_0$. $\tau = 0$
repeat
  Vehicles within the coverage of each UAV determines their bandwidth requirements $b_{n_u^t}(t)$ by (8)
  if the total bandwidth requirement of vehicles is more extensive than $B_0$ then
    Each UAV updates its bandwidth price by $p_u(t)[\tau+1] = p_u(t)[\tau] + \rho_p$, where $\rho_p$ is a small value
  else
    Each UAV updates its price by $p_u(t)[\tau+1] = p_u(t)[\tau] + \varepsilon \nabla U_u(p_u(t)[\tau])$
  end if
  $\tau = \tau + 1$
until Each $pm(t)$ has no significant changes
```

4 Simulation Performance

Table 2 elaborates the parameters of our experimental environment.

Figure 3 shows the comparison result of the UAV's utility. When B_0 is fixed, the utility of the UAV in our proposed scheme is greater than that of the other two conventional schemes. In the linear-based allocation scheme, the bandwidth price is determined according to the linear pricing mechanism. As a result, this bandwidth price is not optimal, and drones cannot have the most excellent utility. In our proposed scheme, the bandwidth price is determined based on the game theory of the optimal bandwidth price.

The performance of the proposed scheme is evaluated by comparing it with the many-to-one scheme [8] scheme, the maximum signal-strength-indicator (max-RSSI) [9] scheme, the maximum signal-to-interference-plus-noise-ratio (max-SINR) [10] scheme, the Auction-Based UAV Swarm Many-to-Many scheme (AMMA) and UE-Optimal Many-to-One Matching scheme (UMOA) [11].

Table 2. Experimental parameters and value.

Parameters	Value	Parameters	Value
Square network	1000 m × 1000 m	Gain between BS and vehicle	10 mW–30 mW
Number of UAV	10, 12, 15	Gain between UAV and ground g_0	−50 db
Desired data rate	1 Mbps–15 Mbps	Path loss μ	−2
AWGN variance	10^{-14} W	Number of iterations	0.1

The system throughput among UAVs and vehicles of our scheme specifies in Fig. 4. The figure elaborates that our scheme has the highest system throughput. The second one is about 40.2% lower than ours compared with other schemes.

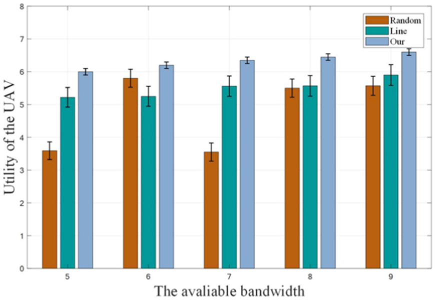


Fig. 3. Utility of the UAV

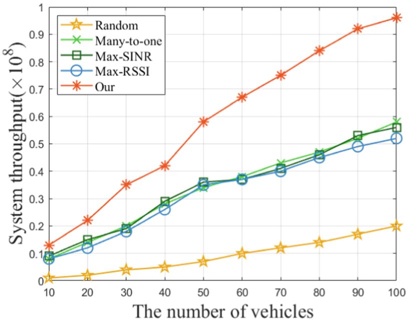


Fig. 4. System throughput

5 Conclusion

In this paper, we propose a novel security bandwidth allocation scheme based on the storage server and smart contracts in UAV-assisted VANET with game theory. Specifically, we firstly developed a secure bandwidth allocation framework. To allocate the secure bandwidth of drones, we design an iterative-based algorithm based on the needs of vehicles and the real-time bandwidth of drones, to maximize utility by Stackelberg equilibrium. Furthermore, we not only elaborate the security analysis of the smart contracts in the network, and resist fraud and malicious attacks, respectively, but also achieve privacy protection and secure bandwidth allocation. Finally, we conduct simulation experiments to verify the effectiveness of our scheme.

Acknowledgment. This work was supported in part by the R&D Program of Beijing Municipal Education Commission under Grant KM202010009010, in part by the Yunnan Key Laboratory of Blockchain Application Technology under Grant 2021105AG070005 (YNB202102), in part by the Beijing Municipal Natural Science Foundation under Grant M21029 and in part by the National Key Research and Development Program of China under Grant 2018YFB1800302.

References

1. Liang, J., Ma, M.: An efficiency-accuracy tradeoff for IDSs in VANETs with markov-based reputation scheme. In: Proceedings of ICC, pp. 1–6 (2021)
2. Bhabani, B., Mahapatro, J.: A delay-efficient channel allocation scheme for disseminating alert messages using WBAN and VANET. In: Proceedings of ICC, pp. 1–6 (2021)
3. Xu, S., Chen, X., He, Y.: EVchain: an anonymous blockchain-based system for charging-connected electric vehicles. Tsinghua Sci. Technol. **26**(6), 845–856 (2021)
4. Wen, Y., Shi, J., Zhang, Q., Tian, X., Huang, Z., Yu, H., et al.: Quality-driven auction-based incentive mechanism for mobile crowd sensing. IEEE Trans. Veh. Technol. **64**(9), 4203–4214 (2015)

5. Nguyen, C.T., Hoang, D.T., Nguyen, D.N., Niyato, D., Nguyen, H.T., Dutkiewicz, E.: Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals. Appl. Opport. IEEE Access. **7**, 85727–85745 (2019)
6. Shen, Y., Liu, Y., Yang, H., Sang, L., He, W.: Cell-cluster network-assisted adaptive streaming media optimization over wireless network. In: Proceedings of WCNC, pp. 1–6 (2021)
7. Yang, X., Zhang, H., Ji, H., Li, X.: Hybrid cooperative caching based iot network considering the data cold start. In: Proceedings of WCNC, pp. 1–6 (2021)
8. Meng, Y., Zhang, Z., Huang, Y., Zhang, P.: Resource allocation for energy harvesting-aided device-to-device communications: a matching game approach. IEEE Access. **7**, 175594–175605 (2019)
9. Elshaer, H., Kulkarni, M.N., Boccardi, F., Andrews, J.G., Dohler, M.: Downlink and uplink cell association with traditional macrocells and millimeter wave small cells. IEEE Trans. Wirel. Commun. **15**(9), 6244–6258 (2016)
10. Guvenc, I.: Capacity and fairness analysis of heterogeneous networks with range expansion and interference coordination. IEEE Commun. Lett. **15**(10), 1084–1087 (2011)
11. Zhang, Q., Wang, H., Feng, Z., Han, Z.: Many-to-many matching-theory-based dynamic bandwidth allocation for UAVs. IEEE Internet Things J. **8**(12), 9995–10009 (2021)
12. Cao, Y., Xu, S., Chen, X., He, Y., Jiang, S.: A forward-secure and efficient authentication protocol through lattice-based group signature in VANETs scenarios. Comput. Netw. **124**, 109149 (2022)
13. Cheng, Y., Xu, S., Zang, M., Jiang, S., Zhang, Y.: Secure authentication scheme for VANET based on blockchain. In: Proceedings of ICCC, pp. 1526–1531 (2021)
14. Cheng, Y., Xu, S., Zang, M., Kong, W.: LPPA: a lightweight privacy-preserving authentication scheme for the internet of drones. In: Proceedings of ICCT, pp. 656–661 (2021)
15. Xiong, Z., Cai, Z., Han, Q., Alrawais, A., Li. W.: ADGAN: protect your location privacy in camera data of auto-driving vehicles. IEEE Trans. Ind. Inform. **18**(2), 1310–1321 (2022)
16. Xiong, Z., Xu, H., Li, W., Cai., Z.: Multi-source adversarial sample attack on autonomous vehicles. IEEE Trans. Veh. Technol. **70**(3), 2822–2835 (2021)
17. Wang, J., Cai, Z., Yu., J.: achieving personalized k-anonymity based content privacy for autonomous vehicles in CPS. IEEE Trans. Ind. Inform. **16**(6), 4242–4251 (2020)
18. Cai, Z., Zheng, X.: A private and efficient mechanism for data uploading in smart cyber-physical systems. IEEE Trans. Netw. Sci. Eng. **7**(2), 766–775 (2020)

FSI: A FTM Calibration Method Using Wi-Fi Physical Layer Information

Yang Zhang[1], Bingxian Lu[1](✉), and Wei Wang[2]

[1] Dalian University of Technology, Dalian, China
bingxian.lu@dlut.edu.cn

[2] School of Intelligent Systems Engineering, Sun Yat-Sen University, Guangzhou, China
ehomewang@ieee.org

Abstract. Fine Time Measurement (FTM) protocol is included by IEEE 802.11–2016 to address the challenging problem of the high accuracy of the existing system in Wi-Fi positioning. Although FTM promises meter-level ranging accuracy in line-of-sight (LOS) conditions, non-line-of-sight (NLOS) and multipath effects cause accuracy to decline sharply. In this paper, by diving into fine-grained PHY layer information of higher time resolution, we explore the relationship deeply between FTM error and multipath channel response. On this basis, we propose FSI, a method for calibrating FTM errors using PHY layer information, which can identify environmental characteristics automatically and estimate the length of signal propagation paths. Finally, we design an optimation method based on the mobility of users, to further improve positioning accuracy in actual environments. Experimental results show that FSI improves the ranging accuracy by 24.80% and positioning accuracy by 28.45%.

Keywords: FTM · PHY · NLOS · Multipath

1 Introduction

Indoor positioning has always been an active research area. At present, ways of indoor positioning are varied, including UWB [8], Wi-Fi [12], acoustics [11], etc. In all of these, technologies that use Wi-Fi infrastructure are attracting increasing attention due to their popularity in indoor environments.

The IEEE 802.11–2016 standardizes a Fine Time Measurement (FTM) protocol, a method based on time-of-flight (TOF) for calculating the distance between Wi-Fi clients and AP. Currently, FTM is supported by many mobile devices and routers, such as the Google Pixel series, Samsung Note 10+ and Compulab WILD AP. Compared to received signal strength information (RSSI), FTM enables the expected meter-level ranging accuracy in open space [4], which determines wide applications of FTM, such as 3-D indoor localization [17] and vehicle tracking [5].

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 365–374, 2022.
https://doi.org/10.1007/978-3-031-19214-2_30

Unfortunately, many works [4,7] have shown the weakness of FTM in non-line-of-sight (NLOS) and multipath conditions. Due to the complex and changeable indoor environment, wireless signals will be affected by the multipath effect and time-varying channel characteristics. When the signal reaches the receiver through different paths, different lengths of these paths will lead to different RTTs. Generally, the signal strength of a direct path may be severely reduced in NLOS, resulting in greater RTT. In this case, a bias is generated regardless of which path the signal follows to reach the receiver. Considering a LOS condition, the direct signal component is dominant over other signal components, when FTM returns a more accurate value. In a complex NLOS environment, the superposition of multipath channel and receiver signals significantly affects the ranging accuracy. The positioning system has good accuracy as long as it can distinguish direct paths and reflected paths. Detecting them accurately requires fine-grained multipath decomposition of channels.

Given this, FTM needs to be calibrated to adapt to changes in the environment. There has been a lot of work focused on the calibration of FTM by different methods, including work based on deep learning [3], work based on geometric [10], and work based on sensor-aided [2]. All the above work requires complex calculations. FUSIC [7] explores the feasibility of calibrating FTM errors by using Multiple Signal Classification (MUSIC) to process Channel State Information (CSI), which is a feasible approach to solving multipath and calculating TOF of the direct path. The propagation model of the wireless signal can be described by CSI in detail from the perspective of a time domain and frequency domain. CSI can reflect the multipath characteristics of the channel, making it suitable for high-precision applications, for example, indoor positioning, wireless ranging [10], action recognition [1], human tracking [13] and so on. The most prominent off-the-shelf device that provides CSI information is the Intel 5300 NIC for the IEEE 802.11n standard. However, 802.11n has been around for 10 years. Newer standards such as 802.11ac and 802.11ax may provide better performance. We extract the channel state information (CSI) from the PHY layer as a fine-grained characteristic using the AX200 NIC and the Picoscence platform [6].

In this paper, we propose an error calibration model FSI based on PHY layer information, which work under both LOS and NLOS conditions, showing good positioning ability. The main contributions of this paper are as follows:

- We propose FSI, a ranging error calibration model based on PHY layer information, which can estimate distances of signal propagation paths by identifying environmental characteristics automatically.
- Combined with natural human mobility, we design an optimization method to further improve the accuracy of the length of the paths, which contributes to the application in practical scenarios.
- We evaluate the FSI in a wide range of environments with different multipath levels. We analyze the calibration capability of FSI in a corridor, an office, a classroom, and a laboratory, and extensive experiments show that FSI significantly improves the ability to provide meter-level indoor positioning.

2 Preliminary

The Relationship of Ranging Error and PHY Layer Information Under Different Conditions. As shown in Fig. 1, when the reflector moves, we can see the changes in characteristics in Fig. 2. We analyze the effect of multipath on ranging in three dimensions, which are the number, the relative signal strength, and the time delay difference of paths, respectively.

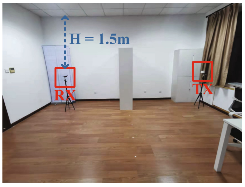


Fig. 1. Multipath setup with a moving reflector.

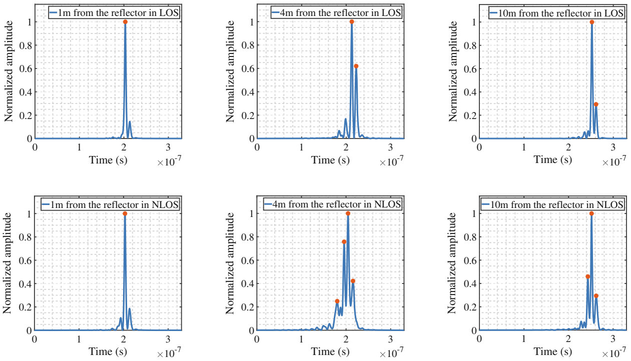


Fig. 2. Changes in characteristics of signal at 160 MHz.

1) *The number of paths changes.* When the difference between the reflected path and the direct path exceeds the resolution of the channel, the reflected path can be distinguished, resulting in a larger ranging error.
2) *The relative signal strength of paths changes.* In indoor environments, due to the diversity of signal propagation paths, the absolute signal strength of the paths often does not reflect the environmental characteristics well. However, practical experiments show the relative signal strength ratio of the direct path shows a trend similar to the error variation as the reflector moves. In the case of maximum error, the relative signal strength ratio of the direct path is smaller.
3) *The time delay difference of paths changes.* As the reflector moves away, the time delay difference between the direct path and the other reflected paths gradually becomes larger. Although the estimated distances of the direct and reflected paths are highly erroneous, the distance differences between the two path lengths correspond to the actual differences of the path lengths [7].

3 System Design

3.1 Overview

This section illustrates a calibration model that fuses Wi-Fi FTM and CSI to provide accurate ranging even in the presence of multipath. It takes as input raw FTM values, IMU and the CSI matrix, and finally returns the distance of a transceiver pair. The system overview for the model is shown in Fig. 3.

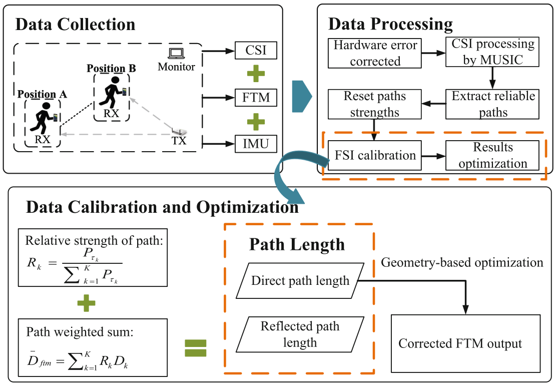
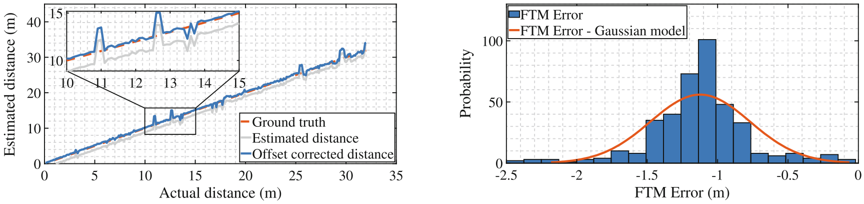


Fig. 3. System overview of FSI.

3.2 Data Processing

Hardware Error Calibration. We place equipments in a playground and measure their distances continuously while changing the actual distance between them from 0.1 m to 32 m. After fitting the curve using a normal distribution in Fig. 4(b), the distances at all different positions are underestimated by about $\mu = -1.12$ m and $\sigma = 0.35$ m. To better cope with the actual environment, we choose -1.12 m as the hardware error. The solid gray line in Fig. 4(a) indicates the estimated distance before correction and the solid blue line indicates the distance after correction. The corrected distance is closer to the ground truth.



(a) Accuracy of FTM ranging in LOS.
(b) FTM error distribution.

Fig. 4. FTM ranging capability and hardware error correction by Gaussian model.

CSI Processing by MUSIC and TOF Estimation. CSI matrix can be losslessly converted into the time-domain power delay profile (PDP) by an appropriate MUSIC [7,15] processing.

Extract Reliable Paths and Reset Paths Strengths. Since the direct path traverses the smallest distance of all received paths, its strength is likely to be present in the earliest component. Actually, different noises often occur in raw CSI data, leading to the emergence of some low-quality strength peaks. First we apply Min-Max Normalization to all paths,

$$P(\tau_k) = \frac{path\_power(\tau_k) - min(path\_power)}{max(path\_power) - min(path\_power)} \qquad k \in K, \tag{1}$$

where K is the number of paths and τ_k is ToF. We consider paths satisfying $P(\tau_k) \geq \xi$ as reliable paths, where ξ is a threshold value for classification. In this paper, we set $\xi = 0.2$. In fact, by this method, the number of signal propagation paths is usually less than 5 because typically we see at best five significant paths in an indoor environment [14].

When we select reliable paths by setting a threshold, it leads to a possible result that the relative strengths of some signals are overestimated. We define a function for redividing signal strengths into three levels according to $P(\tau_k)$,

$$P^*(\tau_k) = \omega_k P(\tau_k), \tag{2}$$

where k is the k^{th} path. When $P(\tau_k)$ belongs to $[0, 0.2]$, it is reassigned to 0. When $P(\tau_k)$ belongs to $(0.2, \delta]$, it is redefined as $\omega_k P(\tau_k)$, where $\omega_k = 1 - \frac{\tau_k - \tau_1}{\sum_{k=1}^{K}(\tau_k - \tau_1)}$. Please note that δ is an experience threshold and is set to 0.8, which can achieve better performance compared to other parameters. We define here a weight coefficient ω_k related to the time delay difference, which assigns a lower weight to the signal strength for a longer reflected path. When $P(\tau_k)$ belongs to $(\delta, 1]$, it reserves its own value $P(\tau_k)$.

3.3 FSI Calibration and Results Optimization

The Length of Paths Estimation. FTM is between direct distance and reflected distance [7], so we have got an assumption from the experimental results above: the FTM mean output is the result of multipath interaction, and paths with higher relative signal strength always play a dominant role. The strength ratio of the k^{th} path is defined as :

$$R_k = \frac{P^*(\tau_k)}{\sum_{k=1}^{K} P^*(\tau_k)}. \tag{3}$$

We consider $\hat{R}_k$ as a weight of length of each path D_k, which finally return a mean value $\bar{D}_{ftm}$ of sample datas of FTM. For $\hat{R}_k$, we define its function: $\hat{R}_k = F(R_k)$, where F is an increasing function and $\hat{R}_k$ is the result of the function change on R_k. $\hat{R}_k$ and R_k may be non-linearly dependent. Here we assume that $\hat{R}_k$ and R_k are directly proportional and the coefficient is 1,

$$\bar{D}_{ftm} = \sum_{k=1}^{K} R_k D_k. \tag{4}$$

Although estimated distances of the direct and reflected paths are highly erroneous, their distance differences correspond to the actual differences [7]. So the time difference of two paths corresponds to the actual time difference:

$$D_k = D_1 + (\tau_k - \tau_1)c, \tag{5}$$

where c is the speed of light. From the above formulas, we can resolve the length of each path D_k.

Triangle Inequality-based to Filter Ranging. Given a fixed AP, we can leverage the mobility of users to collect multiple measurements over different locations. To filter out estimated distances that are inconsistent with the displacement, additional constraints need to be imposed on them. The appropriate value can be selected by using the triangular inequality:

$$|D_{ftm}(A) - D_{imu}| \leq D_{ftm}(B) \leq D_{ftm}(A) + D_{imu}. \tag{6}$$

For ranging values that do not satisfy the triangular inequality, we do the following: if $D_{ftm}(B) < |D_{ftm}(A) - D_{imu}|$, $D_{ftm}(B) = |D_{ftm}(A) - D_{imu}|$; if $D_{ftm}(B) > D_{ftm}(A) + D_{imu}$, $D_{ftm}(B) = D_{ftm}(A) + D_{imu}$.

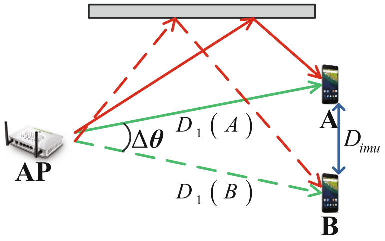


Fig. 5. Optimize D_1 error by using mobility of users.

Distance Optimization Combined with IMU. We can calculate the actual distance between two positions by accessing the phone's inertial measurement unit (IMU) such as accelerometer and gyroscope, a method known as dead reckoning. Although IMU-based ranging results in some errors, its accuracy is still very high when we make a small movement ($< 5m$) [9,16].

We use the Fig. 5 to explain our idea. When the user moves from position A to position B, the angle of movement at AP end is noted as $\Delta\theta$, and by the nature of the triangle, $\Delta\theta$ can be calculated: $\Delta\theta = arccos\frac{(D_{ftm}(A))^2+(D_{ftm}(B))^2-(D_{imu})^2}{2D_{ftm}(A)D_{ftm}(B)}$, where $D_{ftm}(A)$ and $D_{ftm}(B)$ are the measured FTM values at A and B positons. $\Delta\theta$ can be substituted into the following equation to obtain $\hat{D}_{imu}$:

$$\hat{D}_{imu} = \sqrt{(D_1(A))^2 + (D_1(B))^2 - 2D_1(A)D_1(B)cos\Delta\theta}. \tag{7}$$

We assume that a small range of shifts makes a similar impact for D_{imu} and D_{fsi} that are influenced by multipath. Then we can obtain the error on each meter units: $\sigma = \frac{|\hat{D}_{imu} - D_{imu}|}{D_{imu}} = \frac{|D_1 - D_{fsi}|}{D_{fsi}}$. It is less susceptible to multipath reflections, especially when D_1 is generated by a precise and high time resolution. Thus we can obtain the calibrated range values. However, we obtain two values. Since the value of D_1 is higher than the ground truth in NLOS and multipath, we choose the smaller D_{fsi} of the two,

$$D_{fsi} = \frac{D_1}{1+\sigma}. \tag{8}$$

3.4 Localzation Methods

Since we deploy multiple responders in each scenario, we use Weighted Least-Squares (WLS) to calculate the location of the initiators,

$$\widehat{\boldsymbol{R}} = \arg\min_{\boldsymbol{R}} \sum_{i=1}^{N+1} \beta_i \left(\|\boldsymbol{R} - \boldsymbol{R_i}\| - \hat{d}_{fsi}(i) \right)^2, \tag{9}$$

where $N+1$ is the total number of APs, β_i is a weight constant of i th AP with $\beta_i \geq 0, \boldsymbol{R} = [x, y]^T$ is the actual user location, $\boldsymbol{R}_i = [x_i, y_i]^T$ is the i^{th} location of the AP, and $\widehat{\boldsymbol{R}} = [\hat{x}, \hat{y}]^T$ is the estimated location. We select a reference AP as the smallest measured distance among all the distance measurements by a Reference Selection (LLS-RS) [3]: $(\|\boldsymbol{R} - \boldsymbol{R_i}\|)^2 - (\|\boldsymbol{R} - \boldsymbol{R_r}\|)^2 = (\hat{d}_{fsi}(i))^2 - (\hat{d}_{fsi}(r))^2$, where r is r^{th} AP and $i = 1, 2, \cdots, N+1 (i \neq r)$. We can get re-arranging matrix form as follows:

$$\boldsymbol{WAR} = \frac{1}{2}\boldsymbol{WB}, \tag{10}$$

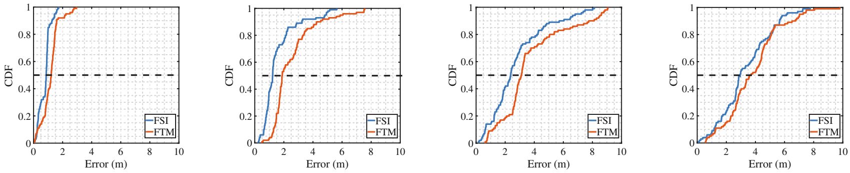
where $\boldsymbol{W} = diag\{\beta_1, \beta_2, \cdots, \beta_{N+1}\}$. Due to R_1 can reflect the magnitude of the error, we determine the weight constant β based on the strength value of direct path R_1 in Eq. 3. When $R_1 > 0.5$, $\beta = 1$. When $0.1 < R_1 < 0.5$, $\beta = 0.5$. When $R_1 < 0.1$, $\beta = 0$. Finally, we can get the optimally estimated coordinates (x, y).

4 Performance Evaluation

4.1 Evaluation Setup

We use Google Pixel 2 phone and a Google Wi-Fi router as the transceiver. They operate at 80 MHz bandwidth and 5.21 GHz center frequency. We use a Dell Vostro 3000 series computer with an AX200 NIC. A Xiaomi AX3000 router is configured at the receiving end. The Xiaomi router operates at 160 MHz bandwidth and 5.25 GHz center frequency. In each antenna, we can obtain a CSI matrix with 2025 sub-carriers, whose carrier bandwidth 78125 Hz.

4.2 Ranging Error



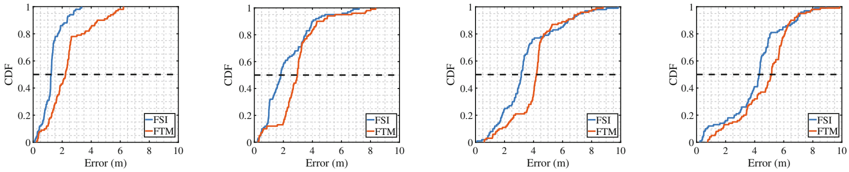
(a) Corridor. (b) Office. (c) Classroom. (d) Laboratory.

Fig. 6. FSI and FTM ranging accuracy in four real indoor rooms.

We set up 20 tested locations in each spatial environment. At each location, we collect 50 consecutive FTM and take the mean value as well as the CSI matrix. After collecting the above data, we move the initiator in a small area to collect again 50 FTM values and the range values returned by the IMU. We focus the evaluation data set on having a common setup with different movement patterns, rather than just moving along a straight line in the same direction.

Taking all the data into account in Fig. 6, FSI achieves a median and 90-percentile of 1.85 m and 3.88 m respectively, outperforming FTM. **The ranging accuracy of FSI improved by 24.80% for the median and 21.14% for the 90-percentile.**

4.3 Positioning Error



(a) Corridor. (b) Office. (c) Classroom. (d) Laboratory.

Fig. 7. FSI and FTM localization accuracy in four real indoor rooms.

In addition to evaluating the object measurement error, we also evaluate the positioning error capability of FSI in the same environment and location. By comparing the output location with the known ground truth, Fig. 7 gives the CDF of the position estimation error for each spatial environment. Overall, taking all the location estimates into account, FSI achieves a median and 90-percentile of 2.59 m and 4.85 m respectively. **The positioning accuracy of FSI improved by 28.45% for the median and 10.35% for the 90-percentile.**

5 Conclusion and Future Directions

In this study, we explore the relationship between PHY layer information and FTM error in a high resolution and present FSI, an error calibration model uses PHY layer information to correct the FTM. Moreover, we use the mobility of the target to optimize the results, which makes FSI suitable for actual environments. FSI can be implemented as a standalone application on mobile devices. Extensive experimental evaluations have validated the feasibility of FSI. As part of future work, we plan to further improve FTM accuracy by fusing multiple sources of information, such as AOA and Doppler frequency shift, and apply them to device tracking. As more Wi-Fi chipsets in mobile devices support larger bandwidth transmission, the ranging accuracy of FTM will also greatly improve.

Acknowledgements. The work was supported by National Nature Science Foundation of China with No. 61902052 and 62027826, "Science and Technology Major Industrial Project of Liaoning Province" with No. 2020JH1/10100013, "Dalian Science and Technology Innovation Fund" with No. 2020JJ26GX037, and "the Fundamental Research Funds for the Central Universities" with No. DUT20TD107.

References

1. Chen, Z., Zhang, L., Jiang, C., Cao, Z., Cui, W.: Wi-Fi CSI based passive human activity recognition using attention based BLSTM. IEEE Trans. Mob. Comput. **18**(11), 2714–2724 (2018)
2. Choi, J.: Enhanced Wi-Fi RTT ranging: a sensor-aided learning approach. IEEE Trans. Vehicular Technol. **71**(4), 4428–4437 (2022)
3. Han, K., Yu, S.M., Kim, S.L.: Smartphone-based indoor localization using Wi-Fi fine timing measurement. In: IPIN (2019)
4. Ibrahim, M., et al.: Verification: accuracy evaluation of Wi-Fi fine time measurements on an open platform. In: MobiCom (2018)
5. Ibrahim, M., et al.: Wi-Go: accurate and scalable vehicle positioning using Wi-Fi fine timing measurement. In: MobiSys (2020)
6. Jiang, Z., et al.: Eliminating the barriers: demystifying Wi-Fi baseband design and introducing the PicoScenes Wi-Fi sensing platform. IEEE Internet Things J. **9**(6), 4476–4496 (2022)
7. Jiokeng, K., Jakllari, G., Tchana, A., Beylot, A.L.: When FTM discovered MUSIC: accurate Wi-Fi-based ranging in the presence of multipath. In: Infocom (2020)
8. Ma, Y., Selby, N., Adib, F.: Minding the billions: ultra-wideband localization for deployed RFID tags. In: MobiCom (2017)
9. Sen, S., Lee, J., Kim, K.H., Congdon, P.: Avoiding multipath to revive inbuilding Wi-Fi localization. In: MobiSys (2013)
10. Shao, W., Luo, H., Zhao, F., Tian, H., Yan, S., Crivello, A.: Accurate indoor positioning using temporal-spatial constraints based on Wi-Fi fine time measurements. IEEE Internet Things J. **7**(11), 11006–11019 (2020)
11. Shen, S., Chen, D., Wei, Y.L., Yang, Z., Choudhury, R.R.: Voice localization using nearby wall reflections. In: MobiCom (2020)
12. Soltanaghaei, E., Kalyanaraman, A., Whitehouse, K.: Multipath triangulation: decimeter-level Wi-Fi localization and orientation with a single unaided receiver. In: MobiSys (2018)

13. Xie, Y., Xiong, J., Li, M., Jamieson, K.: mD-Track: leveraging multi-dimensionality for passive indoor Wi-Fi tracking. In: MobiCom (2019)
14. Xiong, J., Jamieson, K.: Arraytrack: a fine-grained indoor location system. In: NSDI (2013)
15. Xiong, J., Sundaresan, K., Jamieson, K.: Tonetrack: leveraging frequency-agile radios for time-based indoor wireless localization. In: MobiCom (2015)
16. Yu, Y., Chen, R., Chen, L., Guo, G., Ye, F., Liu, Z.: A robust dead reckoning algorithm based on Wi-Fi FTM and multiple sensors. Remote Sens. **11**(5), 504 (2019)
17. Yu, Y., et al.: Precise 3-D indoor localization based on Wi-Fi FTM and built-in sensors. IEEE Internet Things J. **7**, 11753–11765 (2020)

Low-Poisoning Rate Invisible Backdoor Attack Based on Important Neurons

Xiu-gui Yang, Xiang-yun Qian, Rui Zhang, Ning Huang, and Hui Xia(✉)

College of Computer Science and Technology, Ocean University of China,
Qingdao 266100, China
xiahui@ouc.edu.cn

Abstract. The present research on label-consistent invisible backdoor attacks mainly faces the problem of needing a high poisoning rate to achieve a high attack success rate. To address the problem, this paper proposes a low-poisoning rate invisible backdoor attack based on important neurons (INIB) by enhancing the connection between triggers and target labels with the help of the neural gradient ranking algorithm. The method first identifies the neurons with the most significant influence on the target label with the help of the neural gradient ranking algorithm, secondly establishes a strong link between the important neurons and the trigger using the gradient descent algorithm, and then generates a trigger based on the established strong link by minimizing the difference between the current activation value and the expected activation value of the important neurons, thus causing the important neurons to be strongly activated when images have the trigger, which in turn causes the model to misidentify them as the target label. Finally, detailed experimental results show that INIB is able to achieve a very high attack success rate with a very low poisoning rate. Specifically, INIB achieves a 98.7% backdoor attack success rate with the poisoning rate of only 1.64% on the MNIST dataset.

Keywords: Invisible backdoor attack · Label consistent · Low-poisoning rate · Important neuron

1 Introduction

Deep neural networks (DNNs) have been used in a wide range of real-world applications [1]. However, a large amount of recent research has shown that DNNs are highly vulnerable to backdoor attacks. Backdoor attacks [2] are a class of attacks that inject hidden malicious behaviors into DNNs by manipulating certain neurons to make the DNN misidentify a particular sample. The two main

Supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62172377 and 61872205, and the Natural Science Foundation of Shandong Province under Grant No. ZR2019MF018.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 375–383, 2022.
https://doi.org/10.1007/978-3-031-19214-2_31

types of attacks are poisoning the training dataset and directly modifying the model parameters, and the contaminated model is called a backdoor model. For clean data samples, the backdoor model will identify them correctly, whereas, for samples with triggers crafted by attackers, the backdoor model will be triggered to invoke predefined malicious behavior, resulting in incorrect identification. For example, a tainted autopilot system would recognize a stop sign with a trigger as speed limit recognition, causing the autopilot system not to apply the brakes [3], which would pose a serious threat to the lives of passengers and passers-by. To make matters worse, backdoor attacks only require the manipulation of a very small number of neurons to embed backdoors, which results in backdoors in DNNs being difficult to detect. Designing a backdoor attack scheme contributes to the proposed backdoor defense method, which can reverse the security of DNN models and thus improve the security of real-world applications [4].

The existing backdoor attack methods are mainly divided into two categories: non-poisoning-based backdoor attacks and poisoning-based backdoor attacks. The former refers to the injection of malicious behaviors directly into the model, which is mainly achieved by modifying training parameters or model weights but is very difficult to implement and difficult to apply in practice. The latter is the training of a model using a poisoned training set and occurs mainly during the training phase of the model. Poisoning-based backdoor attacks are further divided into two categories: trigger-visible backdoor attacks and invisible backdoor attacks. Trigger-visible backdoor attacks, in which the attacker uses poisoned images with obvious triggers to train the model, can make deep learning models misidentify inputs with malicious triggers, but are difficult to apply in practice because they are highly detectable. Invisible backdoor attacks, in which an attacker uses poisoned images with obscure triggers to train a model, are divided into two categories: label inconsistent and label consistent. The inconsistently labeled backdoor attacks refer to an attacker using the steganography or distorted image to achieve trigger invisibility, but because the label of the poisoned image does not match its real label, it is difficult to evade manual visual inspection. In order to solve the above problem, label-consistent invisible backdoor attacks have emerged, but the attack success rate of such attacks is low and usually requires a high poisoning rate to achieve a high success rate of backdoor attacks.

To solve the above problem, this paper proposes a low-poisoning rate invisible backdoor attack based on important neurons (INIB) by enhancing the connection between triggers and target labels with the help of the neural gradient sorting algorithm. The main contributions are as follows:

(1) In order to reduce the poisoning rate of samples, this paper proposes a trigger generation algorithm based on important neurons, which achieves the effect of achieving a high attack success rate with a very low poisoning rate. The method identifies the neuron with the most significant influence on the target label with the help of a neural gradient ranking algorithm and then establishes a strong connection between this neuron and the trigger with the help of a gradient descent algorithm so that the important neuron

has strong activation when the trigger is present, enhancing the connection between the trigger and the target label.

(2) To verify the effectiveness of the above scheme, we compare INIB to BadNets and Hidden with the help of the backdoor attacks success rate. The experimental results show that while INIB ensures stealthiness, and can achieve a higher success rate of backdoor attacks with a very low poisoning rate. For example, on the MNIST dataset, only 1.64% of the poisoning rate is needed to achieve 98.7% of the attack success rate.

2 Related Work

This section describes the current state of research on backdoor attacks in terms of both poisoning-based backdoor attacks and non-poisoning-based backdoor attacks.

2.1 Poisoning-Based Backdoor Attacks

(1) Trigger-visible backdoor attacks. Gu *et al.* [5] first proposed BadNets, which generate poisoned images by directly hitting the trigger on some of the benign images, and then use the benign images together with the poisoned images to train the model. Xue *et al.* [6] used poisoned images with triggers as well as their compressed versions to generate a poisoned training set in order to avoid corrupted features of the triggers.

(2) Invisible backdoor attacks. Barni *et al.* [7] proposed a clean-label invisible backdoor attack, which generates poisoned images with labels consistent with their real labels. Saha *et al.* [8] proposed a label-consistent hidden trigger backdoor attack in order to evade manual visual inspection. Nguyen *et al.* [9] proposed a WaNet based on image warping, whose attack is mainly achieved by a small and smooth warping field.

2.2 Non-poisoning-based Backdoor Attacks

Non-poisoning-based backdoor attacks are mainly implemented by directly modifying the training parameters or model weights. Clements *et al.* [10] proposed a method to embed a backdoor by modifying certain computational operations in a neural network by assuming that the attacker has full access to the model. Dumford *et al.* [11] proposed a method to directly modify the model weights in a neural network by using a greedy algorithm to search for the target weights. Bagdasaryan *et al.* [12] proposed a backdoor injection technique that uses the loss value computation during the training process. Salem *et al.* [13] proposed triggerless backdoor attacks that alter the functionality of the model and generate specific target labels by removing some target neurons during training. The attacker can trigger the backdoor by deleting these neurons.

3 Threat Model Definition

The classifier is denoted as $F_w : X \to [0,1]^{|Y|}$, where w is the model parameter, $X \subset R^d$ is the instance space and $Y = \{1, 2, ..., M\}$ is the label space. $F(x)$ denotes the posterior vector relative to the class M, $C(x) = \arg\max f_w(x)$ denotes the prediction label, y_t denotes the target label, $D_L = \{(x_i, y_i)|i = 1, ..., N_l\}$ denotes the training data set, and $D_{sL} = \{(x, y) \subset D_L\}$ denotes a subset of D_L.

Backdoor security value S_b: a measure of whether the trigger can successfully activate a hidden backdoor in the classifier, i.e. the success rate of the backdoor attack.

$$S_b(D_{sL}) = E_{(x,y)\sim P_{D_{sL}}} \left[\mathrm{I}\left\{ C(x') = y_t \right\} \right] \tag{1}$$

Model test safety value S_{t}: a measure of model availability, i.e. test accuracy of the model.

$$S_t(D_L) = E_{(x,y)\sim P_{DL}} \left[\mathrm{I}\left\{ C(x) = y \right\} \right] \tag{2}$$

Objective function:

$$MAX_w \lambda_1 \cdot S_b(D_{sL}) + \lambda_2 \cdot S_t(D_L) \tag{3}$$

where x represents the data in the training dataset, x' represents the data with triggers, λ_1, λ_2 are two non-negative trade-off hyperparameters, $\frac{|D_{sL}|}{|D_L|}$ is the proportion of poisoning, $E_{(x,y)\sim P_{D_{sL}}}$ is the mathematical expectation, and the I function represents the result of 1 if the latter condition is true.

4 INIB

The scheme in this paper is divided into three main parts: trigger generation based on important neurons, label-consistent poisoned image generation, and model retraining.

4.1 Trigger Generation Based on Important Neurons

Dentify Important Neurons. Assume that the DNN model F has M output classes, $T \in \{1, 2, ..., M\}$ is the target label of the attack, and the last layer of the model F is a fully connected layer classifier with N output neurons and M input neurons.

The classifier has a weight matrix of $W \in R^{M\times N}$ and a loss function of Γ for model F. For a given set of input samples and their labels, the gradient is calculated by backpropagation. Noting the ownership value connected to the T output neuron as g_{Ti}, the cumulative gradient is described as:

$$G = \frac{\partial \Gamma}{\partial W} = \begin{pmatrix} g_{11} & \cdots & g_{1N} \\ \cdots & \cdots & \cdots \\ g_{T1} & \cdots & g_{TN} \\ \cdots & \cdots & \cdots \\ g_{M1} & \cdots & g_{MN} \end{pmatrix} \tag{4}$$

The neuron with the most significant effect on the output of the target label T is then identified and using the neural gradient ranking algorithm can be represented as:

$$\begin{cases} MAX_{w_b} \left| \left[g_{T1}, g_{T2}, \ldots, g_{TN} \right] \right| \\ w_b < N \end{cases} \tag{5}$$

where the above function returns the index $\{j\}$ of the gradient g_{Tj} neuron with the highest absolute value connected to the T output neuron of the last layer, and the value of the returned index also corresponds to the weight.

Trigger Generation. Establishing a strong connection between the initial trigger and a previously selected set of internally important neurons, thus generating a final trigger such that the important neurons have strong activation in the presence of that trigger.

The trigger generation algorithm uses gradient descent to find a local minimum of the loss function and iteratively improves the input in the direction of the decreasing loss function based on the initially assigned value, so that the final activation of the selected neuron is as close as possible to the expected activation. The loss function is defined as:

$$\cos t = \frac{1}{m} \sum_{i=1}^{m} \begin{matrix} \mathrm{I}_{|neuron_i - \text{target\_}v_i| \le \tau} \frac{(neuron_i - \text{target\_}v_i)^2}{2} + \\ \mathrm{I}_{|neuron_i - \text{target\_}v_i| > \tau} \left(\tau \left| neuron_i - \text{target\_}v_i \right| - \frac{1}{2}\tau^2 \right) \end{matrix} \tag{6}$$

τ in the above equation is a hyperparameter. In Algorithm 1, F denotes the model, p is the trigger, the threshold is the threshold for the termination process, epochs is the maximum number of iterations, Lr is the learning rate, $\{(\text{neuron}_1, \text{target\_v}_1), (\text{neuron}_2, \text{target\_v}_2), \cdots\}$ denotes the set of important neurons selected and the neuron activation values.

4.2 Label-Consistent Poisoned Image Generation

In order to generate consistently labeled poisoned images, we optimize the image with the help of a feature space optimization algorithm. Let $f(x)$ denote the function that propagates x through the neural network to the penultimate layer, and call the activation function at this layer the feature space representation of the input.

The process of attaching the trigger to the source image is represented as:

$$s' = (1 - \partial) \otimes s + \partial \otimes p \tag{7}$$

Optimize the poisoned image by solving the following objective function z:

$$\begin{aligned} & z = \arg\min_{x} \parallel f(x) - f(s') \parallel_2^2 + \mu \parallel x - t \parallel_2^2 \\ & s.t., ||f(x) - f(s')||_2 \le \sigma \text{and} ||x - t||_2 \le \mu \end{aligned} \tag{8}$$

where s is the source image, p is the trigger generated in the previous step, t is the target image, and z is the poisoned image. $\otimes$ refers to pressing the image directly, ∂ is a parameter to balance the degree of paste of the trigger, and μ is a parameter used to balance the similarity relationship between vision and feature space.

At step i of the optimization algorithm, the following is satisfied:

$$\begin{cases} ||f(x)-f(x_{i-1})||_2 \leq \sigma_i \\ ||x-x_{i-1}||_2 \leq \mu_i \end{cases} \tag{9}$$

and the poisoned image generated at each step lies within the input domain: $x_i + \eta_i \in [0, 255]$. We use a forward-backward splitting iterative process [14], which first optimises the $||f(x) - f(x_{i-1})||_2$ using gradient descent, adjusts the coefficients μ so that they satisfy the constraints in $||x - x_{i-1}||_2 \leq \mu_i$, and then iterates.

4.3 Model Retraining

During model retraining, we only fine-tune the parameters of the feature space layer, and we use gradient descent algorithms such as RMSprop and Adam to update the parameters during model training. Gradient descent uses local gradient information to update the parameters and gradually approximates the extreme value point of the objective function.

RMSprop:

$$\begin{aligned} v_{t+1} &= \beta v_t + (1-\beta) g_t^2 \\ \theta_{t+1} &= \theta_t - \tfrac{\eta}{\sqrt{v_{t+1}+\varepsilon}} \end{aligned} \tag{10}$$

Letting $v_0 = 0$, we obtain:

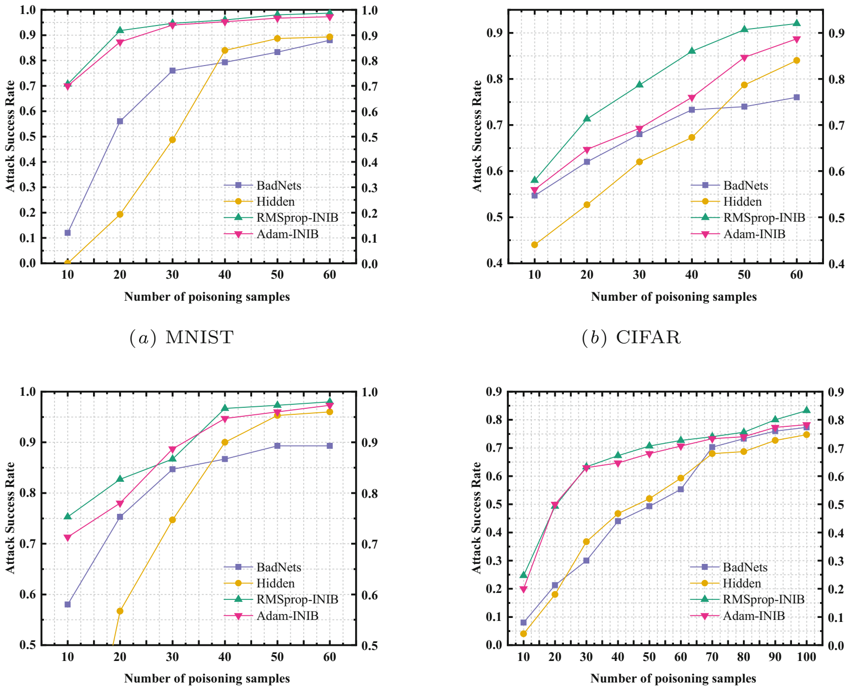
$$v_{t+1} = (1-\beta) \sum_{i=0}^{t} \beta^{t-i} g_i^2 \tag{11}$$

where η is the learning rate, β is a parameter to control the exponentially weighted average, and g_t is the gradient depends entirely on the gradient of the current batch, so η can be interpreted as how much of the gradient of the current batch is allowed to affect parameter updates. RMSprop avoids the problem of too small a coefficient for later updates by adding an exponential decay factor to the squared gradient, and v_{t+1} is often referred to as the exponentially weighted mean.

Adam:

$$\theta_{t+1} = \theta_t - \eta\left(\frac{m_{t+1}}{\sqrt{v_{t+1}+\varepsilon}} + \lambda\theta_t\right) \tag{12}$$

Neural network models are prevented from overfitting by regularization. For Adam algorithm, if the model uses L_2 regularization, its gradient at moment t is: $g_t = \nabla_\theta J(\theta_t) + \lambda\theta_t$, where λ is the weight decay. Adam adds the weight decay term to the gradient update, while g_t remains unchanged.



(*a*) MNIST

(*b*) CIFAR

(*c*) GTSRB

(*d*) ImageNet

Fig. 1. Comparison of the attack success rate

5 Experiment

In this section, we introduce the datasets and experimental parameters used for the experiments in Sect. 5.1. In Sect. 5.2, we verify that this paper's scheme INIB can achieve a high attack success rate with a very low poisoning rate by comparing it with BadNets and Hidden backdoor attack success rate.

5.1 Experimental Parameter Settings

In this paper, four commonly used datasets (i.e., MNIST, CIFAR, GTSRB, ImageNet) are chosen for backdoor attack studies, using the fc7 feature embedding f(.) of the AlexNet [15]. We use AlexNet as the base network with the remaining layer weight parameters unchanged and fine-tune the parameters of the fc8 layer. In the model retraining phase, we trained for 50 cycles, using 30 images for testing.

5.2 Attack Success Rate

To compare the success rate of backdoor attacks, we compare INIB with the trigger-visible backdoor attack method BadNets [5] and the label-consistent invisible backdoor attack method Hidden [8].

In the MNIST dataset, we chose the source image class as 1, 2, 5 and the target image class as 0, and train a multi-classifier. In this experiment, we generate 60 poisoned images and add the different numbers of poisoned images to the training set, where the amount of clean images in the poisoning training set is 3600.

In the CIFAR dataset, we chose the source image class as bird, kitten, dog, and the target image class as aircraft, and train a multi-classifier. In this experiment, we generate 60 poisoned images and add the different numbers of poisoned images to the training set, where the amount of clean images in the poisoning training set is 3600.

In the GTSRB dataset, we chose the source image categories of speed limit 50, speed limit 80, speed limit 100, and the target image category of parking, and train a multi-classifier. In this experiment, we generate 60 poisoned images and add the different numbers of poisoned images to the training set, where the amount of clean images in the poisoning training set is 3860.

In the ImageNet dataset, we chose the source image categories of car, Walkman, archway, and the target image category of dog, and train a multi-classifier. In this experiment, we generate 100 poisoned images and add the different numbers of poisoned images to the training set, where the amount of clean images in the poisoning training set is 1600.

Figure 1(a)–(d) show the comparison of INIB with BadNets, Hidden on the MNIST, CIFAR, GTSRB and ImageNet dataset, where RMSprop-INIB means that in our scheme the RMSprop algorithm is used in the model during the training phase, and Adam-INIB means that in our scheme the Adam algorithm is used in the model during the training phase.

6 Conclusion

To solve the problem that label-consistent invisible backdoor attacks require a high poisoning rate, this paper proposes a low poisoning rate, label-consistent invisible backdoor attack scheme based on important neurons. In this model, the poisoned images are correctly labeled, while the triggers are not visible in the training phase and are not easily noticed in the testing phase. Experiments show that the scheme proposed in this paper requires very few poisoned images to poison the entire network and that the model has a negligible impact on the original task.

We hope that the research in this paper will deepen our understanding of DNNs and reverse the proposed backdoor defense methods, thus better enhancing the security of deep learning models and improving the security of real-world applications.

Acknowledgements. This research is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62172377 and 61872205, and the Natural Science Foundation of Shandong Province under Grant No. ZR2019MF018.

References

1. Cai, Z., Zheng, X.: A private and efficient mechanism for data uploading in smart cyber-physical systems. IEEE Trans. Netw. Sci. Eng. **7**(2), 766–775 (2020)
2. Li, Y., Wu, B., Jiang, Y., Li, Z., Xia, S.T.: Backdoor learning: a survey (2020)
3. Cai, Z., Xu, Z., Wang, J., He, Z.: Private data trading towards range counting queries in internet of things. IEEE Trans. Mobile Comput. 1 (2022)
4. Zheng, X., Cai, Z.: Privacy-preserved data sharing towards multiple parties in industrial IoTs. IEEE J. Sel. Areas Commun. **38**(5), 968–979 (2020)
5. Gu, T., Liu, K., Dolan-Gavitt, B., Garg, S.: BadNets: evaluating backdooring attacks on deep neural networks. IEEE Access **7**, 47230–47244 (2019)
6. Xue, M., Wang, X., Sun, S., Zhang, Y., Wang, J., Liu, W.: Compression-resistant backdoor attack against deep neural networks (2022)
7. Barni, M., Kallas, K., Tondi, B.: A new backdoor attack in CNNs by training set corruption without label poisoning. In: IEEE Internation Conference on Image Processing (2019)
8. Saha, A., Subramanya, A., Pirsiavash, H.: Hidden trigger backdoor attacks. In: AAAI 2020 - Main Technical Track (Oral) (2019)
9. Nguyen, A., Tran, A.: WaNet - imperceptible warping-based backdoor attack (2021)
10. Clements, J., Lao, Y.: Backdoor attacks on neural network operations. In: 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP) (2018)
11. Dumford, J., Scheirer, W.: Backdooring convolutional neural networks via targeted weight perturbations (2018)
12. Bagdasaryan, E., Shmatikov, V.: Blind backdoors in deep learning models (2020)
13. Salem, A., Backes, M., Zhang, Y.: Don't trigger me! a triggerless backdoor attack against deep neural networks (2021)
14. Goldstein, T., Studer, C., Baraniuk, R.: A field guide to forward-backward splitting with a FASTA implementation. Computer Science (2016)
15. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25 (2012)

A Multimodal Deep Fusion Network for Mobile Traffic Classification

Shuai Ding[1,2], Yifei Xu[1,2], Hao Xu[3], Haojiang Deng[4], and Jingguo Ge[1,2](✉)

[1] School of Cyber Security, University of Chinese Academy of Sciences,
Beijing 100049, China

[2] Institute of Information Engineering, Chinese Academy of Sciences,
Beijing 100093, China
{dingshuai,gejingguo}@iie.ac.cn

[3] Department of Network and Information Security Management,
China Telecom Corporation Limited, Beijing 100010, China
xuhao@chinatelecom.cn

[4] Peng Cheng Laboratory, Shenzhen 518000, China
denghj@dsp.ac.cn

Abstract. With the explosive growth of mobile traffic and the demand for privacy protection and network security, mainstream mobile applications use encryption protocols (mostly TLS), so identifying mobile encrypted traffic has become critical. Machine learning methods are based on hand-designed features that are unreliable in the face of complex traffic data. Deep learning currently performs well on this task, but most of them only describe traffic data from one view, ignoring the heterogeneous nature of traffic. In this paper, we apply multimodal Transformers to mobile encrypted traffic classification and propose a novel model (DF-Net) with a deep fusion mechanism. The key point of deep fusion is that a learnable modal-type embedding enables the model to perform early and unconstrained fusion and interaction of cross-modal information to achieve performance improvements. On the premise of ensuring performance, DF-Net adopts lightweight design and the parallel mechanism to improve the overall efficiency of the model. To verify the performance and efficiency of DF-Net, we implement an automated traffic collection framework to collect a real-world traffic dataset that covers 48 popular apps. Experiments show that DF-Net not only achieves excellent performance but also more efficient compared to state-of-the-art methods.

Keywords: Traffic classification · Multimodal learning · Feature fusion

1 Introduction

Traffic classification plays an important role in network management, quality of service (QoS) and anomaly detection. To meet the requirements of network security and privacy protection, most applications use encrypted communication technology (e.g., TLS). According to Google's Transparency Report on HTTPS

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 384–392, 2022.
https://doi.org/10.1007/978-3-031-19214-2_32

traffic, as of February 2022, encrypted connections across Android products and services achieved 95%. Hence, mobile encrypted traffic classification is coming into focus along with the growing demand. However, traditional approaches that rely on deep packet inspection (DPI) or rule-based methods have become less viable, since all the communication contents are randomized after encryption [6]. Recently, end-to-end deep learning (DL) methods [1,3,6,9,10] perform well, but most models describe traffic from only one view, using unimodal input to train the model, which cannot consistently outperform in challenging scenarios.

In this paper, we propose a Deep Fusion Network (DF-Net) with multimodal Transformers [8] for mobile encrypted traffic classification which leverages more discriminative information from multiple modalities of raw traffic to improve the classification performance. DF-Net handles two modalities in a deep fusion manner, it extracts features from the initial TLS packet payload bytes on the one hand, and learns valid representation from the raw packet length sequences on the other. Deep fusion means that the model collectively concatenates the liner projection of payload bytes and the embedded representation of packet length. Specifically, we use a learnable modal-type embedding layer to fuse the heterogeneous data representation, and then the unified fusion vector is fed into the Transformer encoder. Model design follows the *single-stream* [2] approach and allows for a deep, early, and unconstrained fusion and interaction of cross-modal information at the input level. Experiments show that this design not only improves the performance, but also avoids the introduction of additional parameters that affect the model efficiency. Moreover, DF-Net adopts the simplest embedding scheme (liner projection) for payload bytes, and the multi-head self-attention mechanism endows the model with the ability of parallel computing and feature interactions from different representation subspaces [8]. Both the lightweight architecture (fewer parameters) and the parallel mechanism (faster computing speed) can improve the overall efficiency of the model. In order to obtain reliable data, we implement an automatic traffic collection framework for Android apps, which traverses all the widgets on the UI based on the depth-first search (DFS) algorithm to collect abundant and up-to-date traffic. Finally, we built a new mobile traffic dataset that contains 48 popular Android apps.

2 Related Works

AppScanner [7] utilizes statistical features of packet length to train Support Vector Machines (SVM) and Random Forest (RF) for recognizing apps. FlowPrint [4] extracts device, certificate and temporal features to represent each flow, and constructs a fingerprint library by clustering and cross-correlating for efficient traffic classification. FOAP [5] aims at open-world android app fingerprinting which constructs a bilevel recognition model and identifies user actions on specific UI components through. There are also some studies devoted to using deep models (e.g., 1DCNN [10] and Autoencoder [3]) to detect malicious traffic. Fs-Net [6] combines stacked bidirectional GRUs with a reconstruction mechanism to learn features from packet length sequences. Current multimodal models [1,9] rely on CNN and RNN models to extract features separately, and then combine the outputs of the two branches for classification, but they pay little attention to the modality fusion strategy and the issue of model efficiency.

3 The Proposed Multimodal Model

3.1 Modal Choices

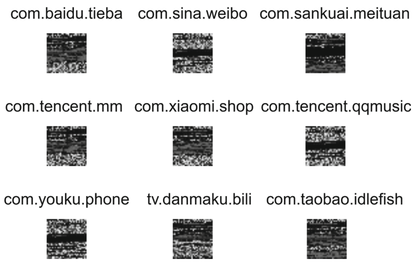


Fig. 1. Visualization of the payload bytes

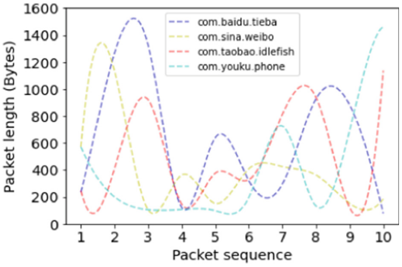


Fig. 2. Packet length trends

In this paper, we describe traffic data from multiple views, allowing the model to leverage more distinguishable information. As the payload data (TCP/IP model layer 4) during the initial TLS handshake (i.e., ClientHello and Server-Hello messages) usually contains plaintext fields. From our observations, there are significant differences in the cipher suites which can be selected to use according to platform support, preference or random selection. Also, some extension fields such as Server Name Indication (SNI) provide distinguishable features. As shown in Fig. 1, the raw packets are converted into bytes and visualized. The L4 layer starts with 900 consecutive bytes of different apps has significant differences. However, the plaintext information in the handshake phase is greatly reduced in TLS 1.3, so it is more necessary to use the packet length feature as a supplement. Figure 2 plots the interpolation smooth curve of the packet length of different apps. We can see that the trends and fluctuations of the packet length of different apps are also significantly different. So we take packet length sequences and payload bytes as the input data. There are also some metadata such as Inter Arrival Time (IAT) and message type that can be used, but IAT is easily affected by the network environment, and the message type sequence is not ample enough to provide recognition space. Besides, more input is not conducive to model lightweight.

3.2 Model Architecture

The DF-Net is a hierarchical model as shown in Fig. 3, which consists embedding layer, Transformer encoder and classification layer.

Embedding Layer. DF-Net follows the *single-stream* [2] approach that the embedding layer collectively operates on a concatenation of multimodal input. The full vector representation is constructed by summing up three embeddings: token embedding, position embedding and modal-type embedding.

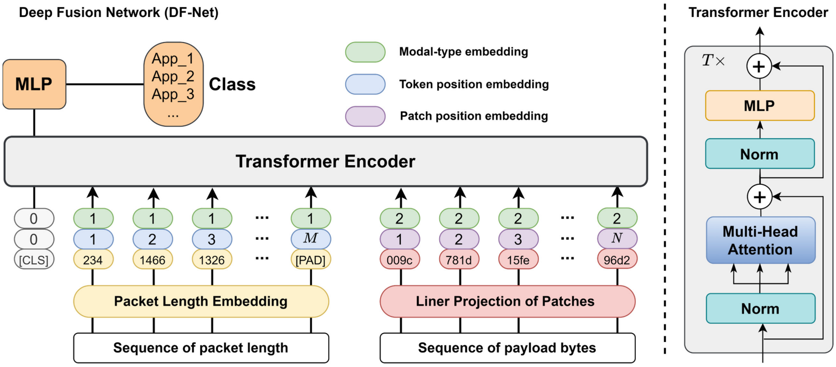


Fig. 3. The overview of DF-Net architecture

Token Embedding. For packet length sequences, every packet length token is like a vocab in the dictionary. There is a learnable embedding matrix $V_L \in \mathbb{R}^{K\times d}$, where K is the size of a dictionary. Here, we set $K = 1500$ because the maximum transmission unit (MTU) of the Ethernet is usually 1500 bytes, and the embedding dimension d is set to 128. Given a packet length sequence with M elements $L = [l_1, ..., l_M]$, each element l_m is fed into embedding matrix $V_L \in \mathbb{R}^{K\times d}$ and converted into a embedding vector $e_{l_m} \in \mathbb{R}^d$. Here, the input sequence length is required to be at least $M = 10$ to support the best result. Finally, we can obtain the embedding sequence:

$$E_L = [e_{l_1}, ..., e_{l_M}], E_L \in \mathbb{R}^{M\times d} \tag{1}$$

As for payload bytes, we use patch projection embedding for image classification because byte sequences are like flattened image pixels. Given input bytes $B = \{\mathbf{x_1}, ..., \mathbf{x_N}\}$, where $\mathbf{x_n} \in \mathbb{R}^P$. Here we truncate consecutive 900 bytes for each flow, so we derive $N = 6$ patches, and each patch contains $P = 150$ bytes, that enough to offer the best result. Followed by linear projection matrix $V_B \in \mathbb{R}^{P\times d}$, we obtain the embedding sequence:

$$E_B = [e_{\mathbf{x_1}}, ...e_{\mathbf{x_N}}], E_B \in \mathbb{R}^{N\times d} \tag{2}$$

Position Embedding. Due to the transmission of traffic data being closely related to order, positional embedding to ensure that the model pays attention to the temporal relationship of tokens through relative positions. We denote the positional embedding of two modalities as: $E_L^{pos} \in \mathbb{R}^{(M)\times d}$ and $E_B^{pos} \in \mathbb{R}^{(N)\times d}$, where the embedding dimension d is same as token embedding. For each modality, position information is computed from scratch.

Modal-type Embedding. We assign different tags to different modals, that fuse the multimodal data to form a unified input to the Transformer encoder in the early stage. This allows for a deep and unconstrained interaction of cross-modal information at the early stage to improve the model performance. The modal-type embedding vectors are represented as: $E_L^{type} \in \mathbb{R}^{M \times d}$ and $E_B^{type} \in \mathbb{R}^{N \times d}$. Then the token and position embeddings are summed with their corresponding modal-type embedding vectors. Finally, the embedding sequence $z \in \mathbb{R}^{(M+N+1) \times d}$ is uniformly input to the next layer. We add a special classification token e_{class} ([CLS]) as the first token of every combined sequence. The final hidden state corresponding to the [CLS] token is used as the aggregate sequence representation (global feature aggregation) for classification task.

$$z = [e_{class}; E_L + E_L^{pos} + E_L^{type}; E_B + E_B^{pos} + E_B^{type}] \tag{3}$$

Transformer Encoder. Transformer encoder consists of $T = 4$ stacked identical layers. Each layer includes a multi-head self-attention (MSA) block and a MLP block. Layer normalization (LN) is applied before every block and residual connections after every block. For standard **qkv** self-attention (SA), each element in an input sequence $z \in \mathbb{R}^{(M+N+1) \times d}$, it computes a weighted sum over all values $\mathbf{v}$ in the sequence. The attention weights A_{ij} are based on the pairwise similarity between two elements of the sequence and their respective query $\mathbf{q}^i$ and key $\mathbf{k}^i$ representations with dimension d_k. Multihead self-attention (MSA) is an extension of SA in which we run h self-attention operations in parallel, called "heads", and project their concatenated outputs. To keep compute and the number of parameters constant when changing h. Besides, each of the layers contains a fully connected feed-forward network, which is applied to each position separately and identically. This consists of two linear transformations with a ReLU activation in between. In this model, the number of heads set to $h = 2$.

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = SU_{qkv} \tag{4}$$

$$SA(z) = Softmax(\mathbf{q}\mathbf{k}^\top / \sqrt{d_k})\mathbf{v} \tag{5}$$

$$MSA(z) = [SA_1(z); SA_2(z); ...; SA_h(z)]U_{msa} \tag{6}$$

$$MLP(\widetilde{z}) = ReLU(\widetilde{z}W1 + b1)W2 + b2 \tag{7}$$

3.3 Classification Layer

A two-layer MLP classifier with GeLU activation (followed by dropout) to predict the output classes. The first position hidden state z_0^T which corresponds to the special [CLS] token will be employed directly for classification, because it can learn a robust representation of the entire sequence due to self-attention mechanism. Softmax classifier outputs the distribution $\mathbf{p}$ over the predictions with ground-truth targets A. In the training phase, we apply the cross-entropy classification loss to train the model as follows:

$$\mathbf{p} = Softmax(GeLU(z_0^T W_1 + b1)W_2 + b_2) \tag{8}$$

$$Loss = CrossEntropyLoss(A, \mathbf{p}) \tag{9}$$

4 Evaluation

4.1 Traffic Collection

DL models are data-driven, so high-quality datasets are crucial. The datasets used in the past have the potential to be outdated, resulting in Concept Drift. To obtain reliable ground truth, we implemented an automatic traffic collection framework for Android apps based on the Appium automation test tool.

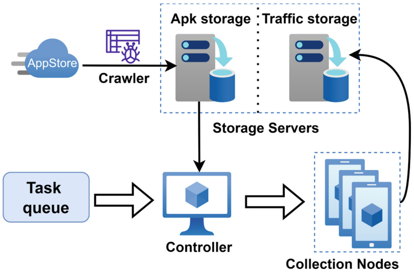


Fig. 4. Automatic collection framework

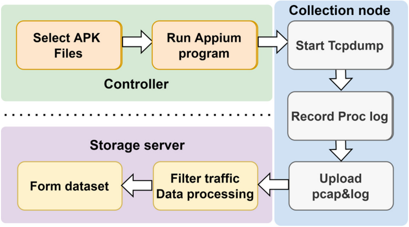


Fig. 5. The workflow of traffic collection

The collection framework and workflow are shown in Fig. 4 and Fig. 5 respectively. The controller issues the tasks, and the distributed nodes collect traffic according to the algorithm and store it to the storage servers. For an application, the Appium program traverses all the widgets on the UI based on the depth-first search algorithm (DFS) by using XPath to parse the XML source code of the user interface (UI), which simulates user operations, such as clicking, sliding, etc. Then the Tcpdump captures traffic during the traversal process. Duo to Proc file system (/proc/net) will display the socket usage of all UIDs in real-time, we get the unique identifier (UID) of the app by accessing the ADB to correlate a Network Flow to an App. The raw traffic will be cleaned and filtered, then we further extract packet length and the payload information of bidirectional flows, and these data constitute our dataset in this paper. Compared with other collection methods (e.g., NetLog [9]), our framework has the following advantages: 1)The framework is designed as a distributed architecture. It achieves automatic collection and labeling, which significantly improves the collection efficiency and provides support for the construction of large-scale datasets. 2)The traversal algorithm can trigger various functions of the application, thereby generating abundant traffic and providing data support for training excellent models. The original traffic were captured on Android devices and emulators (i.e., Xiaomi Mi 8 and MuMu emulator) during May.2021 - Jul.2021. Raw traffic with 35G, and the dataset containing 48 applications and 176,200 encrypted network flows. For brevity, we rank all the apps and present the top 16 apps with the most TLS flows in Table 1.

Table 1. Top 16 Apps With The Most TLS flows

ID	App	Flows	ID	App	Flows
1	com.sankuai.meituan	13890	9	com.xiaomi.shop	5508
2	com.sina.weibo	12706	10	com.dangdang.buy2	4979
3	com.tencent.qqmusic	11780	11	com.qq.ac.android	4826
4	com.youku.phone	8858	12	com.wuba	4485
5	tv.danmaku.bili	8082	13	ctrip.android.view	4064
6	com.baidu.tieba	7963	14	com.zhihu.android	3899
7	com.taobao.idlefish	7820	15	com.sohu.newsclient	3678
8	com.tencent.mm	5526	16	com.sina.news	3606

4.2 Experiments and Analysis

Performance Comparison. *Accuracy*, *Precision* and *Recall* are used as metrics. Considering the presence of class imbalance in our multiclass settings, we use the *Macro F-measure* to evaluate the overall performance. To facilitate presentation, the general overview of performance is shown in Table 2.

Statistical Methods. Appscanner [7] derived 54 statistical features from the packet sizes of each flow. We use the optimal parameters set by the paper and the same data preprocessing method. As seen in the Table 2, DF-Net significantly outperforms ML-based methods, with an improvement up to +15.23% on Macro-F1 compared with the RF method. Furthermore, we can see that the multi-class SVM method performs poorly with only 55.76% F1-score, which indicates that SVM is not suitable for multi-class scenarios. It also illustrates that ML-based methods require careful selection of appropriate models and features to improve performance, which is complicated and time-consuming.

Table 2. Experiment results of comparison methods

Model type	Model name	Accuracy	Precision	Recall	Macro-F1
Statistical	AppScanner-SVM	56.63 ± 0.6	52.37 ± 0.9	59.63 ± 0.9	55.76 ± 1.2
	AppScanner-RF	78.13 ± 0.7	81.92 ± 0.6	76.58 ± 0.6	79.16 ± 0.7
Unimodal	1D-CNN	86.36 ± 0.9	87.62 ± 0.6	84.79 ± 0.7	86.18 ± 0.9
	Fs-Net	79.61 ± 0.5	81.07 ± 0.5	78.23 ± 0.6	79.62 ± 0.7
Multimodal	MIMETIC	87.32 ± 0.4	88.11 ± 0.6	86.91 ± 0.5	87.51 ± 0.8
	App-Net	91.24 ± 0.4	94.26 ± 0.5	89.87 ± 0.6	92.01 ± 0.7
Variant	Single-Byte	86.97 ± 0.5	85.68 ± 0.4	87.98 ± 0.6	86.81 ± 0.8
	Single-Length	82.13 ± 0.5	86.36 ± 0.6	79.67 ± 0.7	82.88 ± 0.9
	Dual-Stream	93.37 ± 0.2	94.69 ± 0.4	92.02 ± 0.2	93.33 ± 0.1
Deep fusion network		**95.26 ± 0.2**	**95.84 ± 0.3**	**92.98 ± 0.5**	**94.39 ± 0.4**

Deep models. Compared with 1D-CNN [10] and Fs-Net [6], DF-Net has a significant performance improvement of +8.21% and +14.77% respectively on Macro-F1. Moreover, compared with two unimodal variant methods based on payload bytes (Single-Byte) and packet length (Single-Length), DF-Net improves the Macro-F1 by +7.58% and +11.51%, respectively. This proves that multimodal features indeed provide more distinguishable information. Besides, we observe that the model based on payload bytes outperforms the model based on packet length, this shows that these plaintext observable data fields, exposed in TLS connections, have more valuable information that can be used to build a more excellent classifier. DF-Net also has an overall lead in performance, with a increase of 6.88% on F1-score compared to MIMETIC [1]. App-Net [9] has a similar performance to our model with a difference of 2.38% on F1-score, but there is a problem with efficiency.

In addition, we also conducted a fine-grained performance experiment that checks their Top-K accuracy (K $\in \{1, 3, 5\}$) in Fig. 6. Obviously, we can see that the performance of all classifiers can be improved when relaxing the results. Our modal also achieves the best performance, when the Top-3 and Top-5 predicted apps are considered. However, Unimodal classifiers (1D-CNN and Fs-Net) are not close to the accuracy of the DF-Net or Appscanner-RF because they cannot infer deep traffic patterns from a single information dimension. We also compare fusion variant methods *Dual-Stream*. From the results, the deep fusion technique achieves the best performance. Although the performance gap is not large, the joint loss needs to manually set predefined weights for each unimodal feature [9], which may lead to modal bias. However, our model utilizes MSA mechanism to learn the feature attention weights for different modalities by itself.

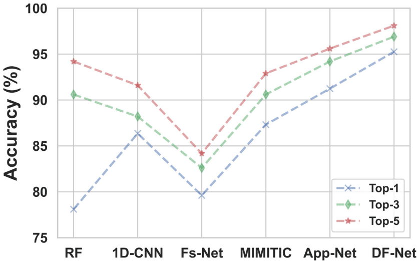


Fig. 6. Top-K Performance comparison

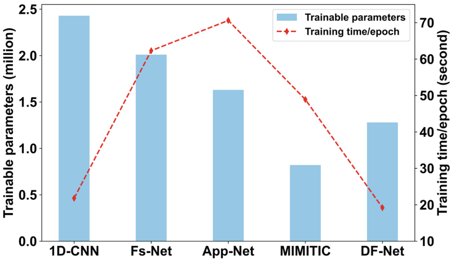


Fig. 7. Model complexity analysis

Analysis on Model. We also analyze the complexity of models including trainable parameters and run-time per epoch for training. The results are shown in Fig. 7. These experimental results demonstrate that our model has the shortest training per epoch. Compared with *Dual-Stream* App-Net [9] with similar performance, the training time is shortened by 3.6 times, the inference time is shortened by 5.1 times, and the parameters are reduced by 21%. MIMETIC [1] has the fewest parameters, but the two-stage training method increases the overall training time (+154%) because the serial input mode slows down the computing speed (inherent flaws of recurrent neural networks). 1D-CNN has more parameters but a faster training speed, due to the shared parameter mechanism of CNN and its stronger parallel ability. In summary, DF-Net adopts the simplest embedding scheme, and Transformer provides better parallelism to overcome the inherent defects of RNN that needs to process the input at each time step serially. Lightweight designs indeed promote model efficiency and meet the requirements of real-time online detection in the real world.

Acknowledgment. This work was supported by National Key Research and Development Program of China (Grant No.2020YFB1806402), Special project for innovative methods of the Ministry of science and technology of China (Grant No.2019IM020100).

References

1. Aceto, G., Ciuonzo, D.: Montieri: mimetic: mobile encrypted traffic classification using multimodal deep learning. Comput. Netw. **165**, 106944 (2019)
2. Bugliarello, E., Cotterell, R., Okazaki, N., Elliott, D.: Multimodal pretraining unmasked: a meta-analysis and a unified framework of vision-and-language berts. Trans. Assoc. Comput. Linguist. **9**, 978–994 (2021)
3. Ding, S., Zhang, D., Ge, J., Yuan, X., Du, X.: Encrypt DNS traffic: automated feature learning method for detecting DNS tunnels. In: 2021 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), pp. 352–359. IEEE (2021)
4. Ede, T.V., Bortolameotti, R., Continella, A., Ren, J., Peter, A.: Flowprint: semi-supervised mobile-app fingerprinting on encrypted network traffic. In: Network and Distributed System Security Symposium, vol. 27 (2020)
5. Li, J., Zhou, H., Wu, S., Luo, X.: FOAP: fine-grained open-world android app fingerprinting. In: 31st USENIX Security Symposium (USENIX Security 22). USENIX Association (2022)
6. Liu, C., He, L., Xiong, G., Cao, Z., Li, Z.: Fs-net: a flow sequence network for encrypted traffic classification. In: IEEE INFOCOM 2019-IEEE Conference On Computer Communications, pp. 1171–1179. IEEE (2019)
7. Taylor, V.F., Spolaor, R., Conti: appscanner: automatic fingerprinting of smartphone apps from encrypted network traffic. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 439–454. IEEE (2016)
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez: attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017)
9. Wang, X., Chen, S., Su, J.: Automatic mobile app identification from encrypted traffic with hybrid neural networks. IEEE Access **8**, 182065–182077 (2020)
10. Wei, W., Ming, Z., Wang, J., Zeng, X., Yang, Z.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 43–48 (2017)

Pick-Up Point Recommendation Using Users' Historical Ride-Hailing Orders

Lingyu Zhang[1,2], Zhijie He[2], Xiao Wang[2], Ying Zhang[2], Jian Liang[2], Guobin Wu[2],
Ziqiang Yu[3], Penghui Zhang[4], Minghao Ji[4], Pengfei Xu[4], and Yunhai Wang[1](✉)

[1] School of Computer Science and Technology, Shandong University, Qingdao, China
cloudseawang@gmail.com
[2] Didi Chuxing, Beijing, China
{hezhijie_i,yingzhangying,liangjian,wuguobin}@didiglobal.com,
wangxiao@didichuxing.com
[3] Yantai University, Yantai, Shandong, China
zqyu@ytu.edu.cn
[4] School of Information Science and Technology, Northwest University, Kirkland, USA
pfxu@nwu.edu.cn

Abstract. The ride-hailing app must provide users with appropriate pick-up points when they submit their travel demands and their locations are recognized, efficiently reducing users' operation complexity and optimizing the software performance. Most apps currently try to search for locations near users' current GPS locations as the Points of Interest (POIs), which is an efficient method of locating, but seriously ignores personal preferences. In this paper, we deeply analyze the historical ride-hailing orders of users on Didi Chuxing platform (http://www.didiglobal.com). We explore the given dataset, get the general regularity of users' commuting, and propose a Pick-Up Points Recommendation Model (PPRM) based on the clustering algorithm. We cluster users' historical orders using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) according to orders' spatial information. In this way, the candidate outputs closest to the user's current environment/feature can be found in a specific category. The linear addition of the candidate outputs severs as the final pick-up point provided. Therefore, our model can offer recommendations of the best pick-up points. In addition, experimental results based on real-world datasets indicate that our model can efficiently and accurately provide users with optimal points.

Keywords: Pick-up point recommendation · Travel pattern mining · Cluster analysis · Ride-hailing system · Data analysis

1 Introduction

Pick-up point recommendation is one of the essential functions of ride-hailing apps. With the increasing of users' historical ride-hailing orders, it is vital to dig out users' travel patterns from users' historical travel records and recommend the optimal pick-up points that meet their preferences.

Location prediction has been a craze for a long time. Numerous location data (such as trajectories, social network sign-in data, and location information obtained by

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 393–405, 2022.
https://doi.org/10.1007/978-3-031-19214-2_33

various smart terminals) covers the users' travel characteristics and can be used to predict the following locations of users [14,20]. For instance, Li et al. [9] and Yoon et al. [23] predicted individual locations by calculating the similarity of moving behaviors and trajectories, respectively. Tseng et al. [16] tried to mine the mobile sequential patterns related to users' movement paths and time intervals and predicted users' following locations. The space areas can be divided by the Voronoi diagram, and a Markov location prediction model [12,15] based on regional features of users' movements is proposed. Furthermore, based on the trajectory data generated by smartphones and wearable smart devices, Kown et al. [4] proposed a location prediction method via pattern matching and similarity measurement. In addition, Zhang et al. [25] deeply mined the users' moving patterns and used the generated moving rules to predict the following locations.

The migration of users' locations has temporal and spatial correlation, and exploring the temporal and spatial patterns is essential for accurate location prediction. Lei et al. [6] proposed a spatiotemporal trajectory model, which could capture the spatiotemporal features of individual trajectories and improve the accuracy of location prediction. Xu et al. [18] converted the location prediction problem into a classification problem by extracting the spatiotemporal features in the historical trajectory data and proposed a learner based on a modified Support Vector Machine. At the same time, temporal and spatial gates that independently process individual movement information are introduced into Long Short-Term Memory (LSTM) to effectively extract individual trajectory features [19]. Moreover, Zhang et al. [26] proposed a multi-task location prediction framework based on LSTM and Convolutional Neural Networks (CNN). In this model, LSTM is responsible for extracting the location sequence and time attributes, and CNN extracts the spatial correlation of each location. In addition, LSTM can not only predict the short-term location of users but also the long-term movement trajectory by mining the periodicity of the users' movement [17]. Location prediction can also be considered as a classification problem based on the users' current feature [21,29]. For example, Lei et al. [5] proposed a spatiotemporal trajectory framework, which extracted the spatial information by the clustering algorithm and then explored individual travel behavior in the form of a probability suffix tree. Li et al. [10] classified users according to certain classification standards and proposed corresponding prediction schemes for each category of users.

The methods mentioned above only consider the basic temporal and spatial information, while ignore the deeper features such as users' preferences [22], context [2], social correlation [3], and location semantics [27]. By analyzing the location information recorded by smartphones on social networks, Zhou et al. [28] concluded that collective spontaneous mobility would affect users' mobility. And this conclusion is proved to be effective for location prediction. Zhang et al. [24] represented users' preferences with a tensor and used the preference tensor to predict the next location. Moreover, a multi-context-based deep neural network location prediction model was proposed by Liao et al. [11], which captures the deep-level preferences by modeling different contexts. In addition, the sparsity of historical trajectory data often negatively affects the accuracy of position prediction. Therefore, individual-group trajectory prediction models were proposed in [7,8], in which the methods of location extraction, clustering, matching prediction, and probability suffix tree are used to reduce the impact of data sparsity on the prediction results and improve the accuracy.

In this paper, we propose a Pick-Up Point Recommendation Model (PPRM) using users' historical ride-hailing data and can determine whether to provide relevant services according to users' requirements on the recommendations. Our contribution can be summarized in the following points:

1. We deeply dig into the personal travel pattern of users and outline some commonalities through the detailed analysis of the historical order data in many aspects. We further design an unique location recommendation model to provide users with optimal pick-up points which suit their preferences.
2. We introduce DBSCAN, which can be effectively classified without specifying the number of, to cluster user's historical orders, and the idea of order clustering before order matching can effectively reduce the running time and computation complexity of the location recommendation model.
3. The experimental results show that the models designed in this paper can predict users' pick-up points nicely and further optimize the related functions of existing apps to facilitate the operations of users.

2 Dataset and Dataset Analysis

2.1 Dataset

The dataset used in this paper is the historical ride-hailing orders of Didi Chuxing's users from January to June, 2019. The items of each order are as follows:

- **User_ID** and **Order_ID**: Each user or order has a distinctive id.
- **Departure_time**: The user's pick-up time in each order.
- **Starting_lat** and **Starting_lng**: The latitude and longitude of the user's pick-up/starting point, respectively.
- **Starting_name**: The point of interest (POI) of the user's pick-up points.
- **Dest_lat** and **Dest_lng**: The latitude and longitude of the user's destination, respectively.
- **Dest_name**: The POI of the user's destination.

2.2 Dataset Analysis

The distribution of users' orders is shown in Fig. 1. We can notice that the historical order volume of most users within half a year is between 280 to 520 (Noted in the orange box), and only a few have more than 750 orders (Noted in the red box). All order information has been anonymously summarized, and abnormal orders caused by the cancellation of passengers or any other reason have been excluded.

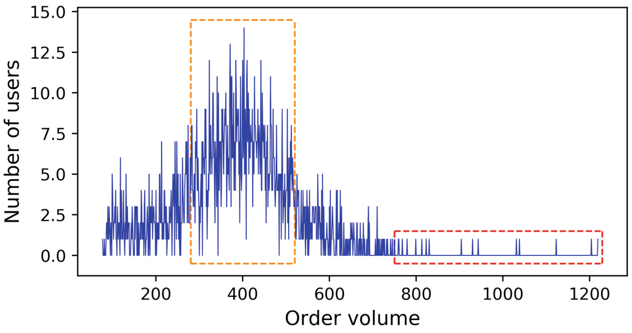
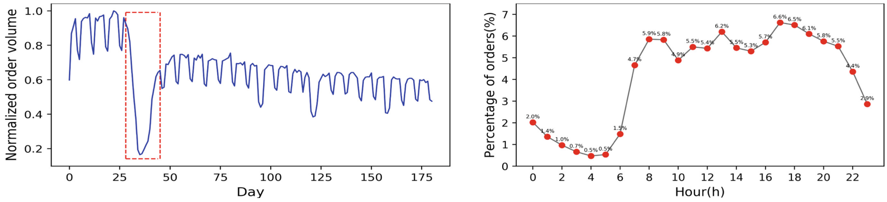


Fig. 1. The numbers of users under different order volumes. (Color figure online)

We separately counted the daily and hourly order volume. The daily order volume has a stage of rapid decay and rise, as shown in the red box of Fig. 2(a). This stage starts at the end of January and ends at the beginning of February. Although the daily order volume later is at a high level, the overall trend is declining. While, the hourly order volume has three peaks, which are corresponding to rush hour at morning/afternoon/evening, respectively, as shown in Fig. 2(b). Moreover, users concentrate on taking taxis during the day, and the daytime (7:00 am–7:00 pm) order volume account for about 68% of the total.



(a) Daily order volume.

(b) Hourly order volume.

Fig. 2. The volume of orders over time. (Color figure online)

Besides, there are differences between orders on workdays and holidays. Figure 3 respectively depicts the proportion of order volume in each period of workdays and holidays. There are three peaks of order volume on workdays, which is similar to Fig. 2(b). However, there is no morning peaks on holiday. Traffic jams in the morning on workdays often occur due to daily commuting. On the contrary, people often choose to go out relatively late on holidays.

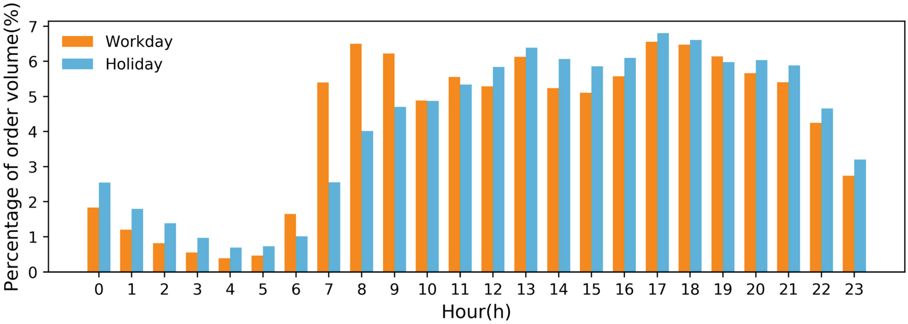
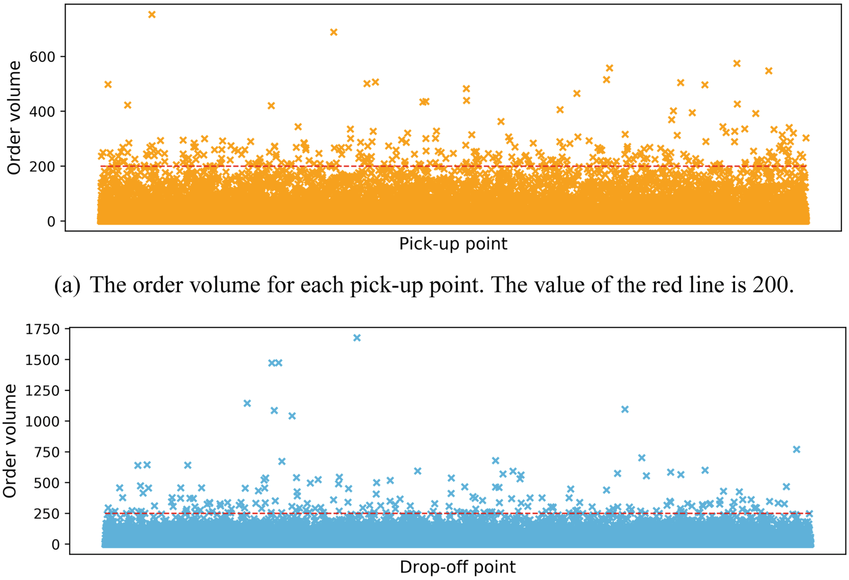


Fig. 3. The order volume on workdays and holidays.

In addition, the dataset of users' orders contains 112581 POIs of pick-up points and 88717 POIs of drop-off points. The order volume of each location is listed in Fig. 4. Since the number of drop-off points only accounts for 78.8% of the number of pick-up points, the orders in the dataset have a certain degree of aggregation in the spatial dimension, which can be seen in Fig. 4 marked in the red lines. It is worth noting that several drop-off points in Fig. 4(b) share extremely high order volume. We check these locations separately and find that they are all located at commercial spots or train stations. These areas have always been the places with high demand for urban commuting.



(a) The order volume for each pick-up point. The value of the red line is 200.

(b) The order volume for each drop-off point. The value of the red line is 200.

Fig. 4. The order volume for each pick-up and drop-off point. (Color figure online)

3 User Travel Behavior Analysis

3.1 The Analysis of Temporal Information in Users' Historical Orders

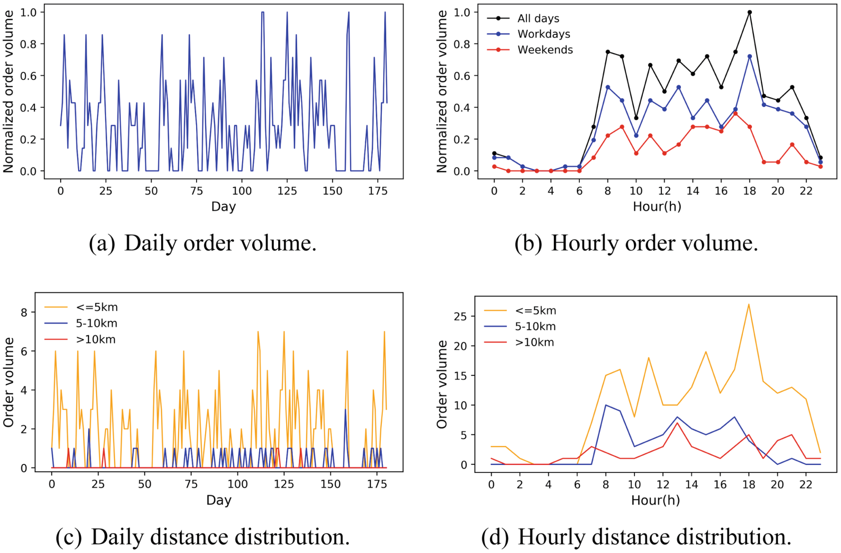
Figure 5(a–b) shows the temporal distribution of users' historical orders. It can be seen that the users took a taxi at least once in 68% of days since the number of workdays is the majority, the trend of orders on workdays is alike as to the overall trend. There are three peaks in the order curve of the workday, which is similar to that in Fig. 2(b). However, the curve of order volume for the weekends is quite different. The trend of orders on weekends is more gradual compared to workdays, and there is only one peak at 17:00.

3.2 The Analysis of Spatial Information in Users' Historical Orders

According to the straight-line distance between the pick-up and drop-off points in each order, we set three levels to divide the distance: short-distance (<5 km), mid-distance (5–10 km), and long-distance (>10 km). Figure 5(c–d) describes the distance distribution. Short-distance and mid-distance trips account for the vast majority of the travel records. In Fig. 5(d), the hourly short-distance distribution curve has a similar trend to the hourly order volume curve in Fig. 5(b). The number of mid-distance trips is smaller than that of short-distance trips but is larger than that of long-distance trips.

The pick-up and drop-off points of most users are highly clustered. We show the relative positions of pick-up points and drop-off points in all historical orders in Fig. 6(a), and we can see that most locations are clustered in a certain space range, as shown in Fig. 6(b).

Through the same analysis of other users, we discover two commonalities of users' travel behavior:



(a) Daily order volume.

(b) Hourly order volume.

(c) Daily distance distribution.

(d) Hourly distance distribution.

Fig. 5. The temporal and spatial distribution of historical orders.

1. The travel time patterns of users can be roughly divided into two categories: regular and irregular. Most users have obvious different travel patterns in different periods, such as workdays and holidays. These users have regular travel patterns of relatively stable travel time and relatively similar pick-up and drop-off points. But there are also users with no regular patterns. In addition, most users' pick-up and drop-off points are clustered, and also, a few users own discrete locations.
2. The average distance of each user's historical orders is distinct, but most users take short-distance and mid-distance trips as their primary travel modes.

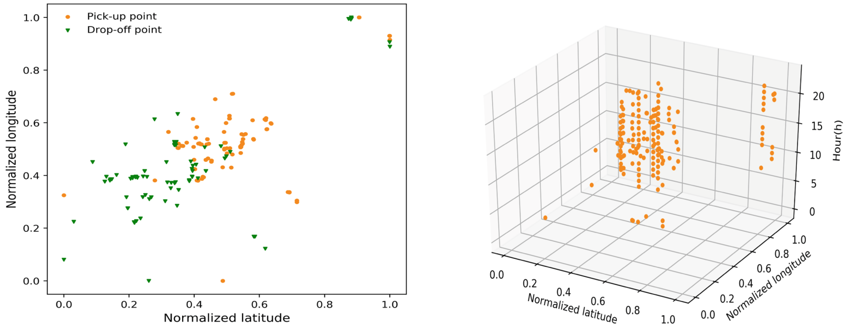
4 Pick-Up Point Recommendation Model

Pick-up Point Recommendation Model (PPRM) can be briefly stated as follows: First, the users' historical orders are clustered by DBSCAN [1] based on the spatial distribution of pick-up points, and the orders that match the users' current environment are searched in a certain category. The overall framework of PPRM is shown in Algorithm 1. Note that all feature vectors in our algorithm have been normalized.

4.1 Order Clustering via DBSCAN

DBSCAN is a density-based spatial clustering method, which treats an area with sufficient density as a cluster (category) and can find arbitrary-shaped clusters in spatial data with noise. Here, a cluster is defined as the largest collection of closely connected points. Compared with other clustering algorithms, like the k-means clustering algorithm [13], DBSCAN has the advantages as follows:

- DBSCAN does not need to specify the number of clusters manually.
- DBSCAN can find clusters of any shape.
- DBSCAN can identify the noise points.



(a) The spatial distribution of pick-up points and drop-off points.

(b) The temporal and spatial joint distribution of pick-up points.

Fig. 6. The spatial distribution of historical orders.

Algorithm 1: PPRM

```
Input: feature vector $v$, historical order data $X$
Output: recommended pick-up point $x_{rec}$
// Order clustering
The orders are clustered by Algorithm 2;
// Order matching
The set of best matching orders $V_{best}$ is obtained by Algorithm 3;
// Pick-up point recommendation
if Match failed then
    The pick-up point is not recommended;
else
    $x_{rec} = \frac{1}{n_{best}} \sum_{i=1}^{|n_{best}|} x_{i,best}$,where $n_{best}$ denotes the number of samples in $V_{best}$
    and $x_{i,best}$ is the sample in $V_{best}$;
    $x_{rec}$ = Inv-normalized($x_{rec}$), where Inv-normalized($\cdot$) denotes the inverse operation
    of normalization;
end
```

Figure 6(a) has shown the spatial distribution of pick-up points, and outliers can be regarded as "noise points." First, we randomly select a point from the set of pick-up points and search all the points within the specified radius centered on the chosen point. Then if the number of searched points exceeds the threshold we set, all the searched points are grouped into one cluster, and the point selected is called the core point. Otherwise, use the next point to continue the above operation. Algorithm 2 shows the complete concept of order clustering based on the pick-up points set.

4.2 Order Matching

According to the clustering results in Sect. 4.1, the current best matching order can be found from a certain category. Compared to traversing the entire historical order data, our method has lower time complexity.

First, we calculate the category center of each category C_k, $k = 1, 2, \ldots, m$, which can be defined as

$$c_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{ik}, \tag{1}$$

where x_{ik} denotes the i-th sample in C_k and n_k is the number of samples in C_k. Then, the feature vector v (normalized latitude and longitude) is extracted according to the user's current environment. Finally, the model tries to search for the best historical

Algorithm 2: Order clustering via DBSCAN

```
Input: search radius $r$, minimum number of samples within the search radius $t$, sample
       point collection $X = \{x_1, x_2, ..., x_n\}$, $n$ is the number of sample points
Output: all clusters
while unclassified samples are existed do
    // Select the initial point
    Randomly select an unclassified point as the initial point $x$ and define a new cluster
    $C_k = \{x\}$;
    //Sample search
    while untraversed points are existed in $C_k$ do
        Select an untraversed point $x_i$;
        if $x_i$ is core point then
            All points within the search radius $r$ of $x_i$ are classified to cluster $C_k$;
        else
            continue;
        end
    end
end
Output all clusters $C_k, k = 1, 2, \ldots, m$;
```

order matching from the closest category. In this paper, the distance d_k between the feature vector v and the category center C_k is defined as Euclidean distance between v and c_k. Similarly, the distance between two feature vectors is also defined as Euclidean distance. Algorithm 3 shows the process of order matching.

$$d_k = \|v - c_k\|. \tag{2}$$

5 Experimental Results and Analysis

In this section, we conduct multiple experiments on our dataset, show the related experimental results, and give the corresponding analysis.

5.1 The Analysis of Order Clustering

The clustering algorithm is carried out only on spatial locations, so we made a data preprocessing as follows: First, we extract the latitude and longitude of the pick-up points from each historical order. And then, we set the number of the training set and test set account for 80% and 20%, respectively. In addition, all samples have been normalized.

Since the data in the training set is unlabeled, we introduce Silhouette Coefficient (SC) to measure the effect of data clustering, and the Silhouette Coefficient of sample x_i is defined as

$$SC(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}, \tag{3}$$

Algorithm 3: Order matching

```
Input: feature vector $v$, category $C_k, k = 1, 2, \ldots, m$
Output: the collection of best matching orders $V_{best}$
Initialize:$V_{best} = \varnothing$
//Compute the category center
for $k = 1, 2, \ldots, m$ do
    compute $c_k$ by Eq. 1;
end
// Match the category of $v$
for $k = 1, 2, \ldots, m$ do
    compute $d_k$ by Eq. 2;
end
The category with the short distance is regarded as the category of $v$, denoted as
  $C_v = \{x_{1v}, x_{2v}, \ldots, v_{nv}\}$;
// Order matching
for $i = 1, 2, \ldots, n$ do
    compute the sample distance $d_{iv}$ between $v$ and $x_{iv}$ by $d_{iv} = \|x_{iv} - v\|$;
end
Select at most 3 samples with the geographic distance less than 100m from $C_v$ as the best
  matching samples and store them in $V_{best}$;
if $|V_{best}| > 0$ then
    Output $V_{best}$;
else
    Match failed;
end
```

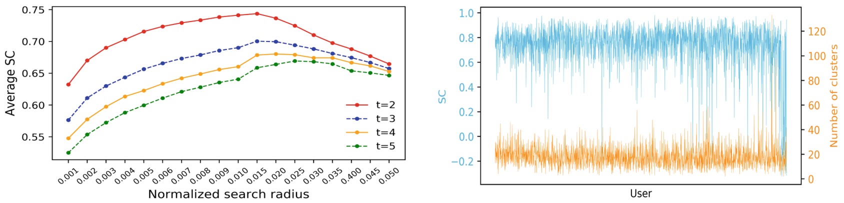
where $a(i)$ denotes the average distance between x_i and other samples in the cluster of x_i, and $b(i)$ denotes the average distance between x_i and all samples in the cluster closest to x_i. $a(i)$ and $b(i)$ can also be called the degree of dissimilarity, and the value range of $SC(i)$ is in $[-1, 1]$. Moreover, the larger the value of the SC is, then the better the clustering effect will be.

Figure 7(a) describes the performance of DBSCAN under various parameter combinations (r and t in Algorithm 2). When $t = 2$ and $r = 0.015$ (the actual geographic distance is about 1 km), the average SC of all users is the smallest. The clustering effect of each user's orders under the best parameter is shown in Fig. 7(b), and the results of most users are satisfactory.

5.2 The Analysis of the Results Obtained by PPRM

In order to ensure the validity of the experiment, we first randomly drift 0–50 m for each sample in the test dataset. Meanwhile, two metrics are introduced to measure the performance of PPRM, i.e., prediction rate (PDR) and distance error (DisErr). PDR is defined as

$$PDR = \frac{n_{rec}}{n_{text}} \times 100\%, \tag{4}$$



(a) Average SC under each parameter, where t is the minimum number of samples within the search radius.

(b) SC and the number of clusters of each user under the best parameters.

Fig. 7. Clustering results via DBSCAN.

where n_{rec} denotes the number of test samples with the pick-up points, and n_{test} is the number of test samples. PPRM outputs the predicted locations in the form of latitude and longitude, so DisErr is defined as the geographic distance by calculating the latitude and longitude of the predicted locations and the actual locations.

According to the parameter combinations of r and t, the results of PPRM are shown in Fig. 8. It shows that the average PDR can reach 82.38%. In other words, PPRM is able to handle 82.38% of the demand scenarios. Moreover, the average DisErr is only 23.14 m, so the recommended and the actual point-up point can be considered as the same POI.

In addition, for the situation where the remaining PPRM doesn't work, our solution is to search for the POIs, which are closest to the user from the database as the recommended pick-up points. The recommended locations in this way may not be in line with the user's preferences, but the unnecessary troubles caused by the user entering the wrong location information can be avoided in many cases.

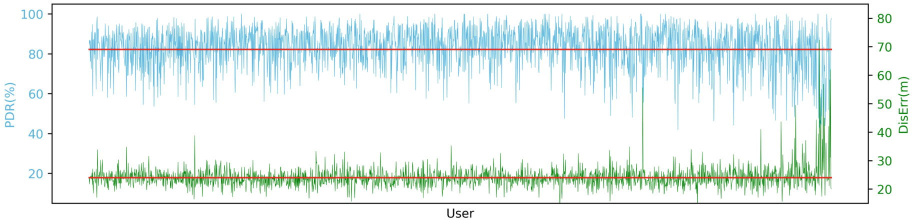


Fig. 8. The results obtained by PPRM, where the two red lines denote the mean values of PDR (82.38%) and DisErr (23.14 m). (Color figure online)

6 Conclusion

In this paper, we deeply mine the users' travel patterns from both temporal and spatial information of users' historical ride-hailing orders and summarize the general regularity characteristics of users' travel behavior. According to the spatial distribution of users' historical pick-up points, we propose a pick-up point recommendation model (PPRM) based on DBSCAN. PPRM consists of two components: order clustering and location recommendation. First, the historical orders of each user are clustered according to the density of pick-up points. Then, the feature vector is extracted from the user's current environment. Finally, the most similar orders are searched and used as the matched orders, and the latitude and longitude of the recommended pick-up point are output by fusing the matched orders. The final experiment results based on real-world datasets show that PPRM can efficiently and accurately provide users with ideal pick-up points.

Acknowledgments. We are grateful to anonymous reviewers for their helpful comments. This work is partially supported by the National Key Research and Development Program of China under Grant No. 2019YFB1600300.

References

1. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, pp. 226–231 (1996)
2. Fan, X., Guo, L., Han, N., Wang, Y., Shi, J., Yuan, Y.: A deep learning approach for next location prediction. In: 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 69–74. IEEE (2018)
3. Gong, Y., Li, Y., Jin, D., Su, L., Zeng, L.: A location prediction scheme based on social correlation. In: 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring), pp. 1–5. IEEE (2011)
4. Kwon, E., et al.: A novel location prediction scheme based on trajectory data. In: 2019 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1292–1294. IEEE (2019)
5. Lei, P.R., Li, S.C., Peng, W.C.: QS-STT: QuadSection clustering and spatial-temporal trajectory model for location prediction. Distrib. Parallel Databases **31**(2), 231–258 (2013). https://doi.org/10.1007/s10619-012-7115-1
6. Lei, P.R., Shen, T.J., Peng, W.C., Su, J.: Exploring spatial-temporal trajectory model for location prediction. In: 2011 IEEE 12th International Conference on Mobile Data Management, vol. 1, pp. 58–67. IEEE (2011)
7. Li, F., Li, Q., Li, Z., Huang, Z., Chang, X., Xia, J.: A personal location prediction method based on individual trajectory and group trajectory. IEEE Access **7**, 92850–92860 (2019)
8. Li, F., Li, Q., Li, Z., Huang, Z., Chang, X., Xia, J.: A personal location prediction method to solve the problem of sparse trajectory data. In: 2019 20th IEEE International Conference on Mobile Data Management (MDM), pp. 329–336. IEEE (2019)
9. Li, S., Qiao, J., Lin, S.: Location prediction method based on similarity of users moving behavior. Comput. Sci. **45**(12), 288–292+307 (2018)
10. Li, Y., Lei, L., Yan, M.: Mobile user location prediction based on user classification and Markov model. In: 2019 International Joint Conference on Information, Media and Engineering (IJCIME), pp. 440–444. IEEE (2019)
11. Liao, J., Liu, T., Liu, M., Wang, J., Wang, Y., Sun, H.: Multi-context integrated deep neural network model for next location prediction. IEEE Access **6**, 21980–21990 (2018)
12. Lin, S.K., Li, S.Z., Qiao, J.Z., Yang, D.: Markov location prediction based on user mobile behavior similarity clustering. J. Northeast. Univ. (Nat. Sci.) **37**(3), 323 (2016)
13. MacQueen, J.: Classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)

14. Mantoro, T., Olowolayemo, A., Olatunji, S.O., Osman, A., et al.: Extreme learning machine for user location prediction in mobile environment. Int. J. Perv. Comput. Commun. **7**(2), 162–180 (2011)
15. Qiao, J., Li, S., Lin, S.: Location prediction based on user mobile behavior similarity. In: 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS), pp. 783–786. IEEE (2017)
16. Tseng, V.S., Lu, E.H.C., Huang, C.H.: Mining temporal mobile sequential patterns in location-based service environments. In: 2007 International Conference on Parallel and Distributed Systems, pp. 1–8. IEEE (2007)
17. Wong, M.H., Tseng, V.S., Tseng, J.C.C., Liu, S.-W., Tsai, C.-H.: Long-term user location prediction using deep learning and periodic pattern mining. In: Cong, G., Peng, W.-C., Zhang, W.E., Li, C., Sun, A. (eds.) ADMA 2017. LNCS (LNAI), vol. 10604, pp. 582–594. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69179-4_41
18. Xu, C., Xu, C.: Predicting personal transitional location based on modified-SVM. In: 2017 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 340–344. IEEE (2017)
19. Xu, F., Yang, J., Liu, H.: Location prediction model based on ST-LSTM network. Comput. Eng. **9**, 136–138 (2019)
20. Yamada, N., Katsumaru, N., Nishijima, H., Kimoto, M.: Location prediction based on smartphone multimodal personal data for proactive support services. In: 2018 Eleventh International Conference on Mobile Computing and Ubiquitous Network (ICMU), pp. 1–2. IEEE (2018)
21. Yasser, K., Hemayed, E.: Novelty detection for location prediction problems using boosting trees. In: Gervasi, O., et al. (eds.) ICCSA 2017. LNCS, vol. 10405, pp. 173–182. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62395-5_13
22. Ying, J.J.C., Lee, W.C., Tseng, V.S.: Mining geographic-temporal-semantic patterns in trajectories for location prediction. ACM Trans. Intell. Syst. Technol. (TIST) **5**(1), 1–33 (2014)
23. Yoon, T.B., Park, K.H., Lee, J.H.: A spatiotemporal location prediction method of moving objects based on path data. J. Korean Inst. Intell. Syst. **16**(5), 568–574 (2006)
24. Zhang, D., Yang, N., Ma, Y.: Explicable location prediction based on preference tensor model. In: Cui, B., Zhang, N., Xu, J., Lian, X., Liu, D. (eds.) WAIM 2016. LNCS, vol. 9658, pp. 205–216. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39937-9_16
25. Zhang, H., Jiang, J., Zhou, H.: Method of mining user mobile rule based on pattern matching degree and location prediction. Comput. Sci. **36**(11), 3258–3261+3296 (2019)
26. Zhang, R., Guo, J., Jiang, H., Xie, P., Wang, C.: Multi-task learning for location prediction with deep multi-model ensembles. In: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 1093–1100. IEEE (2019)
27. Zhang, W., Sun, L., Wang, X., Huang, Z., Li, B.: SEABIG: a deep learning-based method for location prediction in pedestrian semantic trajectories. IEEE Access **7**, 109054–109062 (2019)
28. Zhou, C., Huang, B., Tu, L.: Exploiting collective spontaneous mobility to improve location prediction of mobile phone users. In: 2015 IEEE International Conference on Data Science and Data Intensive Systems, pp. 117–122. IEEE (2015)
29. Zolotukhin, M., Ivannikova, E., Hämäläinen, T.: Novel method for the prediction of mobile location based on temporal-spatial behavioral patterns. In: 2013 IEEE Third International Conference on Information Science and Technology (ICIST), pp. 761–766. IEEE (2013)

DP-Opt: Identify High Differential Privacy Violation by Optimization

Ben Niu[1], Zejun Zhou[1,2], Yahong Chen[1,2], Jin Cao[3], and Fenghua Li[1,2(✉)]

[1] Institute of Information Engineering, CAS, Beijing, China
zhouzejun@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] School of Cyber Engineering, Xidian University, Xi'an, China

Abstract. Differential privacy has become a golden standard for designing privacy-preserving randomized algorithms. However, such algorithms are subtle to design, as many of them are found to have incorrect privacy claim. To help identify this problem, one approach is designing disprovers to search for counterexamples that demonstrate high violation of claimed privacy level. In this paper, we present DP-Opt(mizer), a disprover that tries to search for counterexamples whose lower bounds on differential privacy exceed the claimed level of privacy guaranteed by the algorithm. We leverage the insights of counterexample construction proposed by the latest work, meanwhile resolve their limitations. We transform the search task into an improved optimization objective which takes into account the empirical error, then solve it with various off-the-shelf optimizers. An evaluation on a variety of both correct and incorrect algorithms illustrates that DP-Opt almost always produces stronger guarantees than the latest work up to a factor of 9.42, with runtime reduced by an average of 19.2%.

Keywords: Differential privacy · Disprover · Lower bounds

1 Introduction

Differential Privacy (DP) [11] has become a golden standard that measures the level of privacy guaranteed by randomized mechanisms. DP protects individual's information because attackers cannot tell if an output was generated from database a_1, or its neighbor $A = a_2$ with that individual's record changed. However, designing such differentially private mechanisms can be error-prone, as existing papers have already identified incorrect privacy claims of published mechanisms [8,15]. Therefore, an important area of research is to verify the privacy level of a differentially private mechanism.

Generally, related works are divided into three types: formal verification, disprover, and the combination of both. Formal verification methods develop a proof system and use it to *prove* that mechanisms satisfy differential privacy [1,4,5, 18,20]. However, these techniques are not able to *disprove* an incorrect privacy

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 406–416, 2022.
https://doi.org/10.1007/978-3-031-19214-2_34

claim. On the contrary, disprovers try to search for counterexamples of a mechanism that violate the claimed differential privacy level [2,6,7,10,19]. Typically, two approaches are taken. On one hand, StatDP [10] constructs and tests a statistical hypothesis. Given a preconceived privacy parameter $\epsilon_0 > 0$, it tries to find a counterexample that violates the privacy condition, therefore rejects incorrect mechanisms. On the other hand, works like DP-Finder [6] and DP-Sniper [7] search for the lower bound on differential privacy. Such lower bound is found by maximizing the privacy loss function derived from DP definition. Inputs to this function is considered a counterexample if the corresponding lower bound exceeds the claimed privacy level. Both approaches try to identify counterexamples so as to demonstrate that the privacy claim is incorrect, and further provide insights for developers to fix the bugged mechanism. As opposed to formal verification methods, a disprover cannot prove that a mechanism satisfies the claimed privacy if it fails to generate any counterexample. Another type of methods [3,14,17] combines the previous two methods, and either synthesizes proofs for correct mechanisms or generates counterexamples for incorrect mechanisms.

This Work. We present an enhanced disprover DP-Opt, which aims to resolve the limitations in counterexample construction of the latest work DP-Sniper [7] and produce higher privacy violations. Specifically, our contributions are:

- DP-Opt, an algorithm that leverages the idea of optimization to resolve the limitations of DP-Sniper by transforming the search task into an improved optimization objective to be solved with off-the-shelf numerical optimizers.
- An implementation[1] and evaluation of DP-Opt on a wide variety of randomized algorithms demonstrating significantly higher guarantees on privacy by a factor up to 9.42 with an average reduced runtime of 19.2%, compared with the latest work.

2 DP Disprover Background

2.1 Differntial Privacy

Formally, given a mechanism $M : \mathbb{A} \rightarrow \mathbb{B}$ that inputs database $a \in \mathbb{A}$ and outputs $b \in \mathbb{B}$, M is ϵ-differentially private (ϵ-DP) if for every pair of neighboring inputs $(a_1, a_2) \in \mathcal{N}$ and for every attack $\mathcal{S} \subseteq \mathbb{B}$,

$$\ln\left(\Pr[M(a_1) \in \mathcal{S}]\right) - \ln\left(\Pr[M(a_2) \in \mathcal{S}]\right) \leq \epsilon, \tag{1}$$

where the neighborhood $\mathcal{N} \subseteq \mathbb{A} \times \mathbb{A}$ consists of neighboring database pairs that differ in only one record. The privacy parameter $\epsilon \in [0, \infty)$ quantifies the privacy level guaranteed by M, where smaller ϵ corresponds to higher privacy guarantees, and contrarily, $\epsilon = \infty$ means no privacy at all.

2.2 Prior Knowledge of DP-Sniper

Power. Derived from Eq. 1, *power* [7] of a witness $(a_1, a_2, \mathcal{S})$ is defined as

$$\mathcal{E}(a_1, a_2, \mathcal{S}) := \ln\left(\Pr[M(a_1) \in \mathcal{S}]\right) - \ln\left(\Pr[M(a_2) \in \mathcal{S}]\right).$$

[1] Available at https://github.com/barryZZJ/dp-opt.

The highest power found by disprover is regarded as the *lower bound* on the privacy level of M. Therefore, we aim to find the maximum power so as to measure the level of violation against the claimed privacy of M.

Estimation. With samples $b^{(0)}, \ldots, b^{(N-1)} \sim M(a)$, we can estimate the probability $\Pr[M(a) \in \mathcal{S}]$ as $\hat{P}^N_{M(a)\in\mathcal{S}} = \frac{1}{N}\sum_{i=0}^{N-1} \Pr[b^{(i)} \in \mathcal{S}]$. Therefore, estimation of power $\hat{\mathcal{E}}(a_1, a_2, \mathcal{S})$ is computed by replacing the probability terms with their estimations.

Threshold Attack. Threshold attack [7] is a type of randomized attack that selects b probabilistically according to the membership function $\mathcal{S}^{t,q} : \mathbb{B} \rightarrow [0, 1]$. Specifically, it utilizes the novel idea of posterior probability $p(a_1|b)$ that defines the probability that an output b originates from $M(a_1)$, as opposed to $M(a_2)$. A threshold attack incorporates the output whose posterior probability is above some threshold t, in order to produce high power. Additionally, an output is only included with probability q if its posterior probability is equal to t. This limits the size of the threshold attack and ensures continuousness of power. Formally, the membership function of threshold attack $\mathcal{S}^{t,q}(b)$ is defined as

$$\Pr[b \in \mathcal{S}^{t,q}] = [p(a_1|b) > t] + q \cdot [p(a_1|b) = t], \tag{2}$$

where the Iverson bracket $[\phi]$ outputs 1 if ϕ is true, and 0 otherwise. Moreover, estimation of $\Pr[M(a) \in \mathcal{S}^{t,q}]$ is computed as

$$\hat{P}^N_{M(a)\in\mathcal{S}^{t,q}} = \frac{1}{N}\sum_{i=0}^{N-1}[p(a_1|b^{(i)}) > t] + \frac{1}{N} \cdot q \sum_{i=0}^{N-1}[p(a_1|b^{(i)}) = t]. \tag{3}$$

Parameter c**.** According to [7], the deviation of $\hat{P}^N_{M(a)\in\mathcal{S}}$ increases rapidly when it becomes smaller, causing estimation on power unreliable. To avoid this issue, DP-Sniper discards small probabilities below some constant $c \in (0, 1]$. However, this predefined parameter makes a considerable impact on results, as illustrated in the next section.

3 Motivation and Ideas

3.1 Limitations of DP-Sniper

We now demonstrate the issue of predefining c with the following example.

Example 1. Consider the 0.5-DP Laplace mechanism $\mathcal{L}_{0.5}(a) = a + \mathrm{lap}(0, 2)$, which adds Laplace noise with mean 0 and scale $1/0.5 = 2$ to its input $a \in \mathbb{R}$ [7, Ex. 1]. The top plot in Fig. 1 shows the cumulative distribution function of $\mathcal{L}_{0.5}(0)$ and $\mathcal{L}_{0.5}(1)$ (blue and orange solid line respectively) by constructing the attack $\mathcal{S}^{t,q} = (-\infty, b)$, with c set to $c^* = 0.2$ (red dashed line). The bottom plot demonstrates the corresponding power by the brown solid line.

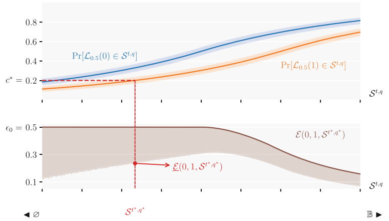
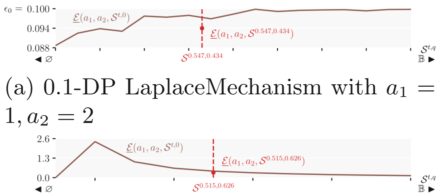


Fig. 1. Cumulative distribution function and power of $\mathcal{L}_{0.5}$, with confidence intervals indicated by the shaded area. (Color figure online)



(a) 0.1-DP LaplaceMechanism with $a_1 = 1, a_2 = 2$

(b) ∞-DP NoisyMax4 with $a_1 = [1, 1, 1, 1, 1], a_2 = [2, 2, 2, 2, 2]$

Fig. 2. Experiment values of $\underline{\mathcal{E}}(a_1, a_2, \mathcal{S}^{t,0})$.

In Example 1, for a pair of neighboring inputs $(0, 1) \in \mathcal{N}$, DP-Sniper constructs the threshold attack $\mathcal{S}^{t^*,q^*}$ by selecting t^*, q^* that satisfy $\Pr[\mathcal{L}_{0.5}(1) \in \mathcal{S}^{t^*,q^*}] = 0.2$. This automatically ensures $\Pr[\mathcal{L}_{0.5}(0) \in \mathcal{S}^{t^*,q^*}] \geq 0.2$ according to the properties of posterior probability. After the external algorithm DD-Search [7] generates different neighboring inputs, it invokes DP-Sniper and selects the best witness constructed, in this case $(0, 1, \mathcal{S}^{t^*,q^*})$. With this, it calculates the lower bound on power $\underline{\mathcal{E}} \approx 0.2$ (discussed in Sect. 3.2), as indicated by the red dot in Fig. 1. However, as the brown shaded area demonstrates, better lower bound on power can be achieved if c^* was initialized otherwise (in this case to around 0.3).

In fact, this problem occurs for almost all mechanisms, as confirmed by our experiments shown in Fig. 2. We enumerated attacks $\mathcal{S}^{t,0}$ of various t (with $q = 0$ for simplicity) and computed each $\underline{\mathcal{E}}$ for 0.1-DP LaplaceMechanism (Fig. 2a) and ∞-DP NoisyMax4 (Fig. 2b). The red dot in each plot is the final lower bound produced by DD-Search.

3.2 Ideas

Determine Optimization Objective. Inspired by Fig. 1, we decide to skip the procedure of determining c, and aims to find a threshold attack $\mathcal{S}^{t^\triangle,q^\triangle}$ that directly maximizes the lower bound on power $\underline{\mathcal{E}}$ for given $(a_1, a_2) \in \mathcal{N}$:

$$\mathcal{S}^{t^\triangle,q^\triangle} = \operatorname*{arg\,max}_{t\in[0,1],\ q\in[0,1]} \underline{\mathcal{E}}\left(a_1, a_2, \mathcal{S}^{t,q}\right).$$

We note that our work also discards small probabilities which induce high deviation, because the lower bound $\underline{\mathcal{E}}$ represents the worst-case scenario, finding the highest $\underline{\mathcal{E}}$ automatically leaves out imprecise probability estimations.

Now we describe the derivation of the optimization objective. Given mechanism M and neighboring inputs (a_1, a_2), for each output b, its posterior probability $p(a_1|b)$ is determined. Thus $\Pr[b \in \mathcal{S}^{t,q}]$ only varies by different combination of $t \in [0, 1]$ and $q \in [0, 1]$ (recall Eq. 2). Then, according to Eq. 3, $\hat{P}^N_{M(a)\in\mathcal{S}^{t,q}}$ also only relies on t, q. Therefore, given neighboring inputs $(a_1, a_2) \in \mathcal{N}$ of mechanism M, lower bound on power $\underline{\mathcal{E}}(a_1, a_2, \mathcal{S}^{t,q})$ can be regarded as a function of $\mathcal{S}^{t,q}$ determined by t, q.

As a result, the aim of our work is to search for the best combination of variables t, q such that the corresponding threshold attack $\mathcal{S}^{t,q}$ produces the highest $\underline{\mathcal{E}}$. This is a maximization problem of a bivariate scalar function $\underline{\mathcal{E}}(t, q)$ under the constraint of $t \in [0, 1], q \in [0, 1]$:

$$t^{\triangle}, q^{\triangle} = \underset{t \in [0,1],\ q \in [0,1]}{\arg\max}\ \underline{\mathcal{E}}(t, q). \tag{4}$$

In addition, the impact of q can be ignored for some mechanisms if $p(a_1|b^{(i)}) = t$ in Eq. 3 rarely occurs. Hence, we transform Eq. 4 into a maximization problem of a univariate scalar function $\underline{\mathcal{E}}(t, 0)$ constrained by $t \in [0, 1]$:

$$t^{\triangle} = \underset{t \in [0,1]}{\arg\max}\ \underline{\mathcal{E}}(t, 0), \tag{5}$$

in expectation of better results in special cases.

Confidence Intervals of Power. We now discuss confidence intervals and derive bounds on power inspired by [6,7]. First, we apply the Clopper-Pearson confidence interval [9] on $\hat{P}^{N}_{M(a) \in \mathcal{S}}$ in order to derive the upper bound $\overline{P}^{N,\alpha/2}_{M(a) \in \mathcal{S}}$ and the lower bound $\underline{P}^{N,\alpha/2}_{M(a) \in \mathcal{S}}$, which both hold except with probability $\alpha/2$. In the top plot of Fig. 1, such bounds on probabilities are illustrated by the blue and orange shaded areas around respective solid lines. Then, we can use them to derive the bounds on power $\hat{\mathcal{E}}(a_1, a_2, \mathcal{S})$.

Theorem 1. *For neighboring inputs $(a_1, a_2) \in \mathcal{N}$, lower bound $\underline{\mathcal{E}}(a_1, a_2, \mathcal{S})$ and upper bound $\overline{\mathcal{E}}(a_1, a_2, \mathcal{S})$ on $\hat{\mathcal{E}}(a_1, a_2, \mathcal{S})$ both hold with probability $1 - \alpha$, where*

$$\underline{\mathcal{E}}\left(a_1, a_2, \mathcal{S}\right) = \ln\left(\underline{P}^{N,\alpha/2}_{M(a_1) \in \mathcal{S}}\right) - \ln\left(\overline{P}^{N,\alpha/2}_{M(a_2) \in \mathcal{S}}\right),$$
$$\overline{\mathcal{E}}\left(a_1, a_2, \mathcal{S}\right) = \ln\left(\overline{P}^{N,\alpha/2}_{M(a_1) \in \mathcal{S}}\right) - \ln\left(\underline{P}^{N,\alpha/2}_{M(a_2) \in \mathcal{S}}\right).$$

The lower bound on power is depicted by the brown shaded area below the brown solid line in Fig. 1. This bound holds even if probability estimations $\hat{p}$ are imprecise, because Clopper-Pearson interval is a type of *exact* interval [16] which has a coverage probability of *at least* $1 - \alpha$ for all values of $\hat{p}$. For this reason, we use $\underline{\mathcal{E}}(a_1, a_2, \mathcal{S})$ as both the optimization objective and final output, and furthermore conclude that the privacy level of M is at best $\underline{\mathcal{E}}$ with probability $1 - \alpha$ at least.

4 Our Disprover

In this section, we present the flow of our disprover. We first introduce DP-Opt that searches for optimal threshold attack by solving the optimization objective. Then we propose the external algorithm PowerSearcher that utilizes DP-Opt and produces the highest lower bound on power.

4.1 DP-Opt: Search for Optimal Threshold Attack

Given a neighboring input pair (a_1, a_2), DP-Opt searches for the optimal threshold attack with the following steps. First, a machine learning classifier $p_\theta(a_1|b)$ parametrized by θ is trained with N_{train} samples. It approximates the posterior probability $p(a_1|b)$ described in Sect. 2.2. We refer to [7] for more details as this is not focused in our work. Then, we exploit off-the-shelf numerical optimizers to solve the optimization objectives Eq. 4 and Eq. 5. Each optimizer tries to maximize $\underline{\mathcal{E}}(t, q)$ or $\underline{\mathcal{E}}(t, 0)$ with N_{check} samples. Among them, the maximum $\underline{\mathcal{E}}$ is selected, along with the inputs t^Δ, q^Δ. Finally, the optimal threshold attack $\mathcal{S}^{t^\Delta, q^\Delta}$ for the given input pair is constructed using parameters t^Δ, q^Δ, and returned by DP-Opt.

4.2 PowerSearcher: Search for High Privacy Violation

Guided by DD-Search [7], we discuss the details of PowerSearcher that leverages DP-Opt to find the highest $\underline{\mathcal{E}}$. First, different neighboring input pairs $(a_1^{(i)}, a_2^{(i)})$ are generated based on heuristic patterns [10]. For each input pair, a candidate witness is constructed by combining the input pair with corresponding optimal attack $\mathcal{S}^{(i)}$. Then, among all candidate witnesses, the optimal witness is selected according to its lower bound $\underline{\mathcal{E}}(a_1, a_2, \mathcal{S})$ computed with N_{check} samples. While most works compare the estimation on power, we compare the lower bound in order to avoid high deviation caused by small probability. In implementation, we reuse the maximum value found in step two to reduce computational cost. Finally, the lower bound on power of the optimal witness is computed again with fresh N_{final} samples and returned by PowerSearcher, along the witness. In this step, the sample size N_{final} is larger than N_{check} to produce a tighter bound.

5 Evaluation

5.1 Implementation

Inherited from [7], we implemented DP-Opt and PowerSearcher in Python based on the notion from Li et al. [13]. Since different classifiers have insignificant impact on performance [7], we only choose logistic regression classifier due to time limitation. Additionally, in attack searching, we applied binary search and reused the same sample on different optimizers. This substantially reduced runtime as computing and optimizing $\underline{\mathcal{E}}(t, q)$ need to repeatedly estimate probabilities and try various combinations of t, q.

Input Pattern Generation. We used the heuristic patterns proposed by Ding et al. [10] for input generation. For example, category *one above* corresponds to $(a_1, a_2) = ([1, 1, 1, 1, 1], [2, 1, 1, 1, 1])$.

Parameters. Following the guideline in [7], we used sample sizes $N_{\text{check}} = N_{\text{train}} = 10.7 \cdot 10^6$, and $N_{\text{final}} = 2 \cdot 10^8$, with $\alpha = 0.1$. The logistic regression model is trained using regularized stochastic gradient descent optimization and binary cross entropy loss, with epoch number 10, learning rate 0.3, momentum 0.3 and regularization weight 0.001.

Optimizers. Upon comparison, we selected several optimizers provided by SciPy in consideration of both performance and runtime cost. Their orders are as follows: Nelder-Mead(bi), Nelder-Mead(uni), COBYLA(bi), Differential Evolution(bi), Differential Evolution(uni), Powell(bi), COBYLA(uni), where *bi* and *uni* correspond to bivariate optimization objective $\underline{\mathcal{E}}(t, q)$ and univariate optimization objective $\underline{\mathcal{E}}(t)$ respectively. In implementation, we set initial guesses $t_0 = 0.5, q_0 = 0.5$, and kept the default values for the rest optional parameters.

5.2 Mechanisms Evaluated

We evaluated mechanisms listed in Table 1, including widely used mechanisms and their variations. They cover a variety of output types such as reals, integers and boolean values. The second column is their neighborhood definition, where $\| \cdot \|_p$ is the p-norm neighborhood $\mathcal{N} = \{(a_1, a_2) \mid \|a_1 - a_2\|_p \leq 1\}$.

Table 1. Evaluated mechanisms with their neighborhoods, expected DP and optimization objectives.

Mechanism	$\mathcal{N}$	ϵ	Objective
LaplaceMechanism [11]	$\|\| \cdot \|\|_1$	0.1	$\underline{\mathcal{E}}(t, q), \underline{\mathcal{E}}(t)$
NoisyHist1 [10, Alg. 9]	$\|\| \cdot \|\|_1$	0.1	$\underline{\mathcal{E}}(t, q), \underline{\mathcal{E}}(t)$
NoisyHist2 [10, Alg. 10]	$\|\| \cdot \|\|_1$	10	$\underline{\mathcal{E}}(t, q), \underline{\mathcal{E}}(t)$
NoisyMax1 [10, Alg. 5]	$\|\| \cdot \|\|_\infty$	0.1	$\underline{\mathcal{E}}(t, q)$
NoisyMax2 [10, Alg. 6]	$\|\| \cdot \|\|_\infty$	0.1	$\underline{\mathcal{E}}(t, q)$
NoisyMax3 [10, Alg. 7]	$\|\| \cdot \|\|_\infty$	∞	$\underline{\mathcal{E}}(t, q), \underline{\mathcal{E}}(t)$
NoisyMax4 [10, Alg. 8]	$\|\| \cdot \|\|_\infty$	∞	$\underline{\mathcal{E}}(t, q), \underline{\mathcal{E}}(t)$
SVT1 [15, Alg. 1]	$\|\| \cdot \|\|_\infty$	0.1	$\underline{\mathcal{E}}(t, q)$
SVT2 [15, Alg. 2]	$\|\| \cdot \|\|_\infty$	0.1	$\underline{\mathcal{E}}(t, q)$
SVT3 [15, Alg. 3]	$\|\| \cdot \|\|_\infty$	∞	$\underline{\mathcal{E}}(t, q)$
SVT4 [15, Alg. 4]	$\|\| \cdot \|\|_\infty$	0.175	$\underline{\mathcal{E}}(t, q)$
SVT5 [15, Alg. 5]	$\|\| \cdot \|\|_\infty$	∞	$\underline{\mathcal{E}}(t, q)$
SVT6 [15, Alg. 6]	$\|\| \cdot \|\|_\infty$	∞	$\underline{\mathcal{E}}(t, q)$
OneTimeRAPPOR [12, Steps 1–2]	$\|\| \cdot \|\|_1$	0.8	$\underline{\mathcal{E}}(t, q), \underline{\mathcal{E}}(t)$
RAPPOR [12, Steps 1–3]	$\|\| \cdot \|\|_1$	0.4	$\underline{\mathcal{E}}(t, q), \underline{\mathcal{E}}(t)$

Parameter Configuration. We set the parameters for each mechanism in accordance with DP-Sniper. Specifically, let ϵ_0 be the target DP guarantee,

- LaplaceMechanism uses $\epsilon_0 = 0.1$.
- NoisyHist1-2 and NoisyMax1-4 set $\epsilon_0 = 0.1$ with input length 5.
- SVT1-6 are instantiated by $\epsilon_0 = 0.1$ with input length 10 and additional parameters $c = 1, \varDelta = 1, T = 1$ (except $T = 0.5$ for SVT1).
- OneTimeRAPPOR is initialized with parameters $k = 20, h = 4, f = 0.95$.
- RAPPOR is parametrized by $k = 20, h = 4, f = 0.75, q = 0.55$.

The corresponding expected privacy guarantees are listed in the third column of Table 1. For all mechanisms, we ran our disprover on each optimization objective (indicated by the last column) for seven times with suitable optimizers.

5.3 Results

Power. Figure 3 compares the average value of the final lower bound $\underline{\mathcal{E}}$ found between PowerSearcher and DD-Search. Results show that PowerSearcher is generally better with at least equal results in certain cases. Specifically, for most mechanisms with finite privacy target, PowerSearcher found tighter bounds, resolving the uncertainty of DD-Search's conclusion. For example, for NoisyHist1, DD-Search only narrows ϵ to $[0.098, 0.1]$ while PowerSearcher ensures it to be 0.1-DP. Especially, we manage to demonstrate NoisyHist2 to be 10-DP correctly in contrast to DD-Search only results in 4.605. For mechanisms known to be ∞-DP, PowerSearcher performs significantly better by a factor up to 9.42, except for NoisyMax3 which is 0.25-DP when input length is 5 (our configuration) [7, Sect. VI]. Unfortunately, for state-of-the-art mechanisms such as RAPPOR, PowerSearcher fails to derive better results. We attribute this to be the fundamental inability of threshold attacks.

Runtime. Figure 4 compares the runtime between PowerSearcher and DD-Search for each mechanism. We managed to reduce an average of 19.2% of runtime consumption, after exploiting the improvement methods mentioned in Sect. 4.2 and Sect. 5.1. We note that since our method is more flexible in choosing optimizers, trade-off between performance and runtime can be further made.

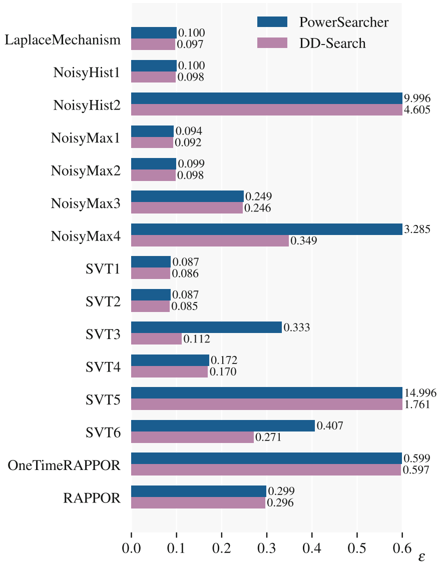


Fig. 3. Average $\underline{\mathcal{E}}$ found between PowerSearcher and DD-Search, where higher values are better.

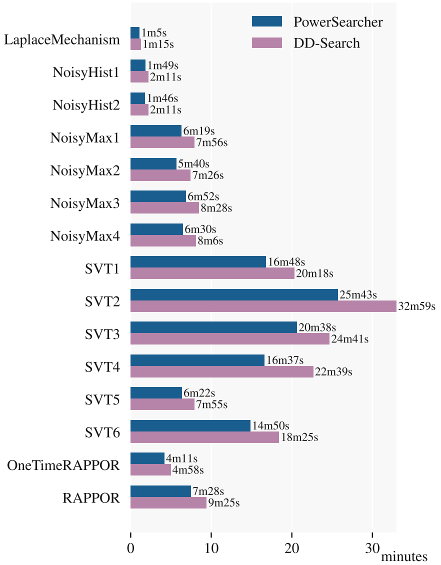


Fig. 4. Runtime of PowerSearcher and DD-Search.

6 Conclusion

We proposed DP-Opt, an improved disprover on the latest work by maximizing the lower bound on privacy level for a given mechanism. It exploits off-the-shelf optimizers to produce threshold attacks that yield optimal lower bound on power, and also avoids small probabilities that are difficult to estimate accurately. Results demonstrate significant improvement on privacy bounds compared with the latest baseline, with a fair amount of runtime saved. Future works are expected to employ an optimal optimizer that generalizes well on all optimization objectives to greatly reduce runtime while preserving better results.

Acknowledgements. This work is supported by the National Key R&D Program of China (2021YFB3100300), the National Natural Science Foundation of China (61872441, 61932015), and the Youth Innovation Promotion Association, Chinese Academy of Sciences (2019160, 2021154).

References

1. Albarghouthi, A., Hsu, J.: Synthesizing coupling proofs of differential privacy. Proc. ACM Program. Lang. **2**(POPL), 1–30 (2017)
2. Askin, Ö., Kutta, T., Dette, H.: Statistical quantification of differential privacy: a local approach. arXiv preprint arXiv:2108.09528 (2021)

3. Barthe, G., Chadha, R., Jagannath, V., Sistla, A.P., Viswanathan, M.: Deciding differential privacy for programs with finite inputs and outputs. In: Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science, pp. 141–154 (2020)
4. Barthe, G., Gaboardi, M., Grégoire, B., Hsu, J., Strub, P.Y.: Proving differential privacy via probabilistic couplings. In: Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, pp. 749–758 (2016)
5. Barthe, G., Köpf, B., Olmedo, F., Zanella Beguelin, S.: Probabilistic relational reasoning for differential privacy. In: Proceedings of the 39th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pp. 97–110 (2012)
6. Bichsel, B., Gehr, T., Drachsler-Cohen, D., Tsankov, P., Vechev, M.: DP-Finder: finding differential privacy violations by sampling and optimization. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 508–524 (2018)
7. Bichsel, B., Steffen, S., Bogunovic, I., Vechev, M.: DP-Sniper: black-box discovery of differential privacy violations using classifiers. In: 2021 IEEE Symposium on Security and Privacy (SP), pp. 391–409 (2021)
8. Chen, Y., Machanavajjhala, A.: On the privacy properties of variants on the sparse vector technique. arXiv preprint arXiv:1508.07306 (2015)
9. Clopper, C.J., Pearson, E.S.: The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika **26**(4), 404–413 (1934)
10. Ding, Z., Wang, Y., Wang, G., Zhang, D., Kifer, D.: Detecting violations of differential privacy. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 475–489 (2018)
11. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
12. Erlingsson, Ú., Pihur, V., Korolova, A.: RAPPOR: randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 1054–1067 (2014)
13. Li, F., Li, H., Niu, B., Chen, J.: Privacy computing: concept, computing framework, and future development trends. Engineering **5**(6), 1179–1192 (2019)
14. Liu, D., Wang, B.Y., Zhang, L.: Verifying pufferfish privacy in hidden Markov models. In: International Conference on Verification, Model Checking, and Abstract Interpretation, pp. 174–196 (2022)
15. Lyu, M., Su, D., Li, N.: Understanding the sparse vector technique for differential privacy. arXiv preprint arXiv:1603.01699 (2016)
16. Thulin, M.: The cost of using exact confidence intervals for a binomial proportion. Electron. J. Stat. **8**(1), 817–840 (2014)
17. Wang, Y., Ding, Z., Kifer, D., Zhang, D.: CheckDP: an automated and integrated approach for proving differential privacy or finding precise counterexamples. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, pp. 919–938 (2020)
18. Wang, Y., Ding, Z., Wang, G., Kifer, D., Zhang, D.: Proving differential privacy with shadow execution. In: Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation, pp. 655–669 (2019)

19. Wilson, R.J., Zhang, C.Y., Lam, W., Desfontaines, D., Simmons-Marengo, D., Gipson, B.: Differentially private SQL with bounded user contribution. arXiv preprint arXiv:1909.01917 (2019)
20. Zhang, D., Kifer, D.: LightDP: towards automating differential privacy proofs. In: Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, pp. 888–901 (2017)

A Fast Direct Position Determination with Embedded Convolutional Neural Network

Rui Xia, Jingchao Wang(✉), Boyu Deng, and Fang Wang

Academy of Military Sciences, Beijing, China
wangjc.2000@tsinghua.org.cn

Abstract. The direct position determination (DPD) method is more accurate than the previous two-step method in passive positioning. Enormous computational complexity in DPD is a severe drawback, which causes both real-time and high-accuracy to be challenging to satisfy. We integrate the formulas of the DPD and the image regression technique in computer vision, offering the unified computational graphs of both to investigate the fundamental reason for large time consumption in DPD. To achieve efficient DPD, we propose a fast DPD with an embedded convolutional neural network (CNNDPD), which is an end-to-end passive positioning network. We use a wavelet transform two-dimensionalization to convert the time domain signal into a time-frequency map and extract the time-frequency attributes effectively for the received data required for positioning. Other information required for positioning is stitched with the findings of time-frequency map processing and sent into fully-connected networks, allowing fuse with time-frequency information effectively. The simulation results show that the CNNDPD has the advantage of fast and highly accurate positioning. In a wide-area localization setting, CNNDPD has 26 times and 46 times faster inference speed than exhaustive search DPD and genetic algorithm DPD, respectively, without reducing accuracy. Furthermore, CNNDPD has a lower false alarm rate than the two benchmarks.

Keywords: Direct position determination · Computer vision · Convolutional neural network · Maximum likelihood estimation

1 Introduction

Passive positioning is the technology that determines the location of an emitter using the signals intercepted by single or multiple receivers. Different from active positioning, passive positioning technology has the advantages of high concealment, strong anti-interference, and low cost in the process of achieving the position of the emitter. These advantages make it widely used in radar and wireless communications [1,3,4,7,8,18,20–22,24,25].

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 417–428, 2022.
https://doi.org/10.1007/978-3-031-19214-2_35

Passive positioning can be mainly divided into the two-step method and direct position determination (DPD). The two-step method can be divided into two steps: estimation of localization parameters and equation solving. The localization parameters estimated in the first step are common: phase difference [1,24], time difference of arrival (TDOA) [3,7], frequency difference of arrival (FDOA) [8,18], doppler rate [4,25]. The most widely used method for estimation of these parameters is the Cross-Ambiguity Function. It obtains the maximum value of the function by searching for feasible localization parameters. In the second step, the parameters estimated in the first step are used to establish and solve the equations. Obviously, the accuracy of solution is limited by the estimation accuracy of the parameter in the first step. Meanwhile, the parameters are difficult to be correlated in the two-step method and only obtain accuracy-restriction results. The DPD [20–22] computes the intercepted signal data instead of estimating the parameters to determine the emitter's position. By looking for the feasible localization that optimizes the cost function, it takes into account the relationship between different parameters. Thus, DPD outperforms the two-step method in terms of localization accuracy [13,19,21,22] and is a more promising approach [4].

The most widely used DPD method is maximum likelihood estimation based (MLE-based) DPD [22]. The most common method to solve the DPD cost function is exhaustive search (ES). Since it needs to traverse all points within the range of possible emitter locations to get the final results, it is challenging to meet the demand for real-time processing, especially in a wide area. This is because a wide area will bring more possible emitter locations and more emitter data. Such as, in satellite positioning scenarios, since the satellite often has a coverage diameter of several hundred to thousands of kilometers, the time of computation required for reliable accuracy results of all emitters is often intolerable. In practice, the estimated result's value is strongly inversely related to the time consumed. When the computation time exceeds a certain threshold, the results will be worthless. For this purpose, we need to reduce the estimation time of the emitter's position. However, because of the setting of scale resolution during the discrete approximation solution of the continuous optimization problem, ES cannot satisfy both the accuracy and the real-time processing. There were two main methods to reduce the time-consuming calculation. One was to apply heuristic algorithms to solve the cost function, e.g., genetic algorithms (GA) [12,15] and particle swarm algorithms [23]. The other one was to improve the form of the cost function, such as expectation maximization (EM) [17], alternate projection (AP) [26]. However, all the aforementioned methods mentioned suffered from the common problem: local extremum problem or sensitivity to parameter initialization, which causes unreliable results. Since deep learning can process data efficiently in parallel during the inference phase, we try to use it to solve this problem.

We propose an end-to-end method, which is called fast DPD with embedded CNN (CNNDPD). In a wide-area localization setting, CNNDPD has 26 times and 46 times faster inference speed than ES DPD and GA DPD, respectively, without reducing accuracy. Furthermore, CNNDPD has a lower false alarm rate than the two benchmarks. Our contribution can be summarized as follows:

– We investigate the similarities between MLE-based DPD and computer vision (CV) perception problems by rewriting both formulas.
– We propose the CNNDPD, an end-to-end method, to achieve efficient DPD. We show how to feed several kinds of input into the CNN effectively in the DPD problem: received time-domain signals are time-frequency translated and supplied into the convolutional layer, while the velocities and positions of receivers are fed into the fully connected layer.
– The experimental results show that our proposed method has higher inference efficiency with comparable accuracy than traditional methods.

2 Background and Preliminary

2.1 MLE-Based DPD

Consider a three-dimensional space with N receivers whose positions are $\boldsymbol{p_i}$ and velocities are $\boldsymbol{v_i}$, all of which $\in \mathbb{R}^{3\times1}$. It exists a emitter whose position is $\boldsymbol{p}$ and speed is $\boldsymbol{v}$, all of which $\in \mathbb{R}^{3\times1}$. The unknown complex signal send at a particular time is $\boldsymbol{s}$. The carrier frequency is f_c. The sampling frequency of the receiver is f_s. The sampling time of the receiver is T. We assume that both the velocity and position of receivers and emitter are constant within the T. All receivers are synchronized with each other in time and frequency, and the received signals can be expressed as

$$\boldsymbol{y}_i(t) = a_i\boldsymbol{s}(t-\tau_i)e^{j2\pi f_i(t-\tau_i)} + \boldsymbol{n}_i(t) \tag{1}$$

where the receiver number $i = 1, 2, ..., N$, the time $t \in [0, T]$, and the attenuation coefficient of the signal received by the $i\text{-}th$ receiver $a_i \in (0, 1]$. Let the time delay and frequency delay of the signals received by the $i\text{-}th$ receiver $\tau_i = \frac{1}{C}||\boldsymbol{p} - \boldsymbol{p_i}||$ and $f_i = \frac{f_c}{C}\frac{(\boldsymbol{v}_i-\boldsymbol{v})^T(\boldsymbol{p}-\boldsymbol{p}_i)}{||\boldsymbol{p}-\boldsymbol{p}_i||}$, respectively (each receiver does down-sampling on the received signal by default), where $[\cdot]^T$ is the transpose operation, and C is the speed of propagation. The $\boldsymbol{n}_i$ is the noise of the $i-th$ received signal, which is assumed to be complex Gaussian noise and independent of each other. The mean and standard deviation of the noise are zero and σ, respectively.

After doing the K-sampling discretization, the Eq. 1 can be represented as

$$\boldsymbol{y}_i = a_i\boldsymbol{F}_i\boldsymbol{T}_i\boldsymbol{s} + \boldsymbol{n}_i \tag{2}$$

Let the received signal $\boldsymbol{y}_i = [y_i(1), y_i(2), ..., y_i(K)]^T$, the signal of emitter $\boldsymbol{s} = [s(1), s(2), ..., s(K)]^T$, and the noise $\boldsymbol{n}_i = [n_i(1), n_i(2), ..., n_i(K)]^T$. The time shift matrix $\boldsymbol{T}_i = \{t^i_{rc}\}_{K\times K}$ generates the time shift of the signal. When $r-c = \frac{\tau_i}{(T/K)}$, $t^i_{rc} = 1$, otherwise $t^i_{rc} = 0$. The frequency shift matrix $\boldsymbol{F}_i = \{f^i_{rc}\}_{K\times K}$ generates the frequency shift of the signal. When $c = r$, $f^i_{rc} = e^{j2\pi f_i(\frac{cT}{K}-\tau_i)}$, otherwise $f^i_{rc} = 0$. The $r = 1, 2, ..., K$ and $c = 1, 2, ..., K$ are the index of the rows and columns of the matrix, respectively. When $\boldsymbol{s}$ is unknown, we can obtain the MLE-based emitter position estimation equation [22].

$$\hat{\boldsymbol{p}} = \arg\max_{\boldsymbol{p}}(L(\boldsymbol{p})) \tag{3}$$

Let $L(\boldsymbol{p}) = \lambda_{max}\{\bar{\boldsymbol{Q}}\}$, $\bar{\boldsymbol{Q}} = \boldsymbol{V}^H\boldsymbol{V}$, and $\boldsymbol{V} = [\boldsymbol{T}_1^H\boldsymbol{F}_1^H\boldsymbol{y}_1, ..., \boldsymbol{T}_N^H\boldsymbol{F}_N^H\boldsymbol{y}_N]$. The $[\cdot]^H$ is the $[\cdot]^T$ with conjugate and $\lambda_{max}\{\cdot\}$ is the operation of taking the maximum eigenvalue.

2.2 CNN-Based CV

CV technology aims to recognize and understand the content in images or videos. It can be divided into two basic classes: classification and regression. Networks targeting the classification such as LeNet-5 [11] and AlexNet [16]. Networks targeting the both classes such as Fast R-CNN [5], YOLO [14], and Mask R-CNN [6]. Despite the different network structures, the common and core part is the convolution operation. The fundamental reason is that the convolution operation has an efficient extraction capability of image features due to its three characteristics: sparse interaction, parameter sharing, and translation invariance [2].

A typical network leveraging convolution operation is CNN. The basic structure of CNN consists of the convolution layers, the pooling layers, and the fully connected layers. The usage of CNN can mainly be divided into two separate phases, namely training and testing. The training phase can be divided into steps of information forward propagation (FP) and error backward propagation (BP). In the information FP step, the activation values of each network layer are updated with Eq. 4. In the error BP step, each layer's weights are updated with Eq. 5. In the testing phase, only the FP of information is completed.

$$\begin{cases} Conv : \boldsymbol{a}^l = \varphi(\boldsymbol{z}^l) = \varphi(\boldsymbol{w}^l * \boldsymbol{a}^{l-1} + \boldsymbol{b}^l) \\ Pool : \boldsymbol{a}^l = pool(\boldsymbol{a}^{l-1}) \\ FC : \boldsymbol{a}^l = \varphi(\boldsymbol{z}^l) = \varphi((\boldsymbol{w}^l)^T\boldsymbol{a}^{l-1} + \boldsymbol{b}^l) \end{cases} \tag{4}$$

$$\hat{\boldsymbol{w}}, \hat{\boldsymbol{b}} = \arg\max_{\boldsymbol{w},\boldsymbol{b}}(-E(\boldsymbol{a}^L)) \tag{5}$$

where $\boldsymbol{a}$ and $\boldsymbol{z}$ are the activation value and state value of neurons, respectively. The $*$ is the convolution operation and $\boldsymbol{\varphi}$ is the activation function. The number of network layers $l = 1, 2......, L$. $\boldsymbol{b}$ and $\boldsymbol{w}$ are the bias and weights of each layer, respectively. The *pool* is the pooling operation. In FC's formula, $\boldsymbol{a}$, $\boldsymbol{z}$, and $\boldsymbol{b} \in \mathbb{R}^{n_l \times 1}$, $\boldsymbol{w} \in \mathbb{R}^{n_{l-1} \times n_l}$. n is the number of neurons. In $Conv$'s formula, $\boldsymbol{b}$ $\in \mathbb{R}^{1\times 1}$, $\boldsymbol{a}$ and $\boldsymbol{z} \in \mathbb{R}^{H\times W}$. H and W are the height and width of input data, respectively. E is the error function. $\hat{(\cdot)}$ is the updated parameters.

3 Our Approach

3.1 Reduct of MLE-Based DPD and CNN-Based CV

There is a possible solution to the conflict between time consumption and accuracy of MLE-based DPD based on the independence of CNN training and testing phases. Only by FP of information in the testing phase can we move the time-consuming operation to the training phase of CNN and get the estimation position efficiently.

Compared with modelling $\boldsymbol{y}_i$ directly using fully connected networks, CNN can process the data more efficiently based on its two mentioned characteristics: sparse interaction and parameter sharing [2]. In addition, the ultimate goal of DPD is to find an estimation point such that the time and frequency delays generated at that point can compensate for the time and frequency delays of each $\boldsymbol{y}_i$. This is the process of aligning the time and frequency of each $\boldsymbol{y}_i$. While $\boldsymbol{y}_i$ is a complex signal, the alignment of frequencies requires complex numbers operations. Since the current neural networks are based on real numbers, it is impossible to operate on frequency information if $\boldsymbol{y}_i$ is directly used for modelling. Therefore, we transform $\boldsymbol{y}_i$ into a complex time-frequency signal and take its amplitude as the input to the network. In this way, the alignment of the time-frequency is done by moving the time-frequency diagrams of each $\boldsymbol{y}_i$ to align the similar parts. In this process, the time alignment is equivalent to the frequency alignment, and neither of them requires the use of complex number operations. Identifying similar parts of the image then relies on the translation invariance feature of the CNN [2].

The wavelet transform (WT) improves the idea of short-time- fourier-transform (STFT) localization while overcoming drawbacks: the window size does not vary with frequency. The telescopic translation operation, which can focus on arbitrary signal details, gradually refines the signal on multiple scales [9]. This characteristic is critical for the requirement of signal refinement feature extraction and estimation of localization parameters in passive positioning. As a result, we use WT to complete the two-dimensionalization.

Table 1. A Unified form of MLE-based DPD and CV regression problem. Let $\boldsymbol{\beta}_1 = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_N, \boldsymbol{v}_1, ..., \boldsymbol{v}_N, \boldsymbol{p}_1, ..., \boldsymbol{p}_N, f_s, f_c, C\}$, $\boldsymbol{\gamma}_1 = \{\boldsymbol{p}\}$, $\boldsymbol{\beta}_2 = \{|cwt(\boldsymbol{y}_1)|, ..., |cwt(\boldsymbol{y}_N)|, \boldsymbol{v}_1, ..., \boldsymbol{v}_N, \boldsymbol{p}_1, ..., \boldsymbol{p}_N, f_s, f_c, C\}$, and $\boldsymbol{\gamma}_2 = \{\boldsymbol{w}, \boldsymbol{b}\}$. $cwt(\cdot)$ is the operation of WT. $|\cdot|$ is the mode-taking operation. f_1, f_2 and f_3 are the nonlinear or linear functions. β and γ represent the receiver-related and receiver-independent parameters, respectively.

Method\Formula	$g_1 = f_1(\boldsymbol{\beta}, \boldsymbol{\gamma})$	$g_2 = f_2(g_1)$	$g_3 = \arg\max_{\beta}(g_2)$	$\hat{\boldsymbol{p}} = f_3(g_3, \boldsymbol{\beta})$
MLE-based DPD	$\bar{\boldsymbol{Q}}(\boldsymbol{\beta}_1, \gamma_1)$ $= \boldsymbol{V}^H \boldsymbol{V}$	$L(\boldsymbol{p}) = \lambda_{max}\{\bar{\boldsymbol{Q}}\}$	$\tilde{\boldsymbol{p}} = \arg\max_p (L(\boldsymbol{p}))$	$\hat{\boldsymbol{p}} = \tilde{\boldsymbol{p}}$
CNN-based DPD	$\boldsymbol{a}^L(\boldsymbol{\beta}_2, \gamma_2)$ $= \varphi((\boldsymbol{w}^L)^T \boldsymbol{a}^{L-1} + \boldsymbol{b}^L)$	$E(\boldsymbol{a}^L) = \sum_{i=1} \lvert\lvert \boldsymbol{a}^L - \boldsymbol{p}_i \rvert\rvert$	$\hat{\boldsymbol{w}}, \hat{\boldsymbol{b}} = \arg\max_{\boldsymbol{w}, \boldsymbol{b}}(-E(\boldsymbol{a}^L))$	$\hat{\boldsymbol{p}} = \boldsymbol{a}^L(\hat{\boldsymbol{w}}, \hat{\boldsymbol{b}}, \boldsymbol{\beta_2})$

We convert the formulas of MLE-based DPD and CNN-based DPD to the same form by rewriting, as shown in the Table 1. Accordingly, we can derive the calculation diagram of the unified form of them as Fig. 1. The reason for the inefficiency of MLE-based DPD is that each time the position estimation is performing, all lines in Fig. 1 need to be traversed. In contrast, the CNN-based DPD traverses along the solid line during training and only along the dashed line during the testing phase. If f_3 is simple enough, it can give the result quickly.

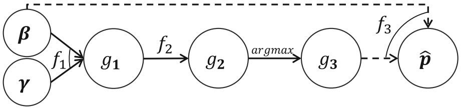


Fig. 1. Calculation diagram of unified form of MLE-based DPD and CNN-based DPD.

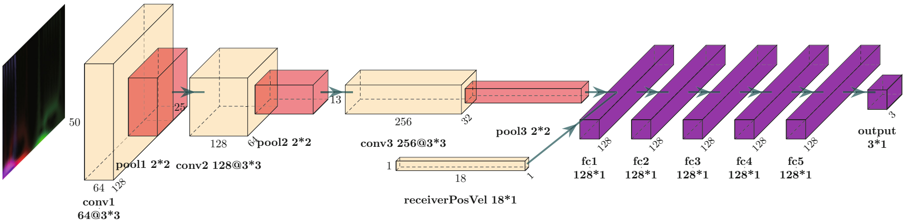


Fig. 2. The framework of CNNDPD. Three convolution layers (*conv*), three pooling layers (*pool*), and six fully connected layers (fc) make up the network. Both the *conv* and fc use the rectified linear unit (ReLU) as the activation function, except the *output* layer, which has no activation function.

3.2 CNNDPD

With the insight and analysis, we propose CNNDPD to verify the correctness of the scheme proposed in the third part of Sect. 2. The simplicity of the network becomes our key guiding direction in network building. The framework of CNNDPD is given in Fig. 2.

The input of the CNNDPD is $\boldsymbol{\beta}_2$, where we can find it contains many different types of data. How to input them into CNN simultaneously becomes a question worth considering. We handle it in the following way. $|cwt(\boldsymbol{y}_i)|$ is processed by convolutional operation because its data amount is large. We need an efficient way to handle it. Multiple $|cwt(\boldsymbol{y}_i)|$ are treated as different channels of the image. Since the amount of $\boldsymbol{v}_i$ and $\boldsymbol{p}_i$ data are small and are not the same type of data as $|cwt(\boldsymbol{y}_i)|$, they are treated as the second input of the network and modeled directly using fc. This is implemented by stitching the receivers' velocities and positions ($receiverPosVel$) with the data from the last pooling layer flattened as the input of fc. The process of $\boldsymbol{\beta}_2$ can be understood as a convolution operation to extract the TDOA and FDOA in $|cwt(\boldsymbol{y}_i)|$. Then fc used them with the $\boldsymbol{v}_i$ and $\boldsymbol{p}_i$ to build the equations and solve. We do not use f_c, f_s and C in CNNDPD because they are the same constants for all samples. Even though they are not fed into the network, it learns the corresponding values independently. If f_c and f_s are variables, they only need to be treated the $receiverPosVel$. The input data dimension is $3 \times 50 \times 128$, which is the time-frequency map of 3 receivers with a data length of 128 and 50 wavelet scale factors. The input data dimension is determined experimentally, as discussed in Sect. 5.

3.3 Data Pre-processing

To balance the effect of differing magnitudes on the model and enhance training efficiency, data normalization is necessary. Table 2 depicts the normalization processes.

Table 2. Normalization of the different magnitude data. max(·) is take maximum operation. The maximum value of the receiver's velocity, which is a constant, is V_{max}. A cube of $P_{max} \times P_{max} \times P_{max}$ is the positioning space. The value of V_{max} and P_{max} in this paper are 7554.6 and 1e6, respectively.

Name	Magnitudes	Normalization operation
WT data	1	$\lvert cwt(\boldsymbol{y}_i)\rvert \leftarrow \frac{\lvert cwt(\boldsymbol{y}_i)\rvert}{\max(\lvert cwt(\boldsymbol{y}_i)\rvert)}$
Speed of the receiver	m/s	$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i/V_{max}$
Location of the receiver	m	$\boldsymbol{p}_i \leftarrow \boldsymbol{p}_i/P_{max}$
Location of the emitter	m	$\boldsymbol{p} \leftarrow \boldsymbol{p}/P_{max}$

3.4 Training

The CNNDPD is trained with a total batch size of 50 for 500 epochs, and the number of samples is 1e5. The samples are divided into 4:1 into the training set and testing set. We use root mean squard error (RMSE) as the loss function, which defined by Eq. 6. $\hat{\boldsymbol{p}}_j$ and M are the estimated position of the emitter and the number of samples, respectively. The Adam [10] optimizer is used for updating the network's weights with a learning rate of 1e-4. The datasets of different SNR are trained separately. In practical situations, we can often calculate the SNR from the received signals and thus select the model closest to it to get a better result.

$$loss \triangleq \sqrt{\frac{1}{M}\sum_{j=1}^{M} ||\hat{\boldsymbol{p}}_j - \boldsymbol{p}||^2} \tag{6}$$

4 Simulation Results

4.1 Dataset

Since there is no publicly available dataset for passive positioning, we simulated it based on Eq. 2 as the following steps:

- Determine the signal $\boldsymbol{s}$, center frequency f_c, and speed C of propagation of the emitter.
- Determine the positions $\boldsymbol{p}_i$, velocities $\boldsymbol{v}_i$, sample rate f_s of the receivers.
- Randomly generate the position $\boldsymbol{p}$ and velocity $\boldsymbol{v}$ of the emitter at specified ranges.
- Generate Gaussian white noise $\boldsymbol{n}_i$ with specified signal-to-noise (SNR) defined as Eq. 7.
- Generate the receivers' signals $\boldsymbol{y}_i$ according to Eq. 2.

Finally, the WT of $\boldsymbol{y}_i$, velocities $\boldsymbol{v}_i$ and positions $\boldsymbol{p}_i$ of the receivers are saved as "data". The emitter's position $\boldsymbol{p}$ is saved as the "label" corresponding to the "data". The signal that we used is the Automatic Identification System (AIS) signal. The fc and fs are 161.975 MHz and 19.2 kHz, respectively. Three receivers are used for positioning. The $\boldsymbol{p_i}$ are $1e6{*}[0.5, 0.25, 0.6]^T m$, $1e6{*}[0.05, 0.95, 0.6]^T m$ and $1e6 * [0.95, 0.95, 0.6]^T m$, respectively. The $\boldsymbol{v_i}$ are all $[5500, 5500, 0]^T m/s$. The emitter is a random point in the $[1e6 \times 1e6]m$ region with a height of 0 and a random velocity of 0–30 m/s. Since the emitter height is 0, in the following experiments, all the methods only calculate the values of the emitter's other position coordinates. When conducting the experiments, for a given SNR (σ_i^y and σ are the standard deviation of $\boldsymbol{y}_i$ and $\boldsymbol{n}_i$, separately). 1e5 samples are generated for training and testing the network, and then 1e3 samples are generated for completing the monte carlo experiments.

$$SNR \triangleq 10 \log_{10} (\frac{\sigma_i^y}{\sigma})^2 \tag{7}$$

4.2 Experimental Settings

We use ESDPD [22] and GADPD [15] as the benchmarks to conduct comparison experiments. We have compared these methods in three aspects: Positioning Accuracy, Time Consumption and False Alarm Rate (FAR). The standard of Positioning Accuracy is the mean absolute error (MAE).

$$MAE \triangleq \frac{1}{M} \sum_{j=1}^{M} |\hat{\boldsymbol{p}}_j - \boldsymbol{p}| \tag{8}$$

Since it is not easy to distinguish between the poor positioning accuracy and a wrong positioning, we give a simple method to quantify the FAR defined as Eq. 9. We can intuitively find that the FAR is inversely correlated with the mean of the cumulative distribution function (CDF).

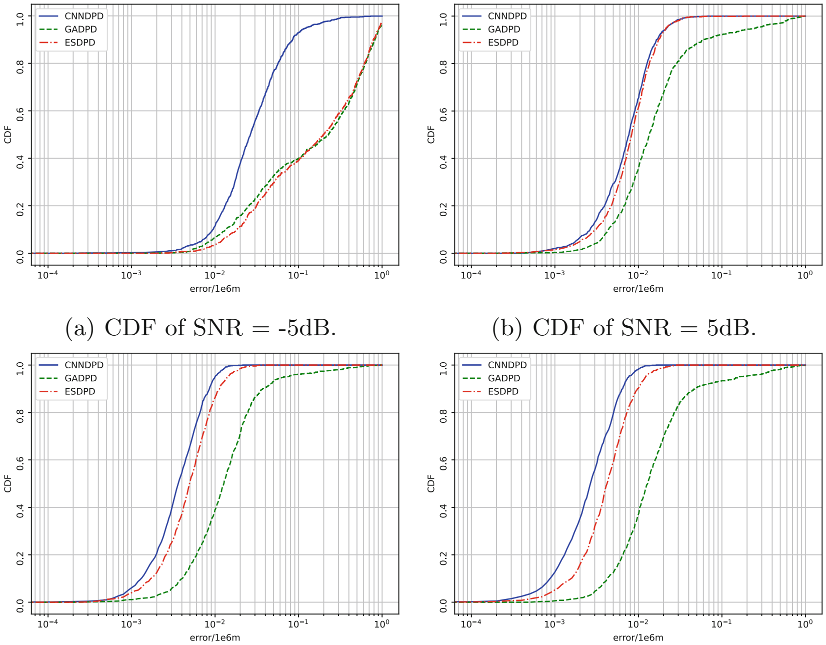
$$FAR \triangleq 1 - mean(CDF) \tag{9}$$

The detailed settings of each method are as follows. The grid size of ESDPD and the population individual coding accuracy of GADPD are set to be twice the

MAE of CNNDPD. The length of data K used for ESDPD and GADPD localization is set to 384. The K of CNNDPD is set to 128. The values that set above are discussed further in Sect. 5. The settings of [15] are partially referred to in the GADPD parameters. 500, 0.9, 0.1, and 0.1, respectively, are the population size, crossover probability, variation probability, and probability of introducing a completely new population. We limited the maximum iterations to 10 to guarantee running speed.

4.3 Result and Discussion

In terms of Positioning Accuracy (Fig. 4-a), CNNDPD performs better in all SNR conditions compared to the benchmarks. In terms of Time Consumption (Fig. 4-b), CNNDPD consumes time in the order of 1e-2s, which is better than the benchmarks. At SNR = 5 dB, the accuracy of ESDPD is comparable to CNNDPD, but the time consumption is 26 times longer. In terms of FAR (Fig. 3 and Table 3), at SNR = −5 dB, the FAR of the benchmarks are about 30% affected by discrete errors and local extremes. In a contrast, the FAR of CNNDPD is only 4.4%, which is lower than the benchmarks. Overall, the CNNDPD we proposed is more efficient than the benchmarks.



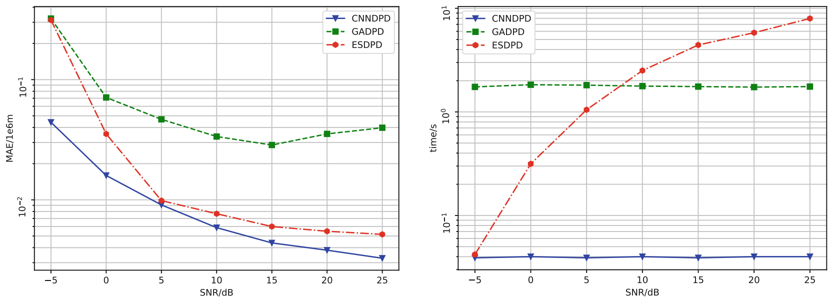
(a) CDF of SNR = -5dB.

(b) CDF of SNR = 5dB.

(c) CDF of SNR = 15dB.

(d) CDF of SNR = 25dB.

Fig. 3. Comparison of CDF.



(a) Positioning Accuracy.

(b) Time Consumption.

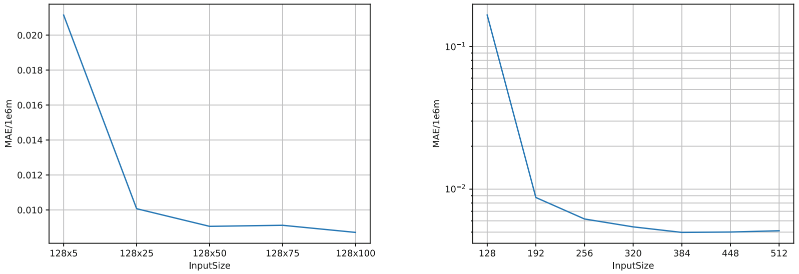
Fig. 4. Comparison of positioning accuracy and time consumption.

Table 3. Comparison of FAR.

Method	FAR			
	SNR = −5 dB	SNR = 5 dB	SNR = 15 dB	SNR = 25 dB
CNNDPD	**4.4%**	**0.91%**	**0.44%**	**0.33%**
GADPD	31.89%	4.86%	2.86%	3.96%
ESDPD	31.09%	0.99%	0.6%	0.52%

5 Ablation Study

In this section, we discuss the effect of input size on the positioning accuracy of CNNDPD, and the effect of data length on the positioning accuracy of ESDPD. The results are shown in Fig. 5.



(a) MAE of different number of WT scale factors (SNR=5dB).

(b) MAE of different data length to ESDPD (SNR=5dB, grid size=6Km).

Fig. 5. Ablation studies of WT scale factors and data length. We can find that the MAE of CNNDPD is decreasing as the number of WT scale factors increases. We choose “128×50” as the input to the network, weighing the accuracy and computation time of CNNDPD. The data length of “128” is chosen to ensure that it is larger than the number of data points corresponding to maximum τ_i of receivers and provides less computation time. For the ESDPD, when the data length is up to 384, the results become reliable.

6 Conclusion

This paper analyzes how MLE-based DPD and CNN-based DPD can be reduced in formulations and explores ways to achieve efficient DPD. We propose a CNN-based DPD method called a fast direct position determination with an embedded convolutional neural network to test the above hypothesis. Furthermore, we solve the problem of feeding multiple types of data into the CNN simultaneously in DPD and demonstrate that the end-to-end DPD approach can be implemented with CNN. CNNDPD can provide accurate and speedy position estimation under wide-area conditions, according to the experiments.

Acknowledgements. Supported by the National Natural Science Foundation of China (Grant No. 62022093 and No. 62101587).

References

1. Ballal, T., Bleakley, C.J.: Phase-difference ambiguity resolution for a single-frequency signal in the near-field using a receiver triplet. IEEE Trans. Signal Process. **58**(11), 5920–5926 (2010)
2. Bouvrie, J.: Notes on convolutional neural networks. Neural Nets (2006)
3. Carter, G.C.: Coherence and time delay estimation. Proc. IEEE **75**(2), 236–255 (2005)
4. Chen, X., Wang, D., Liu, Z.P., Wu, Y.: A fast direct position determination for multiple sources based on radial basis function neural network. In: 2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO) (2018)
5. Girshick, R.: Fast R-CNN. Comput. Sci. (2015)
6. He, K., Gkioxari, G., Dollr, P., Girshick, R.: Mask R-CNN. IEEE Trans. Pattern Anal. Mach. Intell. (2017)
7. Ho, K.C.: Bias reduction for an explicit solution of source localization using TDOA. IEEE Trans. Signal Process. **60**(5), 2101–2114 (2012)
8. Ho, K.C., Lu, X., Kovavisaruch, L.: Source localization using TDOA and FDOA measurements in the presence of receiver location errors: analysis and solution. IEEE Trans. Signal Process. **55**, 684–696 (2007)
9. Hu, G.S.: Modern Signal Processing Tutorial. Tsinghua University Press, Beijing (2004)
10. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. Comput. Sci. (2014)
11. Lecun, Y., Bottou, L.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
12. Lu, Z.Y., Wang, J.H., Wang, D.M., Wang, Y.: Fast algorithm for estimation of TDOA and FDOA based on genetic algorithm. Appl. Res. Comput. **33**(01), 178–180+188 (2016)
13. Pourhomayoun, M., Fowler, M.: Distributed computation for direct position determination emitter location. IEEE Trans. Aerosp. Electron. Syst. **50**(4), 2878–2889 (2014)
14. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Computer Vision and Pattern Recognition (2016)

15. Reng, Y.Q., Lu, Z.Y., Ba, B., Wang, D.M.: Fast direct position determination method based on the sharpening function genetic algorithm. J. XiDian Univ. **44**(04), 144–150 (2017)
16. Technicolor, T., Related, S.: Imagenet classification with deep convolutional neural networks [50] (2014)
17. Tzoreff, E., Weiss, A.J.: Expectation-maximization algorithm for direct position determination. Signal Process. (2017)
18. Ulman, R., Geraniotis, E.: Wideband TDOA/FDOA processing using summation of short-time CAF's. IEEE Trans. Signal Process. **47**(12), 3193–3200 (1999)
19. Vankayalapati, N., Kay, S., Ding, Q.: TDOA based direct positioning maximum likelihood estimator and the Cramer-Rao bound. IEEE Trans. Aerosp. Electron. Syst. **50**(3), 1616–1635 (2014)
20. Weiss, A.J.: Direct position determination of narrowband radio frequency transmitters. IEEE Signal Process. Lett. **11**(5), 513–516 (2004)
21. Weiss, A.J.: Direct geolocation of wideband emitters based on delay and doppler. IEEE Trans. Signal Process. **59**(6), 2513–2521 (2011)
22. Weiss, A.J., Amar, A.: Direct geolocation of stationary wideband radio signal based on time delays and doppler shifts. In: 2009 IEEE/SP 15th Workshop on Statistical Signal Processing (2009)
23. Wu, G.Z., Zhang, M., Guo, F.C.: High-resolution direct position determination based on eigenspace using a single moving ULA. Signal Image Video Process. **13**(5), 887–894 (2019)
24. Yang, J.-R.: Measurement of amplitude and phase differences between two RF signals by using signal power detection. IEEE Microwave Wirel. Compon. Lett. **24**(3), 206–208 (2014)
25. Zhang, S., Xing, M.: A novel doppler chirp rate and baseline estimation approach in the time domain based on weighted local maximum-likelihood for an MC-HRWS SAR system. IEEE Geosci. Remote Sens. Lett. **14**(3), 299–303 (2017)
26. Ziskind, I., Wax, M.: Maximum likelihood localization of multiple sources by alternating projection. IEEE Trans. Acoust. Speech Signal Process. **36**(10), 1553–1560 (1988)

GCD-Filter: Private Set Intersection Without Encryption

Mingli Wu and Tsz Hon Yuen(✉)

The University of Hong Kong, Pokfulam, Hong Kong
mingliwu@hku.hk, thyuen@cs.hku.hk

Abstract. A private set intersection (PSI) protocol is a protocol to get the intersection of two sets, each of which belongs to one party, without disclosing extra information of each party's set to the other party. In this paper, we propose a novel semi-honest PSI protocol without using any encryption primitive in a semi-honest security model. Specifically, we first slice all elements in the set and inject dummy slices. Then we utilize the greatest common divisor (GCD) to find the common parts of the two parties' slice set product, which is the product of all slices. Finally, we filter the elements by utilizing the GCD to find out the intersection. Different from most previous PSI protocols, we get the intersection by calculation rather than comparison.

Our protocol has many advantages over other state-of-the-art PSI protocols, such as robust security against quantum attacks, low communication cost, high computation efficiency when the bandwidth is low, etc. Through extensive experiments, we find the optimum parameters in our setting and demonstrate the performance of our protocol. Different from previous PSI protocols, the communication cost of our protocol varies with the cardinality of the intersection. In comparison, the communication cost of our protocol is the lowest, which gains an over 200% improvement than the communication-optimized PSI protocol *spot-low* (Pinkas et al., CRYPTO'19). In addition, the running time of our protocol is the lowest when the communication bandwidth is about 1 Mbps.

Keywords: PSI · GCD · Low communication costs

1 Introduction

Private set intersection (PSI) allows two parties P_1 and P_2, each of which holds a set X and Y respectively, to identify the intersection $X \cap Y$ without revealing any information about elements that are not in the intersection [15]. PSI is a special application of the secure multi-party computation (MPC) and has widespread applications, such as private contact discovery [3,4,7], secure genome analysis [1,16], P2P bots detection [10], and secure location-based services [11]. Therefore, the last decade has witnessed a lot of PSI protocols from both academia and industry.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 429–440, 2022.
https://doi.org/10.1007/978-3-031-19214-2_36

1.1 What Should We Value in PSI?

Literally, the goal of PSI is to get the "intersection" (i.e., the correctness), while the core of PSI is "private" (i.e., the security). In a semi-honest security model, both parties follow the protocol strictly but are curious to know extra information of the counterpart. In this paper, we focus on the semi-honest model. In addition, there are two important performance metrics: the computational cost (i.e., the running time) and communication cost (i.e., traffic data usage). Depending on different applications and scenarios, these two metrics are differently prioritized, hence different PSI protocols are designed. For example, in private contact discovery, the client owns a contact list with hundreds of contacts while the service provider holds a contact list with millions. It is not appropriate to return the large contact list being securely processed from the service provider to the user, otherwise the traffic usage of the user will be too large. Therefore, it needs to prioritize the communication cost for the user in this case.

1.2 Our Motivations

Although lots of semi-honest balanced PSI protocols have been proposed, the computation cost and communication cost are still high, especially the communication cost. One important reason is the overheads caused by using cryptographic primitives. Kolesnikov et al. [8] proposed the fastest PSI protocol called *BaRK-OPRF* by using batched oblivious pseudorandom function (OPRF). Even in comparison with the insecure naive hashing approach, *BaRK-OPRF* only showed ×4.3 slower. However, the communication cost is high. When the set size is $n = 2^{20}$, its communication cost is ×12.85 more expensive than the naive hashing. As early as 1986, Meadows [9] presented a DH-based PSI protocol, whose communication cost is only ×10.6 more expensive than the naive hashing. Though it holds competitive communication cost against most existing PSI protocols, the computation cost is high because of costly public-key operations. To the best of our knowledge, the semi-honest PSI protocol with the lowest communication cost is *spot-low* [12], whose communication complexity is specially optimized and only ×6.2 more expensive when the set size is $n = 2^{20}$. *Spot-low* exploits sparse oblivious transfer (OT) to make the achievement. In addition to *spot-low*, the authors in [12] also presented *spot-fast* that is computationally optimized.

In this paper, instead of relying on expensive encryption techniques, we utilize the elements themselves and inject dummy elements to achieve PSI. Therefore, the communication overheads caused by the expensive encryption can be saved.

1.3 An Overview of Our Protocol

In this paper, we design our protocol *GCD-Filter*, to achieve PSI. The general idea is to multiply all elements of each party's set (X and Y) to get two set products ($\mathit{\Pi}(X) = \prod_{x_i \in X} x_i$ and $\mathit{\Pi}(Y) = \prod_{y_i \in Y} y_i$). The GCD of these two set products will contain the common elements of the two sets. Then the GCD can be used to filter out common elements by testing divisibility between each element and the GCD. If an element can divide the GCD, we take it as a common element. Simple as it, there are two main challenges:

1. By directly utilizing the naked elements x_i to compute $\Pi(X)$, the first challenge is security. For example, if $\Pi(X) = 5 \times 8 \times 14 = 560$ and $gcd = 40$, then P_2 can easily find out that the non-common element in X is $540/40 = 14$. How do we get the common elements without disclosing non-common elements to the other party?
2. Since each element is also an integer that can be factorized, there may be false positives due to factor collision between different elements. For example, if $\Pi(X) = 5 \times 8 \times 14 = 560$ and $\Pi(Y) = 5 \times 13 \times 28 = 1820$, then $gcd = 140$. P_2 will think that the common factor is 5 and 28. How do we keep the false positives in an acceptable low rate?
3. Calculating the set product by multiplying all elements in a set is time-consuming, especially when the set size is large (e.g., 2^{24}). How do we compute the set product efficiently?

To address the first challenge, we introduce a pre-processing phase to split each element into several slices and fusing them into the set product such that the other party can hardly restore them. For example, a decimal integer $x_i = 1057$ can be written as a binary number 010000100001. We break it to three slices 0100||0010||0001 and represent them as decimal $(4, 2, 1)$. Then x_i is converted to $x'_i = 4 \times 2 \times 1 = 8$ in the pre-processing phase before computing the set product. This conversion to x'_i can partially hide the value of x_i. However, if the size of intersection is large and $x'_i = 8$ is the only non-common factor, then P_2 can deduce that x_i is one of the six combination of $(1, 2, 4), (1, 4, 2), (2, 1, 4)$, etc. To solve this problem, P_1 needs to inject dummy slices to ensure the security.

To deal with the second challenge, we have implemented extensive experiments to explore the optimum parameters (e.g., the number of slices), such that the false positives can be kept in a low rate while still keeping the protocol secure.

Since the time complexity of calculating a set product exponentially increases with the cardinality of the set. As for the third challenge, we take advantage of grouping to hash the elements of the large set into small groups, such that the processing scale can be reduced. Then we can implement our PSI protocol between each corresponding small groups of X and Y to get the group intersections. Finally, we concatenate these small group intersections to get the intersection of X and Y. In addition, we design a new algorithm to accelerate the multiplication.

1.4 Our Contributions

1. **New type of PSI without encryption.** We are the first one to achieve PSI without relying on any encryption primitives (e.g., AES). The security of both parties are ensured by the distribution of the elements themselves and the dummy elements. Our scheme does not rely on the security of symmetric encryption or based on some intractability assumptions.

2. **Robust security.** Even if a malicious P_2 runs a brute-force attack by checking every element in the domain with P_1, he/she can only get a candidate set that is a superset of the intersection. The cardinality of the candidate set is much larger than that of the real set, which means P_2 can get very limited information of P_1. Note that none of the previous works in PSI can be resistant to the brute-force attack. On the other hand, since our protocol does not rely on the security of any encryption primitives, even being quantum-attacked, it is still robust.
3. **Low communication cost.** The communication cost of our protocol is the lowest compared with the state-of-the-art PSI protocols. When the set cardinality is $n = 2^{16}$, our protocol costs 1.35–1.94 MB, while *spot-low* needs 3.9 MB, which is about ×2.01–2.89 more expensive than ours. Also, the communication cost of our protocol increases with the size of intersection. Therefore, our protocol can overcome the deficiency in previous PSI protocols that all encoded elements have to be transferred even there are a few elements in common in both sets.
4. **High running-time efficiency under low bandwidth.** On conditions that the bandwidth is low, our protocol can show its running-time advantages over other state-of-the-art PSI protocols. When the bandwidth is 1 Mbps and the set size is $n = 2^{20}$, our protocol only consumes 274.7 s when the intersection is empty, which is about ×2.10 faster than the computation optimized *spot-fast* [12] that costs 576.2 s.

2 Related Work

In this section, we survey the related PSI protocols and sort them based on the techniques they exploit.

Hashing. The intuitive and simplest approach to achieve PSI is by directly using the hash functions, which is called naive hashing. In naive hashing, both parties just need to map their elements by using a common secure hash function. Then one party can share the hash images to get the common elements. Though it achieves great performance with both low computational cost and communication cost, it is insecure [13–15]. Among other hashing techniques, cuckoo hashing [2,7,8,14] is the most popular one because of its ability to build dense hash tables with high search efficiency. Pinkas et al. [12] proposed a new 2-choice hashing technique and claimed there is almost no dummy elements.

OT-Based. Many PSI protocols [5–8,13–15] took advantages of Oblivious Transfer (OT), which achieved the highest computational efficiency. Dong et al. [5] combined OT, secret sharing, and bloom filter to design a PSI protocol. In their work, they claimed their protocol was fastest than two previously fastest protocols. Pinkas et al. [13] claimed their previous OT extension based work [14] was the fastest one, and improved it by using permutation-based hash functions. Their further work [7] paid their attention to strike a balance between computation and communication using sparse OT extension. Their latest work [6] continued to extend their security model from the semi-honest one to the malicious one by introducing a OT-hybrid model and kept linear communication. In this work, they also discussed the cases of malicious sender's set size and malicious receiver's set size. Kolesnikov et al. [8] presented the ever-known fastest PSI protocol by using OPRF, which was able to implement batched processes.

Public Key-Based. The earliest known PSI protocol is based on Diffie-Hellman key exchange (DH). In 1986, Meadows [9] proposed a DH-based PSI protocol. The key idea of DH approach is to compare $(h(x)^{k_1})^{k_2}$ with $(h(y)^{k_2})^{k_1}$, where k_1 and k_2 are the private keys each party holds. Though DH-based PSI protocols can keep a low communication costs, their computational costs are high due to the expensive public key operation. Cristofaro and Tsudik [3] utilzed blind-RSA signatures to construct a PSI protocol that could scale linearly in computation with the set size.

3 Preliminaries

3.1 Unique Integer Factorization Theorem

According to the unique integer factorization theorem, any integer N can be factored into a unique product of powers with prime bases:

$$N = {p_1}^{c(p_1)} {p_2}^{c(p_2)} \cdots {p_\nu}^{c(p_\nu)} = \varPi_{i=1}^{\nu} {p_i}^{c(p_i)},$$

where $p_1 < p_2 < \cdots < p_\nu$ are primes, and $c(p_i)$ are the number of p_i contained in N, and ν is the number of primes.

3.2 *k*-Ordered Factorization

Factorization or integer factorization is a process of writing a number as a product of its factors. For example, 12 can be written as $2 \times 2 \times 3$. Here we also call $2 \times 2 \times 3$ is a factorization of 12, which can be represented as a 3-tuple $(2, 2, 3)$. Ordered factorization takes $(2, 2, 3)$, $(2, 3, 2)$, $(3, 2, 2)$ as different ones and k is the number of factors of a factorization. In this paper, a k-ordered factorization can correspond to an element.

4 Our Protocol

In this section, we will introduce our protocol step by step. Firstly, each element in the set is preprocessed by splitting and multiplication to provide security. Then we introduce the GCD-Naive algorithm, which is the core part for filtering elements using *gcd*. Finally, we give the complete GCD-Filter algorithm, which divides the set into smaller sub-groups for running the GCD-Naive algorithm, to improve the overall performance.

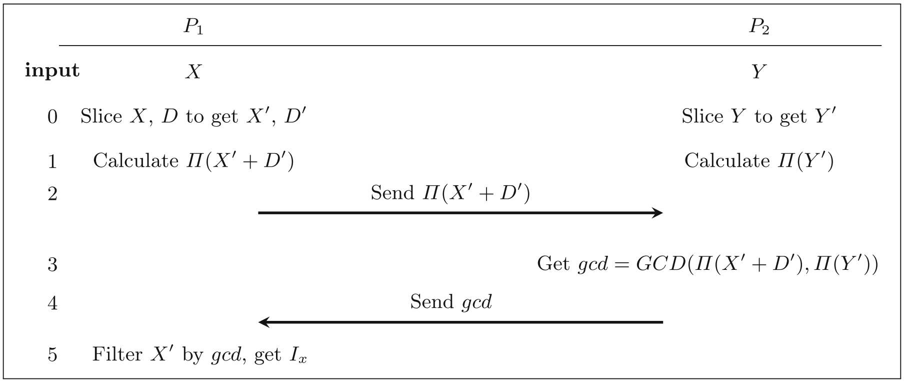

P_1
P_2
input
X
Y
0 Slice X, D to get X', D'
Slice Y to get Y'
1 Calculate $\Pi(X' + D')$
Calculate $\Pi(Y')$
2 Send $\Pi(X' + D')$
3 Get $gcd = GCD(\Pi(X' + D'), \Pi(Y'))$
4 Send gcd
5 Filter X' by gcd, get I_x

Fig. 1. GCD-Naive: get the intersection by directly sending a large set product

4.1 Elements Preprocessing

In reality, the elements in the two sets can be strings or other data types (e.g., email addresses, account IDs) and the length may vary. Transforming these elements into elements from a uniform domain is a common way to protect the privacy of the users. In this paper, we assume that the elements are uniform in $\{0, 1, \cdots, 2^\sigma\}$, where σ is the each element's bit length. Then we start to preprocess them. For each element s_i in a set S, there are two steps:

1. Slice s_i into a k-tuple $(s_{i,1}, s_{i,2}, \cdots, s_{i,k})$, in which the slices $s_{i,j}$ $(j = 1, 2, \cdots, k)$ are with equal bit length σ/k. For example, when $\sigma = 12$ and $k = 3$, by slicing $s_i = 1057_{10} = 010000100001_2$, we can get a triple $(0100_2, 0010_2, 0001_2) = (4_{10}, 2_{10}, 1_{10})$; to put it simple, we can slice $s_i = 1057$ into a triple $(4, 2, 1)$.
2. Then for each s_i, we multiply all its slices in its corresponding tuple to get a new element s'_i, i.e., $s'_i = \prod_{j=1}^{k} s_{i,j}$.

In the first step, it is noted that we have $s_i = \sum_{j=1}^{k} s_{i,j} 2^{(k-i)\sigma/k}$. Then it is obvious that each s_i can be uniquely represented by a k-tuple; and vice versa. In the second step, since s'_i is the product for all s_i's slices, we call it *slice product* of s_i. After preprocessing for each element s_i, we can get a new set S' from S. It is noted that the S' can be a multiset due to collision where $s'_i = s'_j, i \neq j$. For simplicity, we still call S' as a set. Now do the same processing to X and Y, we can get X' and Y', respectively.

4.2 GCD-Naive: Filtering Elements by Using *gcd*

Figure 1 shows the general protocol for PSI by sending a large number to P_2 from P_1. In this protocol, step 0 implements the elements preprocessing. In step 0, X' and Y' are the slice set of X and Y, respectively. D is the dummy set and D' is its slice set. We can also use X' to distinguish elements in the intersection and elements not in the intersection, which corresponds to step 5. It is noted

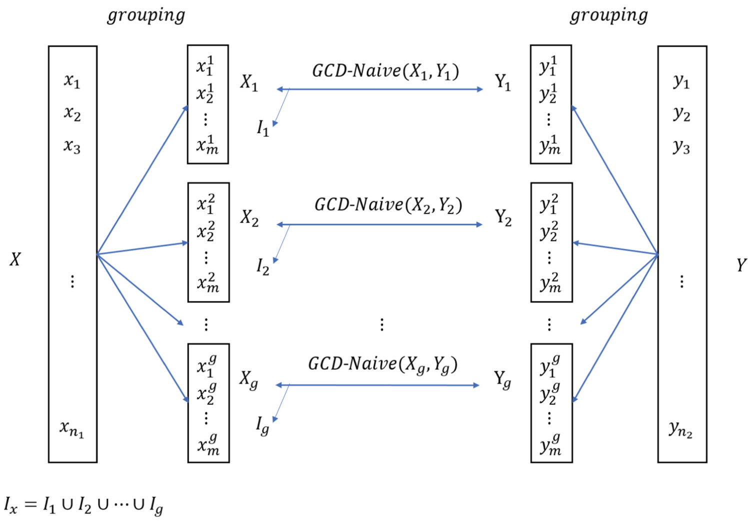

grouping
grouping
x_1
x_2
x_3
⋮
x_{n_1}
X
x_1^1 x_2^1 ⋮ x_m^1
X_1
I_1
$GCD\text{-}Naive(X_1, Y_1)$
Y_1
y_1^1 y_2^1 ⋮ y_m^1
x_1^2 x_2^2 ⋮ x_m^2
X_2
I_2
$GCD\text{-}Naive(X_2, Y_2)$
Y_2
y_1^2 y_2^2 ⋮ y_m^2
x_1^g x_2^g ⋮ x_m^g
X_g
I_g
$GCD\text{-}Naive(X_g, Y_g)$
Y_g
y_1^g y_2^g ⋮ y_m^g
y_1
y_2
y_3
⋮
y_{n_2}
Y
$I_x = I_1 \cup I_2 \cup \cdots \cup I_g$

Fig. 2. The diagram of GCD-Filter

that after step 0, every element in both X and Y is written as an equal k-tuple. After step 0, all slices of x_i are fused into its slice product x'. Furthermore, after step 1, all slices of x_i will be fused into the slice set product $\Pi(X' + D')$. In step 5, P_1 filters each of its element x in group X by dividing its slice product x' with *gcd*. If x' can divide *gcd*, then put x in the intersection I; otherwise not. It is easy to know the correctness of GCD-Naive. If x is in I, its corresponding set product must divide *gcd*; if x's slice product cannot divide *gcd*, x must not be in I. In terms of security, it is difficult for an attacker to extract a k-ordered factorization that corresponds to an element from the large set product.

4.3 GCD-Filter: Grouping to Reduce the Processing Magnitude

The problem of GCD-Naive is the high computation cost when the set size is large. In step 1, both parties need to calculate a slice set product, which is the product of the slices of all elements in the set. When the set size is large, the multiplication process will become expensive. Theoretically, the computation complexity of getting the set product is in exponential (i.e., $O(2^n)$). Though our main focus in this paper is reducing the communication cost, we still need to consider the computation cost to guarantee the protocol is practical and applicable.

Grouping is a strategy that can reduce the processing magnitude. The most popular grouping strategy in existing protocols is based on cuckoo hashing because of its great ability to build a dense hash table. After grouping by cuckoo hashing, each item will be mapped its corresponding bin, which has a predefined size to guarantee security.

In this paper, since our elements are already uniform, we can simply exploit the element itself for grouping. Specifically, we truncate each element and reserve the last $\lfloor \log g \rfloor$ bits for grouping, where $g = \lfloor \frac{n}{m} \rfloor$ is the number of groups and m is the group size. Each element will be mapped to its corresponding group based on its $\lfloor \log g \rfloor$ bits. After grouping, we can turn the computation complexity from exponential $O(2^n)$ to approximate linear $O(g2^m) = O(\lfloor \frac{n}{m} \rfloor 2^m)$. We call it "approximate linear" because in each group, the computation complexity is still exponential $O(2^m)$; but the between groups, the computation complexity is $O(g)$. One more benefit brought by grouping is parallel processing. We can execute parallel processing between groups to further accelerate the computation.

Let us denote X_i (resp., Y_i) as the ith group after grouping X (resp., Y), I_i as the intersection got by executing GCD-Naive between X_i and Y_i, where $i = 1, 2, \cdots, g$. By utilizing grouping, we show the general diagram of our new protocol, GCD-Filter, in Fig. 2.

5 Performance Evaluation

We implement our experiments by running the open-sourced codes of the authors. To enable big integer calculation, we use NTL library 11.4.3 with GMP library 6.2.0. In the balanced setting (i.e., $n_1 = n_2$), we test set size $n \in \{2^{16}, 2^{20}, 2^{24}\}$. Because the real bandwidth is not stable, for fair and reliable comparisons, we have our implementations in local network as [12] and utilize *wondershaper*[1] to control the bandwidth. The bandwidth ranges from 1 Mbps, 10 Mbps, 100 Mbps to 1 Gbps. All experiments are done in a laptop with Intel(R) Core(TM) i5-4210H CPU @ 2.90 GHz and 8 GB RAM.

5.1 Asymptotic Communication Complexity Comparison

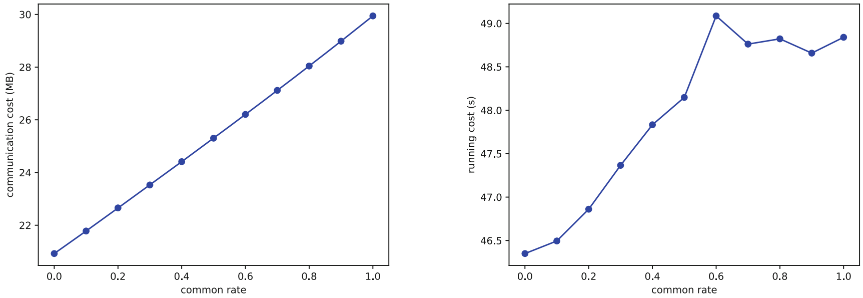
The exact communication costs of GCD-Filter include the slice group products (i.e., $\Pi(X_i' + D_i')$, $i = 1, 2, \cdots, g$) and the slice group product GCDs (i.e., gcd_i, $i = 1, 2, \cdots, g$). For every group i, since gcd_i can divide $\Pi(X_i' + D_i')$, the real communication cost is less than $2\psi(\Pi(X_i' + D_i'))$, where $\psi()$ is the bit length counting function. Then for all groups, the communication cost is less than $2\psi(\Pi(X_i' + D_i'))g$. Denote $L(n)$ as the expected number of bits of the set product for set with size n, theoretically, we can write the communication complexity as $2L(m+d)g = 2L(m+d)\lfloor n_1/m \rfloor$, where d is the number of dummy elements. We analyze $L(m+d)$ and find $L(2^8 + 2^6)$ can be set as 39113. We compare the asymptotic communication complexity of different PSI protocols in Table 1. From this table, we can see our protocol gains obvious advantages in communication complexity. Also, the communication cost does not rely on n_2. Therefore, even in the unbalanced setting that $n_2 \gg n_1$, our protocol can still keep low communication cost, while the other listed protocols cannot. To transfer a large set product online, we can also utilize the unique integer factorization theorem to compress the bit length. Specifically, one can transfer the

[1] https://github.com/magnific0/wondershaper.

Table 1. Theoretical communication cost comparison

Protocol	Communication (bits)	$n = n_1 = n_2$		
		2^{16}	2^{20}	2^{24}
Naive hashing	$n_1(\log n_1 + \log n_2 + \lambda)$	$72n$	$80n$	$88n$
DH-ECC [9]	$(n_1 + n_2)\phi + n_1(\log n_1 + \log n_2 + \lambda)$	$640n$	$648n$	$656n$
BaRK-OPRF [8]	$(3 + s)(\lambda + \log(n_1 n_2))n_1 + 1.2\ell n_2$	$1042n$	$1018n$	$978n$
spot-low [12]	$1.02(\lambda + \log(n_2) + 2)n_1 + \ell n_2$	$488n$	$500n$	$512n$
spot-fast [12]	$2(\lambda + \log(n_1 n_2))n_1 + \ell(1 + 1/\lambda)n_2$	$583n$	$609n$	$634n$
GCD-Filter	$\leq 2L(m + d)n_1/m$	$\leq 306n$	$\leq 306n$	$\leq 306n$

The statistic security parameter is $\lambda = 40$; $\phi = 284$ is elliptic curve size; ℓ is the width of OT extension matrix can be found in [6]; s is the maximum size of the stash for cuckoo hashing and can be found in [8].



(a) Communication cost

(b) Running cost

Fig. 3. The communication cost (MB) and running cost (s) with different common rates when $n_1 = n_2 = 2^{20}$.

power $c(p_1), c(p_2), \cdots, c(p_t)$ of the prime base $p_1, p_2, \cdots, p_t$. In our experiments, we find we can save $\geq 25\%$ communication costs when setting $t \geq 90$.

5.2 Experimental Evaluation

Table 2 mainly shows the performance comparison between different PSI protocols. In this table, the naive hashing is listed for reference.

Communication. During the interaction of GCD-Filter, only two large numbers need to be transferred: a set product, and a gcd. Since the bit length of the gcd depends on the number of common elements of both sets, the communication cost varies. We illustrate the communication cost with different common rates when $n_1 = n_2 = 2^{20}$ in Fig. 3a. As we can see from this figure, the communication cost increases linearly with the increase of the common rate. For better evaluation of GCD-Filter, we implement two cases in different settings based on the common rate of X and Y in Table 2. When the common rate is 0.0

Table 2. Communication cost in MB and running time in second for different PSI protocols with single thread.

Cardinality	Protocol	Commu. (MB)	Running time (s)			
$n_1 = n_2$			1 Gbps	100 Mbps	10 Mbps	1 Mbps
2^{24}	naive hashing	176.0	16.8	43.6	316.4	3054.2
	DH-ECC [9]	–	–	–	–	–
	BaRK-OPRF [8]	2136.7	**182.9**	**227.1**	1924.2	–
	spot-low [12]	–	–	–	–	–
	spot-fast [12]	1095.4	505.155	515.4	**967.2**	–
	GCD-Filter(0.0)	**324.7**	860.5	770.7	1263.7	**4310.0**
	GCD-Filter(1.0)	**461.1**	914.5	820.3	1696.3	**5342.4**
2^{20}	naive hashing	10.0	0.9	2.4	18.1	174.1
	DH-ECC [9]	106.0	989.6	988.2	999.6	1098.7
	BaRK-OPRF [8]	128.5	**2.1**	**11.6**	111.4	1112.1
	spot-low [12]	–	–	–	–	–
	spot-fast [12]	66.5	26.6	27.2	**58.4**	576.2
	GCD-Filter(0.0)	**20.9**	56.1	58.3	87.1	**274.7**
	GCD-Filter(1.0)	**29.9**	62.2	63.7	115.5	**343.0**
2^{16}	naive hashing	0.56	0.05	0.13	0.96	10.0
	DH-ECC [9]	6.6	61.0	63.4	67.2	118.7
	BaRK-OPRF [8]	7.73	**0.14**	**0.72**	6.7	68
	spot-low [12]	3.9	11.5	11.6	13.2	41.2
	spot-fast [12]	4.0	2.02	2.06	**3.6**	36.0
	GCD-Filter(0.0)	**1.35**	3.60	3.75	5.65	**17.40**
	GCD-Filter(1.0)	**1.94**	3.86	4.00	7.45	**21.91**

"-" indicates the corresponding execution runs out of memory or takes too long to run. The best results except naive hashing for each class are marked in bold.

(i.e., there is no common element), GCD-Filter can achieve its lowest communication cost. In contrast, when the common rate is 1.0 (i.e., all elements in X are common elements), the communication cost of GCD-Filter is highest. As is shown from Table 2, the communication cost of our protocol is the lowest compared with other protocols except naive hashing. Compared with the traditional DH-based protocol that is known for low communication cost, our protocol is over 3 times cheaper. Even for *spot-low* that is specially designed to achieve low communication cost, our protocol is also much cheaper. When the set size is 2^{16}, our protocol only needs 1.94 MB when the common rate is 1.0, which is an approximate 2 times improvement compared with 3.9 MB of *spot-low*; when the common rate is 0.0, this improvement becomes more obvious. When setting the common rate ≥ 0.2, the false positive rate ≤ 0.0003.

Running Cost. We also illustrate the running cost with different common rates in Fig. 3b. As we can see from this figure, the running cost shows a general increase trend with the increase of the common rate. The running cost includes two parts: offline time and online time. Generally, when the bandwidth is high, the offline time will dominate the running time; otherwise, the online time will dominate the running time. As is shown in Table 2, when the bandwidth is no less than 100 Mbps, BaRK-OPRF [8] outperforms other protocols; when the bandwidth is 10 Mbps, spot-fast [12] is fastest when the set size is over 2^{16}. Since our protocol has great communication saving, the online time is less than other protocols. When the bandwidth is low, this advantage becomes significant. As is shown in this table, when the bandwidth is 1 Mbps, our protocol outperforms all other secure protocols. For $n_1 = n_2 = 2^{20}$, our protocol needs 343.0 s when the common rate is 1.0, which is about 1.68 times cheaper than spot-fast that needs 576.2 s. It is noted that spot-fast has been computationally optimized in [12].

6 Conclusion

In this paper, we propose GCD-Filter, a novel PSI protocol without relying on any encryption primitives. Specifically, we first inject dummy elements, and slice each element. Then the first party can send the large product of all slices to the other party to get the GCD, which can be used to filter the common elements. We demonstrate that GCD-Filter achieves the lowest communication complexity compared with the-state-of-art PSI protocols. In addition to the low communication cost, GCD-Filter also takes the least running time when the bandwidth is low. In the future, we will explore more possibilities to improve both the computational cost and communication cost.

Acknowledgements. This research is supported by the grant from Hong Kong RGC Early Career Scheme (Project No. 27208621) and HKU Seed Fund for Basic Research (Project No. 202111159151).

References

1. Baldi, P., Baronio, R., Cristofaro, E.D., Gasti, P., Tsudik, G.: Countering GATTACA: efficient and secure testing of fully-sequenced human genomes. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, Chicago, Illinois, USA, 17–21 October 2011, pp. 691–702 (2011)
2. Chen, H., Laine, K., Rindal, P.: Fast private set intersection from homomorphic encryption. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, 30 October–03 November 2017, pp. 1243–1255 (2017)
3. Cristofaro, E.D., Manulis, M., Poettering, B.: Private discovery of common social contacts. Int. J. Inf. Sec. **12**(1), 49–65 (2013)
4. Demmler, D., Rindal, P., Rosulek, M., Trieu, N.: PIR-PSI: scaling private contact discovery. PoPETs **2018**(4), 159–178 (2018)

5. Dong, C., Chen, L., Wen, Z.: When private set intersection meets big data: an efficient and scalable protocol. In: 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS 2013, Berlin, Germany, 4–8 November 2013, pp. 789–800 (2013)
6. Howard, H., Mortier, R.: Paxos vs raft: have we reached consensus on distributed consensus? In: Fekete, A., Kleppmann, M. (eds.) 7th Workshop on Principles and Practice of Consistency for Distributed Data, PaPoC@EuroSys 2020, Heraklion, Greece, 27 April 2020, pp. 8:1–8:9. ACM (2020)
7. Kales, D., Rechberger, C., Schneider, T., Senker, M., Weinert, C.: Mobile private contact discovery at scale. In: 28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, 14–16 August 2019, pp. 1447–1464 (2019)
8. Kolesnikov, V., Kumaresan, R., Rosulek, M., Trieu, N.: Efficient batched oblivious PRF with applications to private set intersection. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016, pp. 818–829. ACM (2016)
9. Meadows, C.: A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party. In: 1986 IEEE Symposium on Security and Privacy, pp. 134–134. IEEE (1986)
10. Nagaraja, S., Mittal, P., Hong, C., Caesar, M., Borisov, N.: BotGrep: finding P2P bots with structured graph analysis. In: Proceedings of 19th USENIX Security Symposium, Washington, DC, USA, 11–13 August 2010, pp. 95–110 (2010)
11. Narayanan, A., Thiagarajan, N., Lakhani, M., Hamburg, M., Boneh, D.: Location privacy via private proximity testing. In: Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, USA, 6–9 February 2011 (2011)
12. Pinkas, B., Rosulek, M., Trieu, N., Yanai, A.: SpOT-light: lightweight private set intersection from sparse OT extension. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11694, pp. 401–431. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26954-8_13
13. Pinkas, B., Schneider, T., Segev, G., Zohner, M.: Phasing: private set intersection using permutation-based hashing. In: 24th USENIX Security Symposium, USENIX Security 2015, Washington, DC, USA, 12–14 August 2015, pp. 515–530 (2015)
14. Pinkas, B., Schneider, T., Zohner, M.: Faster private set intersection based on OT extension. In: Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, 20–22 August 2014, pp. 797–812 (2014)
15. Pinkas, B., Schneider, T., Zohner, M.: Scalable private set intersection based on OT extension. ACM Trans. Priv. Secur. **21**(2), 7:1–7:35 (2018)
16. Wang, X.S., Liu, C., Nayak, K., Huang, Y., Shi, E.: iDASH secure genome analysis competition using ObliVM. IACR Cryptology ePrint Archive 2015/191 (2015)

Incorporating Self Attention Mechanism into Semantic Segmentation for Lane Detection

Genji Yuan[1], Jianbo Li[1,2](✉), Yue Wang[1], and Xianglong Meng[1]

[1] College of Computer Science and Technology, Qingdao University,
Qingdao 266071, China
lijianbo@qdu.edu.cn

[2] Institute of Ubiquitous Networks and Urban Computing, Qingdao 266070, China

Abstract. Lane detection is a challenging task in the field of vision detection. The annotation information of lane is very sparse, and it is faced with the interference of occlusion, illumination and other factors, which seriously affects the capture of lane features by neural network. In this paper, we propose the Self-Attention Lane Segmentation Network (SALSN) which allows attention-driven, long-range dependency modeling for lane detection task. Although traditional convolutional neural networks have demonstrated their powerful performance, their ability to capture global relationships in images has not been fully explored. We introduce a self-attentive module to model the long-range dependencies between lane features. Lanes have strong shape constraints but weak coherence. In SALSN, we utilize a dense feature fusion framework to better capture lane context information and use all element information to generate lane segmentation images. Experimental results show that SALSN is not only effective in learning the remote dependencies of lane features, but also significantly improves the lane detection performance. We have validated our approach on two large-scale lane detection datasets, and our method can achieve more competitive results.

Keywords: Lane detection · Self-attention · Long-range dependency · Multi-stage dependencies

1 Introduction

Lane detection constitutes one of the foundations of automatic driving [1,2] and becomes more challenging due to the diversity of scenarios (different lighting

This research was supported in part by National Key Research and Development Plan Key Special Projects under Grant No. 2018YFB2100303, Shandong Province colleges and universities youth innovation technology plan innovation team project under Grant No. 2020KJN011, Shandong Provincial Natural Science Foundation under Grant No. ZR2020MF060, Program for Innovative Postdoctoral Talents in Shandong Province under Grant No. 40618030001, National Natural Science Foundation of China under Grant No. 61802216, and Postdoctoral Science Foundation of China under Grant No. 2018M642613.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 441–449, 2022.
https://doi.org/10.1007/978-3-031-19214-2_37

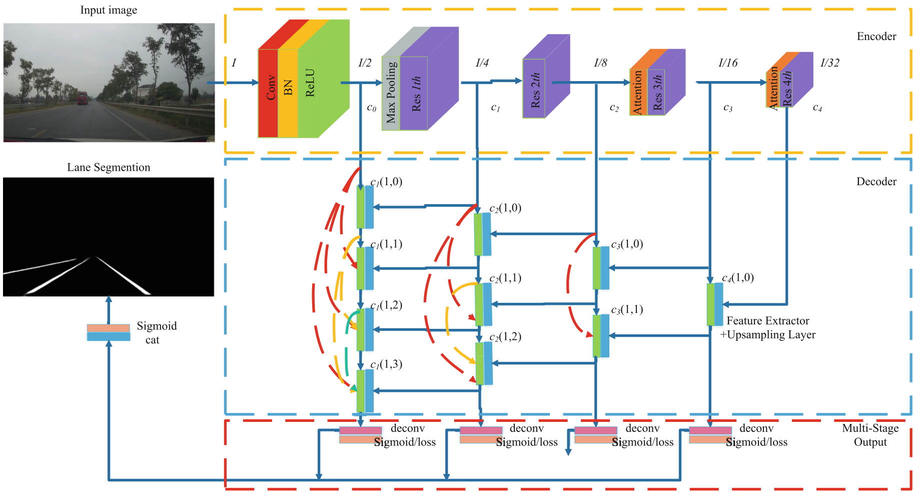


Fig. 1. The network architecture of lane detection in this paper. It consists of a self-attention module, an encoder and a dense skip connection decoder. I represents the resolution of the input image. c_k represents the number of feature map channels at different levels. $c_k(i, j)$ indicates feature extraction (green) and upsampling (blue). At each scale, the pixel-wise prediction loss is calculated independently. Meanwhile, the fused maps at all scales are concatenated and fused to product a multi-scale fusion map. (Color figure online)

conditions, occlusion, and road markings). Traditional lane detection methods [3] rely on hand-crafted features and heuristics to identify lanes. However, the heuristic approach has high computational complexity and is not robust enough for changes in road scenarios. Recently, advanced lane detection methods [4,5] have focused on using deep networks instead of hand-crafted features to achieve dense detection, treating lane detection as a semantic segmentation task.

The methods based on end-to-end [6,7] lane detection can classify the pixels in the image pixel by pixel, and the precise localization makes the accuracy rate further improved. Zheng et al. [8] proposed a module named REcurrent Feature-Shift Aggregator (RESA) to enrich lane features based on preliminary feature extraction using CNN, and also proposed a Bilateral Up-sampling Decoder that can combines coarse-grained and fine-detail features in the upsampling stage, and finally restores the low-resolution feature maps to pixel-wise predictions. Liu et al. [9] proposed a top-to-down lane detection framework, named CondLaneNet, which first detects lane instances and then dynamically predicts the line shape of each instance. They also introduced a conditional lane detection strategy based on conditional convolution and row-wise formulation, while designing a Recurrent Instance Module (RIM) to overcome the problem of detecting lane lines with complex topology. Tabelini et al. [10] proposed an anchor-based deep lane detection model named LaneATT, which is similar to other general-purpose deep object detectors, using anchors for a feature pooling step, while proposing an aggregation Anchor-based attention mechanism for global information.

In this paper, we present a simple and effective way to obtain global information. We utilise the self-attention module to obtain long-range, multi-level dependencies of the lane. The self-attention mechanisms [11] exhibit a good balance between the ability to construct long-range dependencies and computational efficiency. The self-attention mechanism has less model complexity and fewer parameters than the convolutional structure. The model set out in this paper has better robustness for long continuous structure objects such as lanes with strong spatial relationships but fewer feature cues. The main contributions of this paper are as follows:

(1) In this paper, we propose a self-attention lane segmentation network. We complement the convolutional network with a self-attention module that enables the network to capture both global and local information to enhance the learning lane features ability of the convolutional neural network.
(2) We implement multi-scale feature aggregation by densely-connected skip connections. Feature loss can be effectively avoided by combining global contextual information.
(3) We consider that lane detection is highly dependent on sparse lane masks as supervision. Therefore, we exploit lane similarity loss to incorporate lane geometric features into the network and detect lanes using the priori information of lane.

2 Proposed Method

Lane detection is often regarded as a semantic segmentation task. Specifically, we assign a label I_{ij} ($I_{ij} = 1, ..., N_c$) to each pixel of the input image X, where N_c represents the class of segmentation. The goal of the deep network is to learn the mapping ($F : X \mapsto s$) of the input image X to the segmentation map s. The proposed network in this paper not only efficiently captures the long-range dependence between lane pixels, but also flexibly aggregates features between different scales. We call the proposed method Self-Attention Lane Segmentation Networks (SALSN) because of its self-attention module.

2.1 Network Structure

We apply DenseNet [12] to the network architecture of this paper. The proposed SALSN can achieve flexible feature fusion in the decoder. We fuse the outputs of each convolution stage to generate multi-scale and multi-level feature maps.

As shown in Fig. 1, the proposed SALSN extracts lane information from images based on the encoder-decoder architecture. We designed the backbone structure of the encoder based on ResNet [13]. First, the encoder incorporating the self-attention mechanism extracts the feature map and attention map from the image. Then, flexible feature fusion is implemented in the decoder. To take full advantage of the different scales and levels of features, we fuse the feature

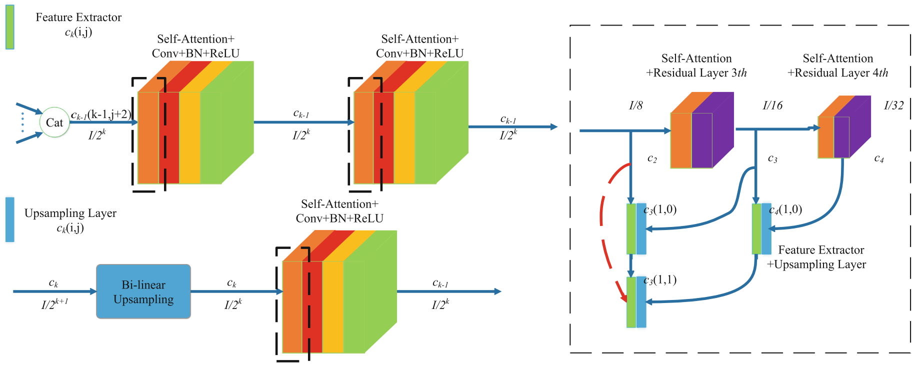


Fig. 2. The decoder consists of two different types of modules: feature extractors and upsampling layers. *Cat* means cascade. The attention module marked by the dashed frame is only applied in the top two layers of the decoder, that is, when $I/32$ and $I/16$.

maps output from each convolution stage to generate multi-scale features. More feature information can be captured by using the features of different convolution stages. From a global perspective, the proposed network is able to make full use of contextual information.

Since the number of lane pixels is far less than that of background pixels, we use feature extractors to extract lane features at different convolution stages. The upsampling module is used to recover the resolution of the feature map. The proposed network in this paper applies a feature extractor and an upsampling module to achieve flexible feature fusion (see Fig. 2). In addition, each convolution stage outputs the corresponding feature image and calculates the loss separately.

2.2 Self-attention Mechanism

As shown in Fig. 3, the self-attention module is able to capture the global geometric features of the image. The calculation of the attention values can be expressed as follows: first, the correlation between each value is queried and the weight factor of each key value is calculated. Then, the weight and the corresponding key value are weighted and summed. Thus, the self-attentive mechanism can be formulated as a series of key-value mappings:

$$O = \frac{1}{D(x)} V(x) \cdot (S(x_j, x_i) F(x_i)) \,, \tag{1}$$

where $O \in R^{C \times N}$ represents the output of the self-attention layer, $D(x)$ represents the normalization factor, $S(x_j, x_i)$ represents the degree of dependence between the region j and the region i, $V(x_i) = W_v x_i$, $F(x_i) = W_f x_i$, and W is the learned weight matrix, which is implemented 1×1 convolution. The output o of the self-attention module has the same size as the input x.

The self-attention module converts the upper-level feature map $x \in R^{C \times N}$ into two feature spaces Q, K for calculation separately, where $Q(x) = W_q(x)$, $K(x) = W_k(x)$, C indicates the number of channels and N indicates the number of feature positions of the upper-level features. We compute the similarity between different regions by extending the Gaussian function, namely:

$$S(x_j, x_i) = exp(Q(x_i)^T K(x_j)), \tag{2}$$

We first process the upper-level features using $P(x)$, $K(x)$. Then, we transpose $P(x)$ and multiply it with $K(x)$ to get $S(x_j, x_i)$, which represents the interrelationship among pixels and can be thought of as a correlation matrix. After that, we normalize $S(x_j, x_i)$ and activate it using the softmax function and we can get attentional map.

We combine the self-attention feature map with the input feature map proportionally. Therefore, the final output result is:

$$y_i = \gamma O_i + x_i, \tag{3}$$

where γ is a learnable scaler and it is initialized as 0.

2.3 Loss Function

The proposed method uses multi-scale and multi-level features to predict the probability distribution of pixels. Assume that T is the lane annotation. In this paper, we compute the classification loss of lanes using the prediction vector z:

$$L_{cls} = L_{ce}(z, T) \tag{4}$$

where L_{ce} is the cross entropy loss function.

In addition to the classification loss, we also adopt the similarity loss function which aim at modeling structural relations of lane. The similarity loss function can be expressed as:

$$L_{smi} = \sum_{i}^{H} \sum_{j}^{W} ||z_{i,j} - z_{i,j+1}||_1, \tag{5}$$

where $|| \cdot ||_1$ represents the L_1 norm.

Moreover, to further improve the ability of network for learning, the shape loss function is proposed to constrain the shape of the lane. We propose an equivariant cross regularization loss to constrain the shape of the lane:

$$L_{shape} = \sum_{i}^{H} \sum_{j}^{W-2} ||(z_{i,j} - z_{i,j+1}) - (z_{i,j+1} - z_{i,j+2})||_1. \tag{6}$$

where $|| \cdot ||_1$ represents the L_1 norm.

Table 1. Comparison with state-of the art methods on TuSimple and DataLake testing subset, the left shows the results of the TuSimle test subset, and the right shows the results of the DataLake test subset. Here "R18-SALSN" denotes ResNet18 as the backbone. '-' means the result is not available. The best performance and the second best performance are indicated in red and blue, respectively.

Method	Tusimple			DataLake			Time
	Accuracy	F-score	IoU	Accuracy	F-score	IoU	
ResNet-34 [15]	92.88	0.720	0.622	92.11	0.718	0.622	-
RESA [8]	96.33	0.822	0.735	96.63	0.842	0.711	135
CondLaneNet [9]	96.36	0.831	0.704	96.24	0.821	0.703	24
LaneATT [10]	96.38	0.828	0.723	96.31	0.806	0.705	151
SAD [16]	96.66	0.862	0.767	96.55	0.863	0.753	19
SGNet [17]	95.87	0.901	0.811	96.50	0.903	0.788	92
R50-SALSN	96.53	0.823	0.702	96.51	0.827	0.705	52
R101-SALSN	97.13	0.835	0.766	96.71	0.855	0.764	92
R152-SALSN	98.75	0.903	0.823	97.95	0.844	0.792	101

We utilize cross entropy as our auxiliary segmentation loss. In this way, the total loss can be written as:

$$\mathcal{L} = L_{seg} + \alpha L_{cls} + \beta L_{smi} + \gamma L_{shape} + \eta L_{IoU}, \tag{7}$$

where L_{seg} is the segmentation loss, L_{IoU} is the intersection-over-union (IoU) loss. The IoU loss is used to increase the intersection-over-union between the predicted lane pixels and ground truth lane pixels. The IoU loss can be written as: $L_{IoU} = 1 - \frac{N_p}{N_p+N_g+N_o}$, where N_p is the predicted number of lane pixels, N_g is the number of ground truth lane pixels and N_o is the number of lane pixels in the overlapped areas between predicted lane areas and ground truth areas. α, β, γ, and η are loss coefficients utilized to balance the influence of segmentation loss, similarity loss, shape loss, and IoU loss on the final task.

3 Experiments

In our network, batch normalization is used to speed up the convergence of the network. Our self-attention model requires a long time to learn the global features of the lane. During the training process, the initial global learning rate is 0.001 and decays to 1/10 of the initial after every 10k iterations. The momentum and weight decay are 0.9 and 0.0005, respectively. We adopted The stochastic gradient descent with momentum (SGDM) optimizer to minimize the loss function. All experiments in this paper are performed by using a single GeForce GTX 3090.

The TuSimple dataset [14] contains 3632 training images and 2782 test images, which are widely used in the field of lane detection. The DataLake dataset [18] is a large-scale lane detection dataset, which contains 10,161 training

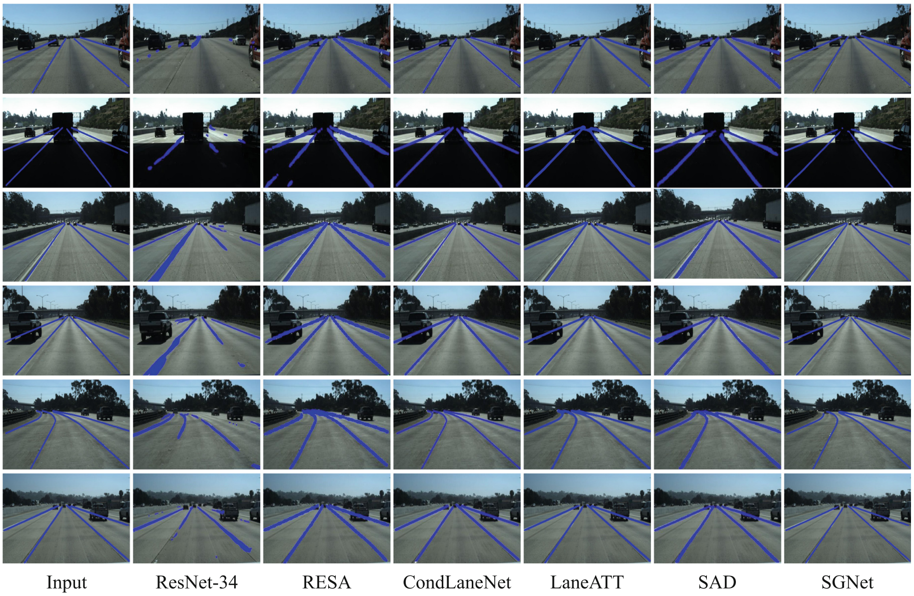


Fig. 3. Examples of experimental results on TuSimple lane detection dataset. From left to right: Input image, ResNet-34 [15], RESA [8], CondLaneNet [9], LaneATT [10] SAD [16], SGNet [17]. Lanes are marked in blue color. Ground truth lanes are drawn on the input image. (Color figure online)

images and 1,000 test images. The DataLake dataset contains a variety of road environments. We resize the images size of the TuSample dataset to 512 × 256 to save memory usage. The main evaluation criterion is accuracy. Besides, F-score and IoU are used for performance evaluation of lane detection.

We show the comparison between our method and state-of-the-art algorithms on the TuSimple test set in Table 1. From the quantitative comparisons in Table 1, our method shows the best performance. It can be clearly seen that our method has higher accuracy. Under good weather conditions, the performance of our method is more competitive. The corresponding qualitative analysis is shown in Fig. 3. It can be clearly seen that our method has higher accuracy. We show the running time of different algorithms in Table 1 so that we can compare the performance and complexity of the algorithms. Table 2 shows the F1 scores of different lane segmentation algorithms in different scenarios. The experimental results show that our method can achieve accurate lane segmentation in different scenarios. Table 3 shows the impact of introducing self-attention mechanism at different locations on network performance. It can be clearly observed that the introduction of self-attention mechanism in high-level or middle-level improves the performance of the model most obviously. Introducing the self-attentiveness mechanism at the low-level layers have a smaller improvement on the network performance. This is because the long-range dependencies between lanes are

Table 2. F1 evaluation of lane segmentation algorithms in different road scenarios.

Method	Normal	Crowd	Hlight	Night	Shadow	Curve
ResNet-34 [15]	0.871	0.681	0.519	0.587	0.618	0.626
RESA [8]	0.883	0.692	0.547	0.596	0.624	0.637
CondLaneNet [9]	0.886	0.694	0.569	0.602	0.627	0.642
LaneATT [10]	0.891	0.699	0.583	0.644	0.638	0.652
SAD [16]	0.911	0.705	0.673	0.685	0.670	0.646
SGNet [17]	0.918	0.703	0.634	0.696	0.684	0.672
R50-SALSN	0.890	0.698	0.559	0.628	0.635	0.644
R101-SALSN	0.916	0.702	0.664	0.683	0.675	0.653
R152-SALSN	0.932	0.715	0.694	0.697	0.732	0.738

Table 3. Performance of different location of the self-attention on TuSimple dataset. c_{ij} denotes that the addition of a self-attention module at the ith and jth channels. ResNet152 as the backbone network. The best performance and the second best performance are indicated in red and blue, respectively.

Location	Accuracy	Location	Accuracy
c_{01}	95.57	c_{12}	97.63
c_{02}	95.33	c_{13}	97.62
c_{03}	95.37	c_{23}	98.75

not effectively captured in the self-attentive mechanism. Besides, the low-level layer mainly detects low-level details of the image. The self-attention module is originally designated to encode more global information.

4 Conclusion

Since lanes have strong shape constraints but weak coherence, we model the long distance dependence between lanes based on a self-attentive mechanism. The network architecture in this paper can incorporate attention features into CNNs to achieve accurate lane line detection. To demonstrate the effectiveness and feasibility of the proposed network, we compare it with advanced lane detection algorithms separately to evaluate the proposed approach performance in lane detection. The experimental results demonstrate that our designed network outperforms all other methods in detecting lane lines. The real-time operation speed of lane detection algorithms is crucial for autonomous driving, and the real-time performance of the algorithms and the maximum safe operation speed of lanes will be the subject of our research in future work.

References

1. Peng, T., Su, L., Zhang, R., Guan, Z., Zhao, H., Qiu, Z.: A new safe lane-change trajectory model and collision avoidance control method for automatic driving vehicles. Expert Syst. Appl. **141**, 112953 (2020)
2. Tang, J., Li, S., Liu, P.: A review of lane detection methods based on deep learning. Pattern Recogn. **111**, 107623 (2021)
3. Wu, P.-C., Chang, C.-Y., Lin, C.H.: Lane-mark extraction for automobiles under complex conditions. Pattern Recogn. **47**(8), 2756–2767 (2014)
4. Fan, R., Wang, H., Cai, P., Liu, M.: SNE-RoadSeg: incorporating surface normal information into semantic segmentation for accurate freespace detection. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12375, pp. 340–356. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58577-8_21
5. Ko, Y., Lee, Y., Azam, S., Munir, F., Jeon, M., Pedrycz, W.: Key points estimation and point instance segmentation approach for lane detection. IEEE Trans. Intell. Transp. Syst. **23**(7), 8949–8958 (2022)
6. Lv, Z., Li, J., Dong, C., Li, H., Xu, Z.: Deep learning in the COVID-19 epidemic: a deep model for urban traffic revitalization index. Data Knowl. Eng. **135**, 101912 (2021)
7. Lv, Z., Li, J., Dong, C., Xu, Z.: DeepSTF: a deep spatial-temporal forecast model of taxi flow. Comput. J. (2021)
8. Zheng, T., et al.: RESA: recurrent feature-shift aggregator for lane detection. In: AAAI Conference on Artificial Intelligence, pp. 3547–3554 (2021)
9. Liu, L., Chen, X., Zhu, S., Tan, P.: CondLaneNet: a top-to-down lane detection framework based on conditional convolution. In: IEEE/CVF International Conference on Computer Vision, pp. 3773–3782 (2021)
10. Tabelini, L., Berriel, R., Paixao, T.M., Badue, C., De Souza, A.F., Oliveira-Santos, T.: Keep your eyes on the lane: real-time attention-guided lane detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 294–302 (2021)
11. Lee, M., Lee, J., Lee, D., Kim, W., Hwang, S., Lee, S.: Robust lane detection via expanded self attention. In: Winter Conference on Applications of Computer Vision, pp. 533–542 (2022)
12. Zhang, J., Lu, C., Li, X., Kim, H.-J., Wang, J.: A full convolutional network based on DenseNet for remote sensing scene classification. Math. Biosci. Eng. **16**(5), 3345–3367 (2019)
13. Wu, Z., Shen, C., Van Den Hengel, A.: Wider or deeper: revisiting the resnet model for visual recognition. Pattern Recogn. **90**, 119–133 (2019)
14. TuSimple: Tusimple benchmark. https://github.com/TuSimple/tusimple-benchmark. Accessed Nov 2019
15. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2017)
16. Hou, Y., Ma, Z., Liu, C., Loy, C.C.: Learning lightweight lane detection CNNs by self attention distillation. In: IEEE International Conference on Computer Vision, pp. 1013–1021 (2019)
17. Su, J., Chen, C., Zhang, K., Luo, J., Wei, X., Wei, X.: Structure guided lane detection. arXiv preprint arXiv:2105.05403 (2021)
18. DataLake. https://aistudio.baidu.com/aistudio/datasetdetail/54289?_=1621952478922

Enhancing Efficiency and Quality of Image Caption Generation with CARU

Xuefei Huang[1](✉), Wei Ke[1,2], and Hao Sheng[1,3,4]

[1] Faculty of Applied Sciences, Macao Polytechnic University, Macau SAR, China
{xuefei.Huang,wke}@mpu.edu.mo

[2] Engineering Research Centre of Applied Technology on Machine Translation and Artificial Inteligence of Ministry of Education, Macao Polytechnic University, Macau SAR, China

[3] State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China
shenghao@buaa.edu.cn

[4] Beihang Hangzhou Innovation Institute Yuhang, Yuhang District, Xixi Octagon City, Hangzhou 310023, China

Abstract. Image caption is textual explanation automatically generated by a computer according to the content in an image. It involves both image and natural language processing, and thus becomes an important research topic in pattern recognition. Deep learning has been successful in accomplishing this task, and the quality of captions generated by existing methods is already high. However, due to the broadness and variety of image caption applications, the current generated captions are still not sufficiently detailed, and the training efficiency can also be improved. Therefore, under the encoder-decoder framework of deep learning, how to use fewer parameters to improve the training efficiency and retain the quality of the generated image descriptions is a huge challenge. In this work, we introduce an improved method based on the encoder-decoder structure, adding an attention mechanism, and applying the content adaptive recurrent unit (CARU), as the decoder, to generate image captions. Inspired by GRU, CARU is designed to have comparable performance with fewer parameters, and is sensitive to the features in hidden layers. The experimental results show, based on MsCOCO dataset, the proposed method achieved better performance than that using GRU as the decoder, and took less training time, effectively improves the training efficiency.

Keywords: Image caption generation · Content adaptive recurrent unit · Feature extraction · Deep learning · Training efficiency

1 Introduction

For an image of complex scenes, people can clearly understand the content and quickly grasp the key points in the image. But for a computer reading only pixel

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 450–459, 2022.
https://doi.org/10.1007/978-3-031-19214-2_38

patterns, it is very difficult to understand the image and derive a caption. The main purpose of image caption is to allow the computer to correctly explain the scene and content in the image, and generates descriptive sentences with human reading habits [1]. Image caption is one of the key research goals in the field of artificial intelligence, and can be widely used in daily life. Such as visual Q&A, image retrieval, intelligent transportation, intelligent medical treatment, and intelligent early education [2].

Image caption is unlike other simple tasks such as image classification and object recognition, but a product of cross-domain fusion. It involves two technologies: computer vision (CV) and natural language processing (NLP). In recent years, the generalization ability of deep learning is being continuously applied in NLP, CV and other AI fields, and many research results have been obtained [3], it has naturally become the most common approach in image caption [4]. In fact, it is a combination of deep learning and the *encoder-decoder* structure, that uses the convolutional neural networks (CNN) as the encoder to obtain the representation vector of the picture, and the recurrent neural network (RNN) as the decoder to *translate* the image features into a sentence.

The descriptive sentences generated by this kind of methods are evaluated by indicators such as BLEU [5] and CIDEr [6], and it is confirmed that good results have been achieved [7]. However, in the face of a largely growing amount of real-time image information, it is needed and also very important to generate text descriptions more efficiently. On the basis of previous research, this work further discusses how to quickly generate accurate and comprehensive image caption sentences, and puts forward an improved scheme based on the encoder-decoder structure, and makes the following contributions.

- Based on the encoder-decoder structure of deep learning, an improved image caption generation model is proposed, which increases the accuracy of image comprehension and enhances the efficiency of feature processing.
- The content-adaptive recurrent unit (CARU) [8] is adopted as the decoder, which has fewer parameters than the gate recurrent unit (GRU), and is more sensible to the features in hidden layers, thus increasing the decoder performance.
- Under the condition of maintaining all the optimization performance of the original structure, the accuracy and operation speed have been improved to a certain extent.

2 Related Work

Image caption can be considered as a dynamic target detection, which generates an image summary from overall information. The previous methods were mainly based on template matching, using some operators to extract the features of an image, and obtaining the classification of the objects that may exist in the image. However, the method of using template matching is not suitable for all types of the images, and also limits the variety of output. In order to describe the text features of various images more accurately, Kuznetsova *et al.* proposed a method of similarity testing instead of templates, i.e., a method based on similar space retrieval [9]. However, the above two methods are not flexible enough. With the development of deep learning [10,11], it becomes the most commonly used methods in the field of image caption.

The deep semantic alignment model [1] explicitly aligns multiple local regions in the image with text description fragments, and proposes to combine Region-based CNN (R-CNN) and bidirectional RNN to construct an image caption model. The m-RNN model [12] combines deep CNN and deep RNN, interacts these two networks in a multi-modal layer, directly models the probability of input words and generated words, and uses the features to predict the words corresponding to the current time step, and generate the image description.

Long short-term memory (LSTM) solves the problem of RNN gradient explosion and can remember long sequences. The structure of gate recurrent unit (GRU) is simpler than that of LSTM, and the network has fewer parameters. Therefore, LSTM and GRU have been quickly applied to the field of image caption. The NIC (Natural image caption) model [13] was combined with Inception_V3 and LSTM network to extract image features and generate description sentences. The two-layer network structure has a deeper network layer and a stronger memory capacity of the model. The ability to generate corresponding description sentences has been significantly improved.

Applying GRU as a decoder in image caption results less training time [7], and it can also improve the accuracy of caption generation to a certain extent. The BeamAtt [14] model proposes a multi-modal architecture method, which combines the beam search on the basis of GRU. The feasibility of this method was justified by the comparisons with other existing methods. In order to make the generated text description more fluent, Pan *et al.* [15] used a double-layer GRU for experiments, and the features extracted from the images were trained in a shorter time, and it had achieved better performance in many evaluation indicators.

On the basis of the NIC model, Xu *et al.* [16] integrated the attention mechanism into the LSTM, so that the model was able to generate a description of a specific field according to the region of the image. The feature weight can be learned and adjusted to realize the *constant attention* of image content, and generate the description word by word.

Although the above methods have made significant achievements in the field of image caption, due to the complexity of image scenes, it often takes a lot of time to implement the text description of an image. Therefore, how to enhance the efficiency of image caption generation is worth our research.

3 Methodology

The deep neural network that is composed of an encoder and a decoder has achieved good performance in the image caption generation task. Therefore, we propose our image caption generation model based on the encoder-decoder structure, combined with ResNet [17], the attention mechanism, and the content

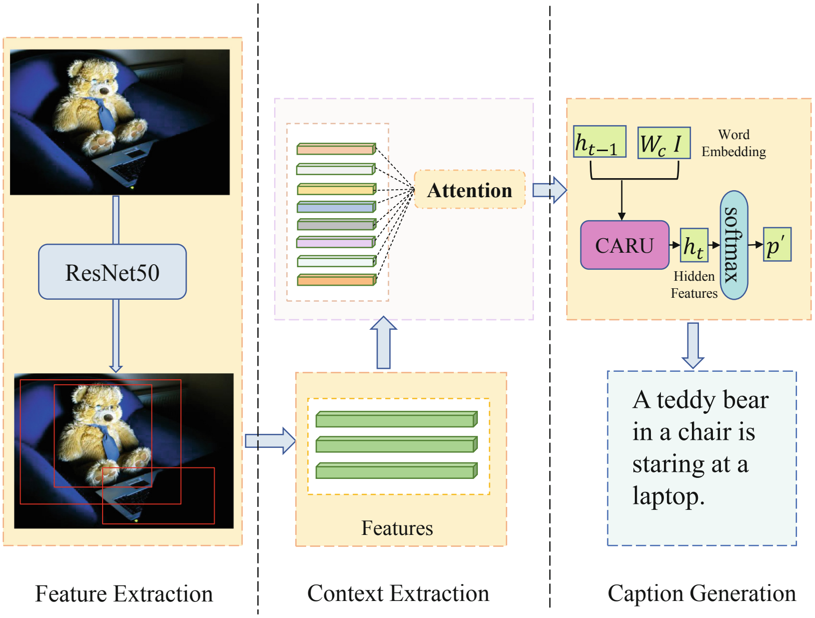


Fig. 1. Overall framework of the proposed model

adaptive recurrent unit (CARU) [8], to achieve the purpose of more comprehensive and accurate generation of image description, while reducing the number of parameters to enhance the efficiency.

3.1 Model Overview

Our model mainly consists of three parts: feature extraction, context extraction and caption generation. The overall framework is shown in Fig. 1. The main process of the model is as follows.

Feature Extraction. ResNet50 is used to extract the features of an image, and the residual network is used to build a deep network, which effectively solves the degradation problem and the gradient problem caused by network deepening, so that the extracted features can be more comprehensive.

Contextual Information Extraction. We use the image features as input of the attention mechanism, from a global perspective, combined with contextual information. It highlights the more important characteristic information to better guide the generation of descriptive sentences.

Caption Generation. The feature vector output by the attention mechanism is used as input of the decoder. We adopt CARU as the decoder, which not only overcomes the problem of long-term dependence, but also has fewer parameters, to process the image feature vector to generate a description sentence.

3.2 Image Feature Extraction

As mentioned earlier, CNN is widely used in feature extraction tasks and has achieved good results. The introduction of ResNet [17] combines the concept of residual representation commonly used in CV, and further applies it to the construction of CNN models. There is even a branch based on residual learning.

ResNet changes the way that most of the original networks have been directly using parameterized layers to learn the mapping between input and output, and uses multiple parameterized layers to learn the residual representation between input and output. Experiments show that trying to learn residuals is much easier than directly learning the mapping, the convergence speed is much faster, and higher classification accuracy can be achieved by using more layers. For images, the deeper the network, the more information at different levels will be extracted, the more hierarchical information combinations between different levels there will be, and the richer the feature information will be obtained.

3.3 Context Extraction

In image caption, the encoder must compress all the extracted image features into a fixed-length vector, and then pass it to the decoder as the input. However, when the fixed-length vector is compressed, some of the information from the input sequence may be lost. In addition, the image features extracted by the encoder, usually only have certain areas that are more relevant to the next word in the generation of the caption. The attention mechanism can imitate the human visual mechanism, paying attention to the important part of the information that assists the judgment in the image, and ignoring the irrelevant information.

By adding an attention mechanism and allowing the decoder to access the input sequence of the entire ResNet part, we introduce an attention weight α on the input sequence. The position containing important information can be prioritized by exploring different targets in the image. Such association between scenes and semantics enriches contextual information, and guides the generation of more comprehensive and descriptive sentences.

3.4 Caption Generation

Since image caption is a combination of NLP and CV, we substitute CARU for the GRU and LSTM as the decoder to generate text description with enhanced accuracy and efficiency. CARU is an improved RNN, which retains the performance of GRU and can overcome the problem of long-term dependency. It introduces a content adaptive gate, similar to the reset gate in GRU, which can convert the hidden states. Prior to this work, CARU has only been applied to NLP tasks and had no experience in the CV field.

In order to better receive the image feature information filtered by the attention mechanism and serve as the initial hidden state of CARU, the decoder only processes the current word instead of facing the entire sequence, as shown below,

$$r_t = W_r \cdot [h_{t-1}, x_t], \tag{1}$$

where the feature vector is input as the initial hidden state h_0, and is delivered to the content adaptive gate. The r_t can control the update of candidate status. The advantage of using CARU is that it alleviates the long-term dependency problem of interpretation, by first weighting the currently received input rather than encoding it directly into a hidden state. The new hidden state z_t is generated by combining the parameters of h_{t-1} and x_t, calculated as,

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]). \tag{2}$$

The purpose is to determine how proportions of the input are to be combined to the new hidden state. This approach allows tracking arbitrary long-term dependencies in the input sequence, which is computational (or practical) in nature: the long-term gradients being backpropagated can converge while training an RNN using the same recurrent unit. The $\widetilde{h}_t$ is similar to the update gate in GRU, used for the transformation of hidden state,

$$\widetilde{h}_t = \tanh(W_{\widetilde{h}} \cdot [h_{t-1}, x_t] + r_t). \tag{3}$$

By using the CARU unit partially, an RNN further solves the vanishing gradient problem, because CARU units also allow gradients to flow unchanged by the content-adaptive gate.

We use the $\odot$ as the Hadamard operator to combine the update gate with the weight of current feature,

$$l_t = \sigma(r_t) \odot z_t. \tag{4}$$

By combining the weight of the current feature, we obtain the capability similar to a GRU reset gate, but only based on the current input instead of the entire content. More specifically, it can be considered as the tagging task that connects the relation between the weight and parts-of-speech. The content adaptive gate combines the current feature with the weight, and finally obtain the output,

$$h_t = (1 - l_t) \odot h_{t-1} + l_t \odot \tilde{h}_t. \tag{5}$$

This affects the gradient amount instead of weakening the current hidden state.

According to the above formulæ, the complete structure of the CARU architecture is shown in the Fig. 2, where the direction of data flow is represented by arrows, the yellow rectangle refers to the neural network layer, and the purple circle refers to the gated operation.

Due to the unique structure of CARU, it has the good performance as LSTM and GRU, and reduces the use of parameters on the basis of GRU, making the operation more convenient. Therefore, using CARU as the decoder can maintain the quality of generated descriptive sentences and improve the training efficiency of the model to a certain extent.

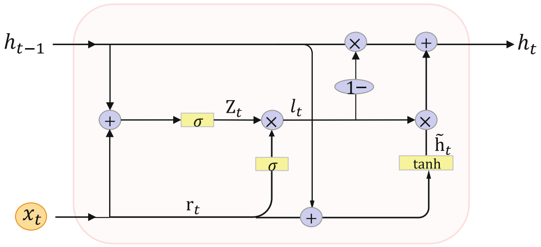


Fig. 2. Flow and structure of CARU

4 Experiments

In order to evaluate the proposed method, experiments were conducted on the public dataset. In the encoder part, ResNet50 is used to extract image features, and CARU and GRU are used as decoders to generate description sentences respectively.

4.1 Dataset and Data Pre-processing

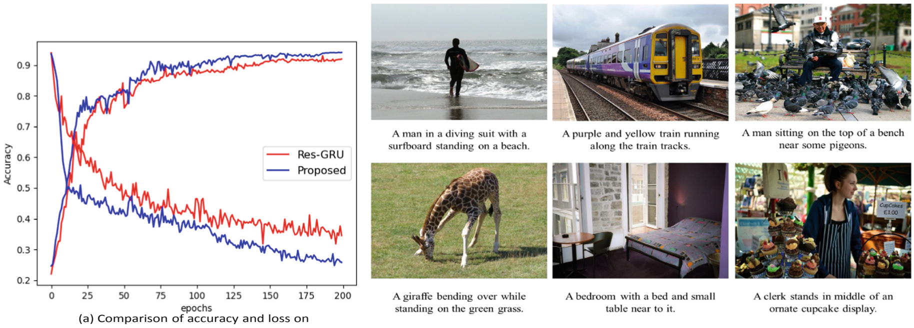
The MsCOCO [18] dataset has the largest amount of data, covering a wide range, and the image subjects involve various fields. It consists of 123,000 images of complex scenes, in which there are common transactions in people, animals and natural environments. Each image corresponds to 5 text descriptions.

In order to verify the effectiveness of our model, we chose to evaluate the performance on the MsCOCO dataset[1]. Before training the model, all letters in the text descriptions were converted to lowercase, and non-alphanumerical characters were trimmed. To avoid the generated text description being disturbed by rare words, it was also necessary to delete the words with a number of occurrences less than 5 in the text.

4.2 Implementation

First, ResNet50 was used as the encoder to extract the features of the images in the MsCOCO dataset, and the feature vector was mapped to the semantic space. The learning rate was set to 1e−4, and Adam algorithm was used as the optimizer. Next, the attention weight was added to the feature vector, where the attention dimension was 512. Then CARU was used as the decoder to process the vocabulary, and *Softmax* calculated the probability of the next output word, before finally description sentences were generated. The learning rate was set to 8e−4. Adam was again used for optimization. All experiments used the same dataset in each test. The batch size was 128, with a total of 200 epochs.

[1] In order to facilitate evaluation, we follow the experience of Karpathy *et al.* to separate the validation set, dividing the 10,000 images equally into two parts, 5000 for test, 5000 for verify.



(a) Comparison of accuracy and loss on MsCOCO dataset after 200 epoch

A man in a diving suit with a surfboard standing on a beach.

A purple and yellow train running along the train tracks.

A man sitting on the top of a bench near some pigeons.

A giraffe bending over while standing on the green grass.

A bedroom with a bed and small table near to it.

A clerk stands in middle of an ornate cupcake display.

(b) Images with text descriptions generated by the proposed method

Fig. 3. Performance and output of the proposed method

So as to compare the proposed model, a model using Pytorch to implement ResNet50 as encoder, overlay attention mechanism, and GRU as decoder, referred to as the *R-GRU* model, was constructed, in which the setting of super parameters was consistent with the proposed model.

4.3 Results and Analysis

The proposed model is compared with the R-GRU model on the MsCOCO dataset. As expected, the model using CARU has faster convergence speed and higher accuracy. The comparison of training results is shown in Fig. 3(a).

After the training, several images were randomly selected from the dataset, and the captions were generated using the proposed model. As shown in Fig. 3(b), the proposed model can successfully detect the objects, the quantity, the positions between objects, and the relationship between objects in the images. The generated text descriptions were relatively smooth and logical, but lacks richer emotions, which can be further improved.

Additionally, we compared the performance of the proposed model with the state-of-the-art image caption generation models on the MsCOCO dataset. Due to the low performance of text descriptions generated by early template-based methods and retrieval-based methods, and no experiments were conducted using standard datasets and evaluation metrics, the methods selected for the performance comparisons were all based on deep learning. The selected methods included those mentioned in Sect. 2: the multi-modal space-based method m-RNN [12], the visual space-based method NIC [13], the attention-based methods Soft-Attention [16] and VQA [2], and the R-GRU model for comparing CARU efficiency. The performance comparison results are summarized in Table 1.

In the table, B@N represents the performance evaluation index BLEU, and CIDEr is an evaluation index specially used in image description. Except for R-GRU, the data of each indicator comes from the corresponding original literature, a '–' is displayed if there is no such an indicator. As mentioned above, the proposed model outperformed most of the other models.

Table 1. Performance comparison with state-of-the-art methods on MsCOCO dataset.

Model	B@1	B@2	B@3	B@4	CIDEr
m-RNN [12]	67.1	49.0	35.0	25.0	–
NIC [13]	66.6	46.1	32.9	24.6	–
Soft-Attention [16]	70.7	49.2	34.4	24.3	–
VQA [2]	**79.8**	–	–	36.3	120.1
R-GRU	78.8	50.8	35.6	36.1	119.7
Proposed	79.1	**51.2**	**35.7**	**36.4**	**121.3**

5 Conclusion

In this work, an improved image caption generation model is proposed. We base the model on the existing encoder-decoder structure, ResNet50 is used as the encoder, attention mechanism is added, and CARU is used as decoder. Experiments show that the proposed model is better than using GRU as decoder, and the generated description statements are more in line with human language logic. Moreover, the proposed model uses fewer parameters and shortens the running time. Our experience also demonstrates that CARU also has good performance in solving CV/NLP hybrid problems. Future work will focus on combining other excellent CNNs to obtain more effective image features to improve the accuracy and comprehensiveness of caption generation.

Acknowledgements. This study is partially supported by the National Key R&D Program of China (No.2019YFB2101600), the National Natural Science Foundation of China (No.61 872025), Macao Polytechnic University (RP/ESCA-03/2020), and the Open Fund of the State Key Laboratory of Software Development Environment (No. SKLSDE-2021ZX-03). Thanks for the support from HAWKEYE Group.

References

1. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3128–3137 (2015)
2. Anderson, P., et al.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6077–6086 (2018)
3. Hodosh, M., Young, P., Hockenmaier, J.: Framing image description as a ranking task: data, models and evaluation metrics. J. Artif. Intell. Res. **47**, 853–899 (2013)
4. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
5. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)

6. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: consensus-based image description evaluation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4566–4575 (2015)
7. Parikh, H., Sawant, H., Parmar, B., Shah, R., Chapaneri, S., Jayaswal, D.: Encoder-decoder architecture for image caption generation. In: 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA), pp. 174–179. IEEE (2020)
8. Chan, K.-H., Ke, W., Im, S.-K.: CARU: a content-adaptive recurrent unit for the transition of hidden state in NLP. In: Yang, H., Pasupa, K., Leung, A.C.-S., Kwok, J.T., Chan, J.H., King, I. (eds.) ICONIP 2020. LNCS, vol. 12532, pp. 693–703. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63830-6_58
9. Kuznetsova, P., Ordonez, V., Berg, A., Berg, T., Choi, Y.: Collective generation of natural image descriptions. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 359–368 (2012)
10. Xu, H., Cai, Z., Li, R., Li, W.: Efficient CityCam-to-edge cooperative learning for vehicle counting in ITS. IEEE Trans. Intell. Transp. Syst. **23**(9), 16600–16611 (2022)
11. Wang, J., Cai, Z., Yu, J.: Achieving personalized k-anonymity-based content privacy for autonomous vehicles in CPS. IEEE Trans. Industr. Inf. **16**(6), 4242–4251 (2019)
12. Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., Yuille, A.: Deep captioning with multimodal recurrent neural networks (m-RNN). arXiv preprint arXiv:1412.6632 (2014)
13. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: a neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3156–3164 (2015)
14. Sawarn, A., Srivastava, S., Gupta, M., Srivastava, S.: BeamAtt: generating medical diagnosis from chest X-Rays using sampling-based intelligence. In: Srivastava, S., Khari, M., Gonzalez Crespo, R., Chaudhary, G., Arora, P. (eds.) Concepts and Real-Time Applications of Deep Learning. EICC, pp. 135–150. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-76167-7_9
15. Pan, Y., Wang, L., Duan, S., Gan, X., Hong, L.: Chinese image caption of inceptionv4 and double-layer GRUs based on attention mechanism. J. Phys. Conf. Ser. **1861**(1), 012044 (2021)
16. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. In: International Conference on Machine Learning, pp. 2048–2057. PMLR (2015)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
18. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48

IMBR: Interactive Multi-relation Bundle Recommendation with Graph Neural Network

Jiabao Sun[1,2], Nan Wang[1,2](✉), and Xinyu Liu[1,2]

[1] School of Computer Science and Technology, Heilongjiang University,
Harbin, China
2211923@s.hlju.edu.cn, wangnan@hlju.edu.cn

[2] Key Laboratory of Database Parallel Computing of Heilongjiang Province,
Harbin, China

Abstract. Traditional approaches focus on an individual item of most interest to users. However, in most realistic scenarios, the platform needs to recommend a group of items at one time for users' convenience, called bundle recommendation. e.g., a music playlist containing multiple songs. The existing bundle recommendations usually use manual methods to artificially build bundles for different items, ignoring the obtained bundles and the potential relationships among the items in the bundle, especially the relationships between bundles. Therefore, how integrating multiple complex interactions into bundles and obtaining high-quality bundle recommendation is an important problem. To solve the problem, we propose a novel model named IMBR (short for Interactive Multi-Relation Bundle Recommendation with Graph Neural Network). Firstly, we construct a multi-relation interaction graph to capture the interaction relation from the user view. At the same time, we get bundle subordination relation from the item view. They can obtain richer representations of users, bundles, and items. Secondly, we design a bundle frequent term constraint algorithm (BFTC) to constrain the composition of items in a bundle and pay attention to the similarity between bundles. Finally, we leverage a multi-task learning framework to capture user personalized preferences to improve the performance of bundle recommendation. Extensive experiments on two real-world datasets with different scales show that our method can significantly outperform various baseline approaches.

Keywords: Recommendation system · Bundle recommendation · Graph neural networks

1 Introduction

Recommendation system have been playing an increasingly important role in informed consumption, services, and decision-making in the overloaded infor-

This work was supported by Heilongjiang Provincial Natural Science Foundation of China(No. LH2021F047).

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 460–472, 2022.
https://doi.org/10.1007/978-3-031-19214-2_39

mation era and digitized economy [1]. The main purpose of a recommendation system is to predict users' preferences according to their historical behavior and recommend the items they are interested in [2]. However, in many scenarios, the platform needs to show a set of items to users to get more choices from them, such as e-commerce platforms recommend clothing packages, music platforms recommend playlists [3], and so on. We term all such scenarios that concern recommending a set of items as Bundle Recommendation which needs to predict a user's preference on a bundle of items rather than a single item [4].

Traditional collaborative filtering based approaches regard bundles as atomic units. [3,5] simultaneously utilize the user's interactions with both items and bundles under the Bayesian Personalized Ranking(BPR) [6] framework. With the rise of deep learning, some researchers are applying deep learning techniques for bundle recommendation. [7] designed an attention network to capture user preferences for component items to represent bundles, and jointly model user-bundle interactions and user-item interactions in a multi-task manner, which transferring the benefits of one task to another utilizing multi-task learning. To better capture neighbor interaction information, [8] unifies user-item interactions and user-bundle interactions into a heterogeneous graph with item nodes as the bridge, and GCN [9] was used to capture the representation of bundles. However, in most real-world scenarios of bundle recommendation is more complex, where user preferences can be derived from both interactive items and interactive bundles. Similarly, there are special relationships between bundles and their containing items. These relationships inevitably affect the performance of bundled recommendation. In this paper, the main work is to improve the performance of bundle recommendation by adding these relations to bundle recommendation.

In this paper, we utilize different relations to solve bundle recommendation problems based on graph neural networks and propose a novel model IMBR. Recent work [8] has considered utilizing items as middleware to build bundle-bundle matrices via a path bundle-item-bundle. However, these works do not consider multiple relations simultaneously to obtain higher-order representations of users, items, and bundles. Moreover, previous bundle recommendation indicated the user's preferences by taking the bundle as a whole. Such a training procedure implicitly assumes each item in the bundle is related to the user's preferences, which might not always hold. therefore, this behavior may lead to unreliable items being mixed into the bundle. It is worth noting that the IMBR model designs a constraint algorithm to simplify the bundles, considering that there are many overlapping items between different bundles, which greatly compromises the diversity of exposed projects within the bundles. The contributions of our paper are summarized as follows:

- We present a novel model Interactive Multi-Relation Bundle Recommendation with Graph Neural Network, which utilizes a combination of different types of relations between nodes and graph neural networks, effectively solving the problem that bundle recommendation bundles are too sparse as well as missing representations and insufficient information.
- We utilize graph neural networks to fuse multiple relationships in the bundle and constructed the multi-relation interaction graph from the user perspective

and project perspective to obtain better embedding representations of users, items, and bundles.
- We design a bundle frequent item constraint algorithm to reduce the rate of excessive occurrence of identical or similar items in the bundle solve the problem of similar items in the bundle at the time of characterization and increase the diversity of the bundle.
- We have conducted a lot of experiments on two real datasets, to verify the validity of the IMBR. The experimental results show that our model is superior to the existing mainstream models in solving the bundle problem.

2 Related Work

2.1 Graph Neural Network

In recommendation system, the main challenge is to learn valid user/item representations from interactions and auxiliary information (if present). In recent years, graph neural network techniques have been widely used in recommendation system because most of the information in recommendation system is essentially graph structure and GNNs have advantages in graph representation learning. For example, [10] utilized different dependency relations between nodes to solve the CTR prediction problem. In the bundle recommendation, [8] firstly constructed the user-item/bundle interaction graphs with GNN.

2.2 Bundle Recommendation

In recommendation domain, several works [9] have been made in solving the bundle recommendation problem. Bundle recommendation mainly concentrates on not only capturing the relationships among users, items, and bundles, but also recommending lists of items to users. LIRE [5] solve the recommendation problem of user-generated item lists on the Bayesian Personalized Ranking (BPR) framework. [3] proposed jointly modeling the interaction between user-item and user-bundle, which combines two types of latent factor models, BPR and word2vec. DAM [7] design a factorized attention network to aggregate the item embeddings in a bundle to obtain the bundle's representation.

3 Problem Formulation

Given a set of users $U = \{u_1, u_2, \cdots, u_N\}$, a set of items $I = \{i_1, i_2, \cdots, i_M\}$, and a set of bundles $B = \{b_1, b_2, \cdots, b_K\}$, where N, M, and K are the number of users, items, and bundles, respectively. For each bundle $b_s \in B$, it consists of a set of items, $b_s = \left\{i_{s_1}, \ldots, i_{s_j}, \ldots, i_{s_{|b_s|}}\right\}$, where $|b_s|$ denotes the bundle size (larger than 1), and each item v_{s_j} in the bundle belongs to the set I $(e.g., v_{s_j} \in I)$. we define user-item interactions matrix, user-bundle interactions matrix and bundle-item subordination matrix as $\boldsymbol{X}_{N \times M} = \{x_{ui} | u \in U,\ i \in I\}$, $\boldsymbol{Y}_{N \times K} = \{y_{ub} | u \in U,\ b \in B\}$, and $\boldsymbol{Z}_{K \times M} = \{z_{bi} | b \in B,\ i \in I\}$ with a binary value at each entry, respectively. An observed interaction $x_{ui} = 1$ means user u once interacted item

i, and an observed interaction $y_{ub} = 1$ means user u once interacted bundle b. Similarly, an entry $z_{bi} = 1$ means bundle b contains item i. The problem of bundle recommendation is formulated as follows:

Input: User-bundle interaction matrix $\boldsymbol{X}_{N\times M}$, user-item interaction matrix $\boldsymbol{Y}_{N\times K}$, and bundle-item subordination matrix $\boldsymbol{Z}_{K\times M}$.

Output: A recommendation model that estimates the probability that user u will interact with bundle b.

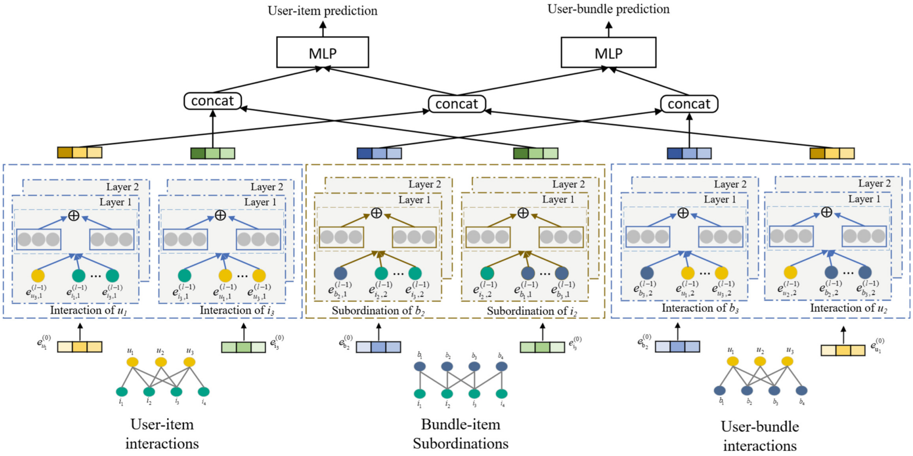


Fig. 1. The framework of the proposed IMBR.

4 Methodology

In this section, we will illustrate the details of our proposed method for bundle recommendation. The overall architecture of our proposed method is illustrated in the Fig. 1.

4.1 Multi-relation Graph Representation

To obtain a more meaningful representation of the bundle, we first constructed three relation graphs that these were the user-item interaction graph, the user-bundle interaction graph, and the bundle-item subordination graph from the user-view and the item-view. The interaction relation and subordination relation can be represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where nodes are $\mathcal{V}$ consisting of user nodes $u \in U$, bundle nodes $b \in B$ and item nodes $i \in I$. Edges are $\mathcal{E}$ consisting of the user-item interaction edges (u, i) with $x_{ui} = 1$ meaning an observed interaction between the user u and the item i, user-bundle interaction edges (u, b) with $y_{ub} = 1$ representing an observed interaction between the user u and the bundle b, and bundle-item subordination edges (b, i) with $z_{bi} = 1$ meaning that bundle b contains item i.

For each node on the relational graph, we apply one-hot encoding to its input and compress it into a dense vector. We use F_U, F_I, F_B to denote the initial user embedding matrix, the item embedding matrix and the bundle embedding matrix, which have the same embedding size D, respectively. The dense feature vector of a user, item and bundle can be defined as follows:

$$f_u = F_U^T v_u, \quad f_i = F_I^T v_i, \quad f_b = F_B^T v_b \tag{1}$$

where $v_u \in \mathbb{R}^M$, $v_i \in \mathbb{R}^N$, $v_b \in \mathbb{R}^K$, denotes the one-hot feature vector for user u, item i, and bundle b, respectively.

4.2 Interaction Matrix Construction

The purpose of the recommendation system is to capture the user's preferences and recommend items that the user likes. Therefore, from the user's view, we can obtain two kinds of interaction relations from the user-item interaction history and the user-bundle interaction history. We consider the construction of a corresponding interaction graph for them, the adjacency matrix of interaction graph as:

$$A_{ui} = \begin{bmatrix} I_u & X \\ X^T & I_i \end{bmatrix}, \quad A_{ub} = \begin{bmatrix} I_u & Y \\ Y^T & I_b \end{bmatrix} \tag{2}$$

where $I_u \in \mathbb{R}^{N \times N}$, $I_i \in \mathbb{R}^{M \times M}$, $I_b \in \mathbb{R}^{K \times K}$ are identity matrices for users, items and bundles. Inspired by GCN [9] to avoid ignoring the characteristics of the nodes themselves, we assume that every node is self-connected. Then we construct the initial feature matrix of the adjacency matrices $H_{ui}^{(0)} \in \mathbb{R}^{L_1 \times D}$, $H_{ub}^{(0)} \in \mathbb{R}^{L_2 \times D}$ and $L_1 = N + M$, $L_2 = N + K$, where D is the dimension of the feature vectors. The feature matrix of is defined as follows:

$$H_{ui}^{(0)} = \begin{bmatrix} F_{U,1} \; F_{I,1} \end{bmatrix}, \quad H_{ub}^{(0)} = \begin{bmatrix} F_{U,2} \; F_{B,1} \end{bmatrix} \tag{3}$$

To better capture the representations of users, items, and bundles from user-item interactions and user-bundle interactions. We refine the representation of nodes by building an embedding propagation layer that it is similar to the most GNN-based methods [9,11–13], where the update process for l-th layer is as follows (Since the update process is the same for both interactions, instead of the user-item interaction and the user-bundle interaction, we use *):

$$H_{u*}^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}} \widetilde{A}_{u*} \widetilde{D}^{-\frac{1}{2}} H_{u*}^{(l)} W_{u*}^{(l)}) \tag{4}$$

$$H_{u*} = \sum_{i=0}^{l} H_{u*}^{(i)} \tag{5}$$

where $\widetilde{A}_{u*}$ is the adjacency matrix of the undirected graph $\mathcal{G}$ with added self-connected. $W_{u*}^{(l)}$ is the trainable matrix for l-th layer. σ is the sigmoid activation function. $H_{u*}^{(l)}$ is the input embedding for l-th layer.

Since the initial feature matrix $H_{u*}^{(0)}$ is made up of the initial feature matrix of user and the initial feature matrix of item/bundle, we can get the feature matrix $F'_{I,1}$ for item, $F'_{B,1}$ for bundle and $F'_{U,1}$, $F'_{U,2}$ for user.

4.3 Containment Relation Matrix Construction

In the previous section, we constructing the two interactions from the user's view and obtain the feature matrix GNN-based methods. Previous bundle recommendation works have focused on the interaction relations. They have ignored the core of bundle recommendation which is the composition of the bundle.

In this section, we will learn about bundle-item subordination relations. We construct a subordination relation adjacency matrix from the item view, which is different from interaction relation. The subordination relation considers the degree of similarity between bundles and constructs a bundle-bundle similarity matrix. The adjacency matrix of subordination relation graph is defined as:

$$A_{bi} = \begin{bmatrix} B & Z \\ Z^T & I_i \end{bmatrix} \tag{6}$$

where $B \in \mathbb{R}^{K \times K}$ is the similarity matrix of bundle. $H_{bi}^{(0)} \in \mathbb{R}^{L \times D}$ is the initial feature matrix containing the bundle nodes and the item nodes, and $L = K + M$. The feature matrix of is defined as follows:

$$H_{bi}^{(0)} = \left[F_{B,2} \ F_{I,2}\right] \tag{7}$$

Similar to Eq. (4) and Eq. (5), the process of updating the l-th layer of subordination relation is as follows:

$$H_{bi}^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}} \widetilde{A}_{bi} \widetilde{D}^{-\frac{1}{2}} H_{bi}^{(l)} W_{bi}^{(l)}) \tag{8}$$

$$H_{bi} = \sum_{i=0}^{l} H_{bi}^{(i)} \tag{9}$$

After the above update operation, we can split H_{bi} into two parts. The feature matrix of bundle becomes $F'_{B,2}$ and the feature matrix of item becomes $F'_{I,1}$.

4.4 BFTC Algorithms

Due to the complexity of composition terms, the same item may appear in different bundles. It is will result high similarity between bundle and bundle, which will lead to inaccurate bundle feature. Therefore, we design a bundle frequent term constraint algorithm(BFTC), which uses TF-IDF to constrain the composition of bundles to avoid frequent items from weakening the bundle representation. Let the items in the bundle more accurately representate the characteristics of the bundle.

TF-IDF is a common method for information retrieval which can be divided into two phases. It first calculates the frequency of items in the bundle and then the frequency of items in the inverse bundle of the corpus. In our model, the corpus can be seen as a record of all user interactions. For the above assumptions, we first calculates the frequency of occurrence of a item v_i in the bundle b_u generated for the user u. The calculation formula is defined as follows:

$$TF_{v_i} = \frac{n_{ij}}{\sum_k n_{kj}} \tag{10}$$

Algorithm 1: BFTC Algorithm

```
Require:
    The set of users $U = \{u_1, u_2, \ldots, u_N\}$;  The set of items $I = \{v_1, v_2, \ldots, v_M\}$;
    The set of bundles $B = \{b_1, b_2, \ldots, b_K\}$;  constraint rate $\delta \in [0.01, 0.02, 0.05]$;
Ensure:
    The set of constrained bundles $B^* = \{b_1, b_2, \ldots, b_{K'}\}$;
for $i = 1$ to $M$ do
    "equation (11)"
end for
for $b = 1$ to $K$ do
    "equation (10)"
    for each item $i$ in bundle $b$ do
        "formula(12)" and $Rank(\mathbf{v}_i)$
        if $Rank(\mathbf{v}_i) < ceil(|b_s| \cdot \delta)$ then
            $drop(\mathbf{v}_i, \mathbf{b})$
        end if
    end for
end for
```

where TF_{v_i} is the frequency of item v_i appearing in bundle b_u, $\sum_k n_{kj}$ is the number of the items in the bundle. The value of TF_{v_i} is usually normalized. To some extent, it can indicate the importance of the item v_i. At the same time, the importance of the item v_i decreases inversely with the frequency of its appearance in a large list of bundles. Next, we calculate the reverse frequency for IDF_{v_i}:

$$IDF_{v_i} = \log \frac{N+1}{N_{v_i}+1} \tag{11}$$

$$C_{v_i} = TF_{v_i} \times IDF_{v_i} \tag{12}$$

Here, N is the total number of bundle lists, and N_{v_i} is the number of bundles containing item v_i. We calculate the weight C_{v_i} for item i by multiplying the two frequencies together, with lower weights indicating more frequent occurrences of the item. Therefore, we can obtain the frequency of items within a particular bundle, and the reverse frequency of the item in the overall bundle list.

4.5 Interactive Multi-relation Model Predictions

In summary, we can obtain three relational bipartite graphs that they are the bundle-item containment bipartite graph, the user-item interaction bipartite graph, and the user-bundle interaction bipartite graph. We can obtain two kinds of each node embedding representations from each of the above relationship graphs. We can concatenate the outputs from each bipartite graph to generate the final representations of the nodes as follows:

$$e_u = [e_{u,1}, e_{u,2}], \quad e_i = [e_{i,1}, e_{i,2}], \quad e_b = [e_{b,1}, e_{b,2}] \tag{13}$$

where $e_{u,1} \in F'_{U,1}$, $e_{u,2} \in F_{U,2'}$, $e_{i,1} \in F'_{I,1}$, $e_{i,2} \in F'_{I,2}$, $e_{b,1} \in F'_{B,1}$, $e_{b,2} \in F'_{B,2}$. After obtaining a unified representation of each type of node, we propose to concatenate the unified embedding of two nodes, and then use a two-layer MLP to learn the complex implicit interactions. The score between a user u and an bundle b for recommendation task is computed as follows:

$$\hat{y}_{ub} = \sigma_2(W_2^T(\sigma_1(W_1^T([e_u,\ e_b]) + b1)) + b2) \tag{14}$$

where W_x, b_x and σ_x denote the weight matrix, bias vector and activation function for the x-th layer of MLP, respectively. Similarly, we can calculate the score $\hat{y}_{ui}$ between user u and item i. We take the bundle with interaction as a positive sample and randomly select an unobserved bundle as a negative sample. Then for model optimization, we adopt the Bayesian Personalized Ranking loss [6]:

$$\mathcal{L}_{bundle} = \sum_{(j,e,f)\in R} - \ln \sigma(\hat{y}_{je} - \hat{y}_{jf}) + \lambda_b \|\theta_b\|_2^2 \tag{15}$$

where $R = (j,e,f)|(j,e) \in y^+, (j,f) \in y^-$ denotes the training dataset involving the observed interactions y^+ and unobserved counterparts y^-, $\sigma(\cdot)$is the sigmoid function, λ is the coefficient controlling L_2 regularization. θ_b is the set of model parameters for the bundle prediction task. When there is only one item in the bundle, the system can also recommend for single item. Similarly, we define the loss function for the Item prediction task as follows:

$$\mathcal{L}_{item} = \sum_{(s,p,q)\in Q} - \ln \sigma(\hat{y}_{sp} - \hat{y}_{sq}) + \lambda_i \|\theta_i\|_2^2 \tag{16}$$

where $Q = (s,p,q)|(s,p) \in y^+, (s,q) \in y^-$ denotes the training dataset involving the observed interactions y^+ and unobserved counterparts y^-.

$$\mathcal{L}_{loss} = \mathcal{L}_{bundle} + \mathcal{L}_{item} \tag{17}$$

Table 1. Dataset statistics

Dataset	NetEase	Youshu
User	18,528	8,039
Bundle	22,864	4,771
Item	123,628	32,770
User-Bundle	302,303	51337
User-Item	1,128,065	138,515
Bundle-Item	1,778,838	176,667

5 Experiments

In this section, we conduct experiments on two real-world datasets for bundle recommendation to evaluate our proposed IMBR, with the purpose of answering following research questions:

- RQ1: How does IMBR perform compared with previous approaches?
- RQ2: How do different components affect the results of IMBR?
- RQ3: How do parameters influence the result of IMBR?

5.1 Datasets and Evaluation Metrics

We evaluate the proposed IMBR and all baselines on following two real-world public datasets are shown in Table 1: *NetEase* is a music dataset collected from Netease Cloud Music[1] which is provided by the work EFM [3]. The dataset enables users to construct a list of songs with a specific theme, and we deliberately to select the list of users and songs to verify bundle recommendation algorithm. *Youshu* is a dataset of book sales, which is obtained from a Chinses book review site[2], similar to netease cloud music. It's just interaction between the user and the object item.

In this paper, we conduct experiments in two recommendation tasks. For top-N recommendation task, we adopt two widely-used evaluation protocols to evaluate the effectiveness of our proposed method: Recall@K and NDCG@K, and we set K = {20, 40, 80}.

Table 2. Performance comparison on the two datasets

Dataset	Method	R@20	N@20	R@40	N@40	R@80	N@80
NetEase	MFBPR	0.0355	0.0181	0.0600	0.0246	0.0948	0.0323
	GCN	0.0402	0.0204	0.0657	0.0272	0.1051	0.0362
	NGCF	0.0384	0.0198	0.0636	0.0266	0.1015	0.0350
	RGCN	0.0470	0.0210	0.0667	0.0280	0.1112	0.0378
	DAM	0.0411	0.0210	0.0690	0.0281	0.1090	0.0372
	BGCN	0.0491	0.0258	0.0828	0.0346	0.1304	0.0453
	Ours	**0.0633**	**0.0312**	**0.1045**	**0.0395**	**0.1615**	**0.0511**
	Improv.	28.92%	20.93%	26.21%	14.16%	23.85%	12.80%
Youshu	MFBPR	0.1959	0.1117	0.2735	0.1320	0.3710	0.1543
	GCN	0.2032	0.1175	0.2770	0.1371	0.3804	0.1605
	NGCF	0.2119	0.1165	0.2761	0.1343	0.3743	0.1561
	RGCN	0.2040	0.1069	0.3017	0.1330	0.4169	0.1595
	DAM	0.2082	0.1198	0.2890	0.1418	0.3915	0.1658
	BGCN	0.2347	0.1345	0.3248	0.1593	0.4355	0.1851
	Ours	**0.2690**	**0.1401**	**0.3602**	**0.1642**	**0.4777**	**0.1905**
	Improv.	14.61%	4.16%	10.90%	3.08%	9.69%	2.92%

5.2 Baseline

- MFBPR [6]: This work applies a Bayesian Personalized Ranking learning framework to the matrix factorization method.

[1] http://music.163.com.

[2] http://www.yousuu.com.

- GCN [9]: This method uses GCN to construct the user-item-bundle tripartite interaction graph for bundle recommendation.
- NGCF [13]: This method uses NGCF to construct the user-item-bundle tripartite interaction graph for bundle recommendation.
- RGCN [14]: RGCN is GCN based method developed to deal with the multi-relational graph.
- DAM [7]: This work uses the factorized attention mechanism and multi-task framework to capture bundle-level association and collaborate signals.
- BGCN [8]: BGCN proposes a graph neural network model to explicitly model complex relations between users, items, and bundles.

5.3 Performance Comparison

The results of all the methods are reported in Table 2. For the results, we have the following observations. Compared with all advanced baselines IMBR is always the best performance in two datasets. We can make the following observations from the experimental results: 1) In the two datasets, IMBR improved by 28.92% and 14.61% in Recall@20 compared to the best baseline, respectively. In terms of NDCG@20 compared to baseline and best increased by 20.93% and 4.16%, respectively. The above results show that IMBR is effective in solving the bundle recommendation problem by explore multiple relationships. 2) Compared with traditional method MFBPR, we find that the some GNNs methods [8,9,13,14] have a significant improvement in bundle recommendation. It is verified that graph neural networks can capture complex topology and high-order connections well. 3) DAM still outperforms most GNNs models although without graph neural networks. Therefore, it can be demonstrated that IMBR is effective to introduce user-item interaction and multi-task learning methods to the relation graph structure inspired by DAM.

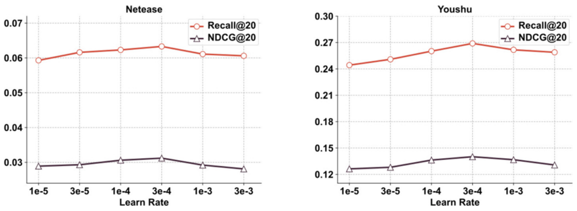
Table 3. Performance comparison of variations

Model	NetEase		Youshu	
	R@20	N@20	R@20	N@20
w/o IR-B	0.0566	0.0262	0.2480	0.1226
w/o IR-I	0.0571	0.0269	0.2488	0.1264
w/o CR	0.0503	0.0247	0.2391	0.1218
w/o BC	0.0577	0.0269	0.2558	0.1326
w/o ML	0.0589	0.0293	0.2688	0.1341
ℓ-1	0.0601	0.0301	0.2611	0.1355
ℓ-2	**0.0633**	**0.0312**	**0.2690**	**0.1401**
ℓ-3	0.0591	0.0302	0.2549	0.1369
ℓ-4	0.0557	0.0281	0.2513	0.1329
IMBR	0.0633	0.0312	0.2690	0.1401

5.4 Ablation Study

To learn the importance of interaction relations, subordination relations, bundle constraint modules, and multi-task learning modules in the IMBR model. We investigate the underlining mechanism of our IMBR with five ablated models, and we also research the performance comparison of models with different propagation layers ℓ on all two datasets, in which we vary ℓ from 1 to 4. As shown in Table 3, we have the following observations:

- IMBR outperforms w/o IR-B and w/o IR-I. User-bundle interaction and User-item interaction are removed in w/o IR-B and w/o IR-I, which proves that both interactions are essential in the user-view.
- IMBR outperforms w/o CR. Bundle-item containment relation is removed in w/o CR, which shows that it is helpful to mine multi-relation in item-view. Indicates that the composition of the bundle is essential
- IMBR outperforms w/o BC. In this part, we remove the BFTC algorithm and use the original unprocessed data. The experimental results show that our algorithm is effective.
- w/o ML is the least competitive. We remove $\mathcal{L}_{item}$, which shows that our multi-task learning framework is also helpful for IMBR.
- Study of ℓ-th layer. We find that its performance first improves and then drops when the layer number increases from 1 to 4. This indicates that will suffer from oversmoothing issues when higher-order neighbors are used.



(a) learn rate of NetEase (b) learn rate of Youshu

Fig. 2. Impact of learning rate on *NetEase* and *Youshu*

5.5 Hyper-Parameters Analysis

We analyze the effects of learning rate. The experimental results are shown in Fig. 2. The learning rate is selected from [1e-5, 3e-5, 1e-4, 3e-4, 1e-3, 3e-3] by applying grid search. It can be seen that the fold trend generally rises first and starts to fall after the learn rate reaches 3e-4. The *NetEase* and *Youshu* datasets achieve the best experimental results at a learning rate of 3e-4.

6 Conclusion

In this paper, we study the multi-relation problem of nodes in the bundle recommendation. We first propose a novel model IMBR which is based on different relations between different nodes of graph, and uses graph neural networks to extract different view of multi-relation from different view, thus representing users' preferences more clearly. Next, we consider the complexity of the items in the bundle and design a bundle frequent item constraint algorithm to obtain a more accurate representation of the bundle. Finally we use a multi-task learning framework to model user-items and user-bundles. The performance of bundle recommendation is further improved. Combined with experiments on two real datasets, it is demonstrated that our proposed IMBR approach outperforms existing bundle recommendation methods.

References

1. Wu, S., Sun, F., Zhang, W., Cui, B.: Graph neural networks in recommender systems: a survey. arXiv preprint arXiv:2011.02260 (2020)
2. Han, P., Wang, N., Li, K., Li, X.: CASR: a collaborative attention model for session-based recommendation. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 290–296. IEEE (2021)
3. Cao, D., Nie, L., He, X., Wei, X., Zhu, S., Chua, T.S.: Embedding factorization models for jointly recommending items and user generated lists. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 585–594 (2017)
4. Yu, W., Li, L., Xu, X., Wang, D., Wang, J., Chen, S.: ProductRec: product bundle recommendation based on user's sequential patterns in social networking service environment. In: 2017 IEEE International Conference on Web Services (ICWS), pp. 301–308. IEEE (2017)
5. Liu, Y., Xie, M., Lakshmanan, L.V.: Recommending user generated item lists. In: Proceedings of the 8th ACM Conference on Recommender Systems, pp. 185–192 (2014)
6. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
7. Chen, L., Liu, Y., He, X., Gao, L., Zheng, Z.: Matching user with item set: collaborative bundle recommendation with deep attention network. In: IJCAI, pp. 2095–2101 (2019)
8. Chang, J., Gao, C., He, X., Jin, D., Li, Y.: Bundle recommendation and generation with graph neural networks. IEEE Trans. Knowl. Data Eng. (2021)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
10. Wang, Y., Luo, Q., Ding, Y., Wang, D., Deng, H.: DemiNet: dependency-aware multi-interest network with self-supervised graph learning for click-through rate prediction. arXiv preprint arXiv:2109.12512 (2021)
11. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
12. Zhu, H., Feng, F., He, X., Wang, X., Li, Y., Zheng, K., Zhang, Y.: Bilinear graph neural network with neighbor interactions. arXiv preprint arXiv:2002.03575 (2020)

13. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 165–174 (2019)
14. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38

Information Processing and Data Management

A Privacy Preserving and Format-Checkable E-voting Scheme

Yuhong Sun(✉), Shiyu Wang, Fengyin Li, and Hua Wang

School of Computer Science, Qufu Normal University, Rizhao 276826, Shandong, China
sun_yuh@163.com

Abstract. Electronic voting (e-voting) is widely used because of its convenience and efficiency. In response to the security problems in e-voting, such as the legality of voters, privacy disclosure, etc., this paper proposes a novel e-voting scheme that can check the format of ballots without disclosing its content based on homomorphic encryption. Firstly, voters encrypt their ballots with Paillier encryption before sending them to the counter. Then, the counter decomposes the encrypted ballots using the proposed *n*-ary conversion protocol, and performs the format check of the ballots. Only ballots with the correct format are counted. During the whole process of voting, no one except the voter himself can know each ballot's content, even the counter, so that the privacy of ballots is preserved. Finally, the counter performs an additive homomorphism operation on the encrypted ballots and the voting manager decrypts it to tally the result. Besides the requirements including the legality, privacy, and integrity, we furtherly consider the validity of the ballots in e-voting and make the scheme more practical than the existing methods.

Keywords: E-voting · Format-checkable · Paillier encryption · *n*-ary conversion protocol

1 Introduction

In recent years, electronic voting (e-voting) is becoming a popular method of replacing the traditional paper voting [1]. The e-voting [2] is superior to the traditional voting in efficiency and economy, but it faces several security threats such as the voter legitimacy, ballot uniqueness, privacy preserving, etc.

The existing e-voting schemes, mostly use the cryptographic protocol to guarantee the privacy and the correctness of the voting result, and can be roughly classified into three main categories: schemes based on the mix-net [3] schemes based on blind signature [4] or ring signature [5], and schemes based on homomorphic encryption [6].

The recent work, such as Kumar et al. [7], who proposed a secure end-to-end verifiable e-voting scheme based on identity-based blind signatures, where the end-to-end verification allows the voter to check whether his ballot was recorded correctly as he intended. The shortcoming of this scheme is that the selected candidate's name appears on the ballot and the privacy of the ballot cannot be preserved well. In 2019, Yining et al.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 475–488, 2022.
https://doi.org/10.1007/978-3-031-19214-2_40

[8], proposed an e-voting scheme based on secret sharing and k-anonymity, which satisfies unconditional security, but suffers from the problem that it is hard to check whether the voter has cheating behavior. And in 2016, Shahandashti et al. [9] proposed a voting strategy named 'DRE-ip' that improved the DRE (Direct-Recording Electronic) system and can publicly verify the voting result without decrypting the ballots. The shortcoming of this system is that the voting result, even the format check by NIZK, relies on the recording party heavily. Although the blockchain technology to e-voting can increase the security and transparency of voting schemes and reduce the reliance on third-party institutions, an unavoidable fact is that they suffer from the latency due to the verification in a p2p network, such as [10, 11], etc.

In fact, whether the blind signature-based schemes or the homomorphic encryption-based schemes must meet such a contradiction: the ballot should be hidden by operation of encryption or blinding for the privacy before it is issued to the receiver, but the receiver cannot confirm whether the ballot in a correct format. Examples such as in literatures [7–14], there are not any format checks before the ballots are accepted. However, in reality, the voter may cast a ballot that includes more than one "approval" for the same candidate. The encryption or blinding operation will prevent this behavior from being discovered.

To solve the above problem, we propose a novel e-voting scheme based on Paillier encryption that can check the format of the ballot, so that the cheating behavior from voters can be revealed by the counter. Specifically, the contribution of this paper can be summarized as follows: i) We investigated the existing e-voting schemes and analyzed the necessity of format check in reality. ii) A privacy preserving and format check e-voting scheme based on Paillier encryption is proposed, that can check the format of the ballot without disclosing its content. iii) We analyzed the proposed scheme from multiple security requirements. iv) We gave the comparison of the proposed scheme and other e-voting schemes from the performance.

The rest of the paper is organized as follows. In Sect. 2, some preliminaries are introduced, and the system model is presented in Sect. 3. Section 4 presents the proposed format-checkable e-voting scheme. Section 5 provides security analysis. Finally, Sect. 6 concludes the paper.

2 Preliminaries

2.1 Paillier Cryptosystem

KeyGen. Randomly select two large prime numbers p, q, and $g \in Z_{N^2}$, let $N = p \cdot q$, $\lambda = 1cm(p-1, q-1)$, $l(u) = \frac{u-1}{N}$, $gcd\left(l(g^{\lambda} mod\ N^2), N\right) = 1$. The public key is (N, g), and the secret key is λ.

Encrypt. Randomly select integer $r \in Z_{N^2}^{*}$, for the plaintext $m \in Z_N$, the ciphertext is $c = g^m \cdot r^N mod\ N^2$.

Decrypt. The plaintext $m = \frac{l(c^{\lambda} mod\ N^2)}{l(g^{\lambda} mod\ N^2)} mod\ N$.

The Paillier cryptosystem has the additive homomorphism [15] property:

$$E(m_1) \cdot E(m_2) = (g^{m_1} \cdot r_1^N) \cdot (g^{m_2} \cdot r_2^N) = g^{m_1+m_2}(r_1 r_2)^N = E(m_1 + m_2)\ mod\ N^2$$

2.2 Boneh-Boyen Signature

In this paper, we use the Boneh-Boyen signature scheme as [16].

KeyGen. Let G, G_T be prime p order cyclic groups, g is a generator of G, and there exists a bilinear pairing e: $G \times G \to G_T$. The secret key is $x \in Z_p$, and the corresponding public key is $y = g^x$.

Sign. The signature on a message m is $\sigma = g^{1/(x+m)}$.

Verify. Verification is done by checking that $e(\sigma, y \cdot g^m) = e(g, g)$.

2.3 Secure Bit-Decomposition Protocol (SBD)

Suppose that there are two parties, Alice and Bob. Bob holds the Paillier encrypted value $E(x)$, where $0 < x \leq 2^{\mu}$ (μ is the domain size of x in bits). Let $\left(x_0, x_1, \cdots, x_{\mu-1}\right)$ denotes the binary representation of x. The goal of SBD is to convert the encryption of x into the encryptions of the individual bits of x, without disclosing any information regarding x to both parties. We use the SBD protocol in this paper as [17, 18]: $\text{SBD}(E(x)) \to (E(x_0), \cdots, E(x_{\mu-1}))$. The details of the protocol are shown in algorithm 2.

3 System Model and Notations

3.1 System Entities

The main participants of this scheme include the election commission authority (*ECA*), voters (V_is), the authentication center (*AC*), and the counting center (*CC*), The entities and their interactions are shown in Fig. 1, and their functions are described as follows.

Election commission authority (*ECA*): *ECA* initializes the system, interacts with the *CC* during the counting phase, decrypts and announces the final voting results.

Authentication center (*AC*): *AC* authenticates voters, verifies the legitimacy of each voter's identity, and issues him the unique voting identification ID_i, which is unrelated to his identity information.

Voters (V_i): A voter V_i must be authenticated to obtain a unique voting identification ID_i before participating in the voting.

Counting center (*CC*): *CC* collects the encrypted ballots, checks the format of each ballot, and counts the ballots with the correct format.

3.2 Trust Assumption

In our scheme, the trust assumptions are as follows.

1) *ECA* is assumed to be honest, it is authoritative and usually acts as the voting manager in reality.
2) *AC* and *CC* are assumed to be semi-honest, or honest but curious.
3) The voter V_i is not assumed to be honest, because he may cast a ballot consisting of more than one "approval" for the same candidate.

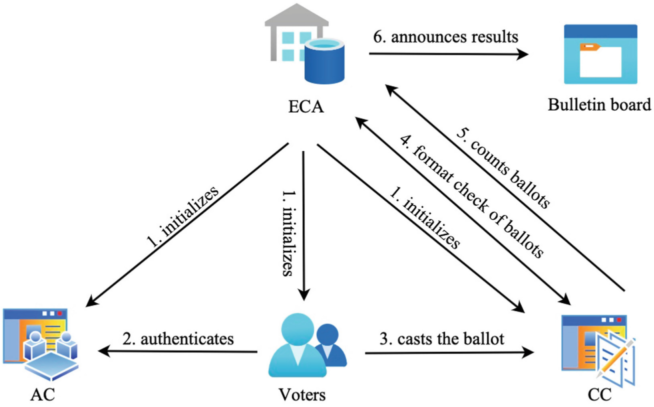


Fig. 1. System model

3.3 Notations Description

The notations used throughout the paper are listed in Table 1.

Table 1. Notations and descriptions

Notations	Descriptions
S_i	Identity information of V_i
ID_i	Unique voting identification of V_i
$\left(v_0, v_1, \ldots, v_{m-1}\right)$	V_i's ballot in plaintext for each candidate
$E(M_i)$	Paillier encryption of decimal ballot M_i
$E(M_i) \rightarrow (E(c_{m-1}), \ldots, E(c_0))$	The n-ary conversion of $E(M_i)$
SBD $(E(M_i)) \rightarrow (E(\beta_0), \ldots, E(\beta_{\mu-1}))$	The SBD conversion of $E(M_i)$
$List^{V_i}$	List of all legal voters
$List^{AC}$	List of authenticated voters
$List^{CC}$	List of voters who have cast their ballot

4 Our Scheme

4.1 Scheme Overview

Our voting scheme can be applied in the scenario of k-out-of-m (voters can choose k individuals from m candidates, $k \geq 1$). The implementation of the proposed scheme consists of four phases.

Initialization phase: Each participating entity generates its own public/private keys following the Paillier cryptosystem and the Boneh-Boyen signature system.

Authentication phase: AC authenticates each voter based on V_i's identity information S_i according to $List^{V_i}$, and issues the encrypted ID_i to the legal voter. Lastly, AC publishes authenticated voters' ID_i in the $List^{AC}$.

Voting phase: During the voting phase, V_i generates an encrypted ballot and sends it to CC. CC firstly decomposes the encrypted ballot, then it checks whether the format of the ballot is correct.

Counting phase: After format check, CC counts the encrypted ballots with the correct format by homomorphic addition and sends the result to ECA. ECA decrypts the counting result and publishes the final voting result. The CC publishes ID_i as the $List^{CC}$.

4.2 Initialization Phase

Given a security parameter κ, each participant generates its own public/secret keys, and publishes their public keys. AC holds the $List^{V_i}$ with all legal voters' identity information, and generates $List^{AC}$ with initial values null. CC also initializes $List^{CC}$ with null.

The Paillier cryptosystem is choosen for the property of additive isomorphism. Firstly, ECA selects the parameters as (p, q, g), generates its public key $(N, g)_{ECA}$ and secret key λ_{ECA}, where $N = p \cdot q, \lambda = 1cm(p-1,\ q-1)$. Similarly, V_i generates public key $(N, g)_{V_i}$ and secret key λ_{V_i}. AC generates public key $(N, g)_{AC}$ and secret key λ_{AC}.

We choose the short signature of Boneh-Boyen signature system. Firstly, V_i selects the parameters as (G, G_T, g), generates its public key y_{V_i} and secret key $x_{V_i} \in Z_p$, where $y_{V_i} = g^{x_{V_i}}$. Similarly, AC generates public key y_{AC} and secret key $x_{AC} \in Z_p$. CC generates public key y_{CC} and secret key $x_{CC} \in Z_p$.

4.3 Authentication Phase

Before voting, each voter must be authenticated. The authentication phase is listed as follows.

Step1. V_i signs his identity information S_i to generate $sig_{V_i}(S_i)$, then encrypts S_i and $sig_{V_i}(S_i)$ together using AC's public key as $E_{AC}(S_i, sig_{V_i}(S_i))$, and sends it to AC.

Step2. After receiving $E_{AC}(S_i, sig_{V_i}(S_i))$, AC decrypts it to get $(S_i, sig_{V_i}(S_i))$, then checks whether S_i is in $List^{V_i}$. If it is, AC verifies $sig_{V_i}(S_i)$. The V_i is regarded as a legal voter if $sig_{V_i}(S_i)$ is verified. Otherwise, AC rejects this authentication request.

Step3. Next, AC checks whether S_i exists in $List^{AC}$. If it is not, which means V_i is authenticated for the first time, then V_i is recorded in $List^{AC}$. Otherwise, it indicates V_i has already authenticated before, and AC rejects this request.

Step4. V_i is considered legal and eligible to vote if it is verified both in *Step2* and *Step3*. Then AC generates a unique voting identification ID_i for V_i independent of S_i to ensure the anonymity of V_i. Each voter only knows his own ID_i, and no one can associate the ID_i with his real identity S_i except AC. Then AC signs and encrypts the ID_i as $E_{V_i}(ID_i, sig_{AC}(ID_i))$, and returns it to V_i.

Step5. V_i decrypts $E_{V_i}(ID_i, sig_{AC}(ID_i))$ to get ID_i and $sig_{AC}(ID_i)$, and verifies $sig_{AC}(ID_i)$. If it is valid, V_i keeps $(ID_i, sig_{AC}(ID_i))$ as his voting certificate, and AC appends the S_i and ID_i to $List^{AC}$.

Step6. In the end, all ID_i in $List^{AC}$ are published. Voters can check whether they have been regarded as legitimate voters. If an authenticated voter cannot find his ID_i, he can apply for authentication again to AC.

4.4 Voting Phase

This scheme assumes that there are N' voters $(V_1, \cdots, V_{N'})$ and m candidates $(C_1, \cdots, C_m)$, and each voter can vote for all candidates. Firstly, the ballot is represented as an m-bit n-ary number. When the candidate $C_j (0 \leq j \leq m-1)$ is approved, the j-th bit of the ballot is 1. Otherwise, it is 0. If there are too many voters, CC can divide N' voters randomly into t groups, where each group consists of n_t voters. When CC receives a ballot from a voter, it adopts the proposed n-ary conversion protocol to decompose the encrypted ballots, where $n = n_t + 1$. Then, CC checks the format of encrypted ballots without disclosing the content. The voting phase for each group is listed as follows.

Step1. V_i $(0 \leq i \leq n_t - 1)$ creates a ballot $(v_0, v_1, \cdots, v_{m-1})$ for all candidates as his wish, and converts this ballot from n-ary to a decimal M_i, where $M_i = v_{m-1} \cdot n^0 + \cdots + v_0 \cdot n^{m-1}$. Then, he generates $E_{ECA}(M_i)$ using ECA's public key. Lastly, V_i sends $(ID_i, sig_{AC}(ID_i), E_{ECA}(M_i))$ to CC.

Step2. After receiving $(ID_i, sig_{AC}(ID_i), E_{ECA}(M_i))$, CC firstly checks whether the ID_i exists in $List^{CC}$. If it exists, it means that the voter has cast a ballot before, and CC neglects him. Otherwise, $sig_{AC}(ID_i)$ is verified by CC and if it is valid, it indicates that the V_i is an authorized voter.

Step3. If above verifications are valid, CC decomposes $E_{ECA}(M_i)$ from a decimal number to an n-ary number without decryption. The specific steps are shown in algorithm 1. In this process, the SBD protocol in algorithm 2 is invoked to decompose $E_{ECA}(M_i)$ into bits in binary, and algorithm 4 is invoked to decide whether the last bit of the data is 0 or 1 in the next operation.

Step4. After decomposing $E_{ECA}(M_i)$, CC performs the format check on the decomposed ballot $(E_{ECA}(c_{m-1}), \cdots, E_{ECA}(c_0))$. Firstly, CC inputs $(E_{ECA}(c_{m-1}), \cdots, E_{ECA}(c_0))$ and 2 into algorithm 5. If the algorithm outputs false, this means that the format of the ballot is incorrect. Then, CC rejects the ballot and sends feedback to the V_i. Otherwise, CC appends ID_i to $List^{CC}$.

Here, we give an example to show how the encrypted ballot decomposition works. Suppose there are three voters and four candidates, so $n = 3 + 1 = 4$. V_i creates a ballot (0011) according to his own will, and calculates $M_i = 1 + 1 \times 4 = 5$, then he encrypts M_i to get $E_{ECA}(5)$ using ECA's public key. After CC receives the $E_{ECA}(5)$, it executes algorithm 1.

Firstly, CC executes the SBD protocol to decompose $E_{ECA}(5)$ into $(E_{ECA}(0), E_{ECA}(1), E_{ECA}(0), E_{ECA}(1))$, which is the initial value of q for the first cycle. After the first cycle of decomposition operation in algorithm 1, CC gets $E_{ECA}(c_0) = E_{ECA}(A) = E_{ECA}(1)$ as shown in Table 2. The last set of $q = (0001)$ is used as the initial value of q for the next cycle. And similarly, CC gets $E_{ECA}(c_1) = E_{ECA}(1)$, $E_{ECA}(c_2) = E_{ECA}(0)$, $E_{ECA}(c_3) = E_{ECA}(0)$ in the next three cycles. Finally, the decomposed encrypted ballot is $(E_{ECA}(c_3), E_{ECA}(c_2), E_{ECA}(c_1), E_{ECA}(c_0)) = (E_{ECA}(0), E_{ECA}(0), E_{ECA}(1), E_{ECA}(1))$.

Algorithm 1 *n*-ary Conversion

```
**Input**: $E_{ECA}(M_i)$, $m$ ($m \leq \mu$, $\mu$ is the number of bits in the binary format of $M_i$)
**Output**: $(E_{ECA}\left(c_{m\text{-}1}\right), \cdots, E_{ECA}(c_0))$
@CC:
executes SBD
$(E_{ECA}(\beta_0), \cdots, E_{ECA}(\beta_{\mu\text{-}1})) \leftarrow \mathrm{SBD}(E_{ECA}(M_i))$
marks $(E_{ECA}(q_0), \cdots, E_{ECA}(q_{\mu\text{-}1})) \leftarrow (E_{ECA}(\beta_0), \cdots, E_{ECA}(\beta_{\mu\text{-}1}))$
initializes $(E_{ECA}(c_{m\text{-}1}), \cdots, E_{ECA}(c_0)) \leftarrow (E_{ECA}(0), \cdots, E_{ECA}(0))$
for $j$=0 to $m$-1 do
  for executing $\mu$ times do (for $j$=0 to $\mu$-1)
  initializes $(E_{ECA}(a_0), \cdots, E_{ECA}(a_{\mu\text{-}1})) \leftarrow (E_{ECA}(0), \cdots, E_{ECA}(0))$
  $E_{ECA}(a_{j\text{-}1}) = E_{ECA}(a_j)$
  $E_{ECA}(a_{\mu\text{-}1}) = E_{ECA}(q_0)$
  $E_{ECA}(q_{j\text{-}1}) = E_{ECA}(q_j)$
  calculates $E_{ECA}(A) = E_{ECA}(a_{\mu\text{-}1}) \cdot 2^0 + \cdots + E_{ECA}(a_0) \cdot 2^{\mu\text{-}1}$
  executes ComparePro $(E_{ECA}(A), n)$
  calculates $E_{ECA}(q_{\mu\text{-}1}) = E_{ECA}(1) \cdot E_{ECA}(Q)^{N\text{-}1} = E_{ECA}(1\text{-}Q)$
  $E_{ECA}(B) = E_{ECA}(\text{-}q_{\mu\text{-}1})^n = E_{ECA}(\text{-}nq_{\mu\text{-}1})$
  $E_{ECA}(A) = E_{ECA}(A) \cdot E_{ECA}(B) = E_{ECA}(A+B)$
  end for
$E_{ECA}(c_j) = E_{ECA}(A)$
end for
```

Algorithm 2 SBD ($E_{ECA}(M_i)) \rightarrow E_{ECA}(\beta_0), \cdots, E_{ECA}(\beta_{\mu-1})$)

```
Input: $E_{ECA}(M_i)$
Output: $(E_{ECA}(\beta_0), \cdots, E_{ECA}(\beta_{\mu-1}))$
$L \leftarrow 2^{-1}$
$T \leftarrow E_{ECA}(M_i)$
for $\delta=0$ to $\mu-1$ do
    @CC:
        $Y \leftarrow T \cdot E_{ECA}(b), b \in Z_N$
        sends $Y$ to $ECA$
    @ECA:
        receives $Y$ from $CC$
        $d \leftarrow D_{ECA}(Y)$
        if $d$ is even $\alpha \leftarrow E_{ECA}(0)$
        else $\alpha \leftarrow E_{ECA}(1)$
        sends $\alpha$ to $CC$
    @CC:
        receives $\alpha$ from $ECA$
        if $\alpha$ is even $E_{ECA}(\beta_\delta) \leftarrow \alpha$
        else $E_{ECA}(\beta_\delta) \leftarrow E_{ECA}(1) \cdot \alpha^{N-1}$
        return $E_{ECA}(\beta_\delta)$
    $Z \leftarrow T \cdot E_{ECA}(\beta_\delta)^{N-1}$
    $T \leftarrow Z^L$ (update $T$ with the encrypted value of current quotient in algorithm 3)
end for
```

Algorithm 3 Binary($M_i) \rightarrow (\beta_0, \cdots, \beta_{\mu-1}$)

```
Input: $M_i$
Output: $(\beta_0, \cdots, \beta_{\mu-1})$
for $\delta=0$ to $\mu-1$ do
    $\beta_\delta \leftarrow M_i \bmod 2$ ($\beta_\delta$ is updated to current quotient)
    $M_i \leftarrow \left\lfloor \frac{M_i}{2} \right\rfloor$
end for
```

Algorithm 4 ComparePro

```
Input: $E_{ECA}(A)$, $n$
Output: $Q$
1. @CC:
(1) encrypts $n$ to get $E_{ECA}(n)$
(2) calculates $E_{ECA}(S)=E_{ECA}(A)\cdot E_{ECA}(n)^{N-1}=E_{ECA}(A-n)$
(3) chooses two random numbers $r_1$, $r_2$, $r_1>r_2$, $L(r_1)<L(N)/8$
(4) encrypts $r_2$ to get $E_{ECA}(r_2)$
(5) flips a coin $c$ randomly
    if $c$=0, calculates $E_{ECA}(R)=E_{ECA}(r_2)\cdot E_{ECA}(S)^{r_1}=E_{ECA}(r_2+r_1S)$
    if $c$=1, calculates $E_{ECA}(R)=E_{ECA}(r_2)\cdot E_{ECA}(S)^{N-r_1}=E_{ECA}(r_2-r_1S)$
(6) sends $E_{ECA}(R)$ to $ECA$
2. @ECA:
(1) receives $E_{ECA}(R)$ from $CC$
(2) decrypts $E_{ECA}(R)$ to get $R$
(3) if $R>\frac{N}{2}$, $u$=1
    else $u$=0
(4) sends $u$ to $CC$
3. @CC:
(1) receives $u$ from $ECA$
(2) if $c$=0, calculates $Q=u$
    if $c$=1, calculates $Q=1-u$
      return $Q$    (if $Q$=0, it shows $S\geq 0$, $A\geq n$, if $Q$=1, it shows $S<0$, $A<n$)
```

Algorithm 5 FormatCheck

```
Input: $(E_{ECA}(c_{m-1}), \cdots, E_{ECA}(c_0))$, 2
Output: True or False
for $j$=0 to $m$-1 do
  $z\leftarrow$ ComparePro $(E_{ECA}(c_j), 2)$
  if $z$=0 break
    return False
end for
if $j=m$-1
  return True
```

Table 2. Example of n-ary conversion ($j = 0$)

Round	a	q	n	Operations
1	0000 0000	0101 1010	4	Shift left a and q together $A \leftarrow 0 < n, Q \leftarrow 1, q_{\mu-1} \leftarrow 0$ $B \leftarrow -n \cdot q_{\mu-1} \leftarrow 0, A \leftarrow A + B \leftarrow 0$
2	$\underline{0000}$ 0001	$101\underline{0}$ 0100	4	Shift left a and q together $A \leftarrow 1 < n, Q \leftarrow 1, q_{\mu-1} \leftarrow 0$ $B \leftarrow -n \cdot q_{\mu-1} \leftarrow 0, A \leftarrow A + B \leftarrow 1$
3	$\underline{0001}$ 0010	$010\underline{0}$ 1000	4	Shift left a and q together $A \leftarrow 2 < n, Q \leftarrow 1, q_{\mu-1} \leftarrow 0$ $B \leftarrow -n \cdot q_{\mu-1} \leftarrow 0, A \leftarrow A + B \leftarrow 2$
4	$\underline{0010}$ 0101 $\underline{0001}$	$100\underline{0}$ 0000 $000\underline{1}$	4	Shift left a and q together $A \leftarrow 5 > n, Q \leftarrow 0, q_{\mu-1} \leftarrow 1$ $B \leftarrow -n \cdot q_{\mu-1} \leftarrow -4, A \leftarrow A + B \leftarrow 1$

4.5 Counting Phase

After voting, *CC* can obtain t groups, where each group consists of n_t encrypted ballots $E_{ECA}(M_i)$. Then, *CC* performs homomorphic addition on ballots for each group. Finally, *ECA* decrypts the counting result and announces the final voting result. The counting phase is listed as follows.

Step1. CC calculates each group of ballots homomorphically to obtain $E_{ECA}(M_\varepsilon) = \prod_{i=1}^{n_t} E_{ECA}(M_i)$, where $(1 \le \varepsilon \le t)$.

Step2. For each group, *CC* signs $E_{ECA}(M_\varepsilon)$ to generate $sig_{CC}(E_{ECA}(M_\varepsilon))$, and sends $(E_{ECA}(M_\varepsilon),\ sig_{CC}(E_{ECA}(M_\varepsilon))$ to *ECA*.

Step3. ECA verifies $sig_{CC}(E_{ECA}(M_\varepsilon))$, decrypts $E_{ECA}(M_\varepsilon)$ to get the voting result M_ε, and converts M_ε to the n-ary number, where each digit is the election result for each candidate in ε-th group. Then, *ECA* aggregates the result of each group and announces the results on a bulletin board.

Step4. In the end of counting phase, $List^{CC}$ is published. V_i can check whether his ID_i exists in $List^{CC}$. If V_i cannot find his ID_i, he can reapply to *CC*.

4.6 Complexity

The complexity in the presented scheme is mainly on the computation cost and communication cost. Since the initialization phase and authentication phase are preparation for the voting, we consider the costs during the voting and the counting phases. Firstly, a voter creates his ballot by one encryption. Then, the *CC* checks the legality of the voter by signature and needs one verification. The *CC* runs algorithm1 to decompose the encrypted ballot. In the process, algorithm 2 is called to convert the encrypted ballot to binary, where μ times of loops for the multiplication and addition and two communications between *CC* and *ECA* are needed. Then, algorithm 1 calls algorithm 4 to decide the last bit of the data in the next operation. Algorithm 4 needs two encryptions and one decryption with two communications between *CC* and *ECA*. By the all, algorithm 1 will be completed with the complexity of $O(\mu + m^{*}\mu^{*}1) = O(m\mu)$ encryptions and decryptions and $2m\mu$ communications, where m is the number of candidates and μ is the bit length of each ballot. Next, algorithm 5 is used to check the format of ballot by revoking algorithm 4 in m times, and the complexity of the algorithm 5 is O(m). Last, the *CC* calculates the encrypted ballot results by multiplying them only by *N'* times.

5 Scheme Analysis

5.1 Security Analysis

Correctness. Each voter creates his ballot as an n-ary number and converts it to a decimal number before encryption. The counter uses the n-ary conversion protocol to decompose the ballots and obtains each encrypted digit of the n-ary number. Since $n = n_t + 1$, the addition on the n_t ballots cannot produce any carry-over, so *CC* only needs to do the homomorphic addition once on the encrypted ballots for one group. The *AC* can decrypt and aggregate the voting result to get the final result.

Format Check. The *CC* adopts the ComparePro protocol to check the correctness of the format of the encrypted ballot. If a voter casts a ballot that contains more than one "approval" for the same candidate, the ComparePro protocol can detect the incorrect ballot, and *CC* rejects the ballot. This solution overcomes the lack of format correctness checking in existing voting systems, and avoids the possibility of fraud by voters.

Privacy. Privacy includes the privacy of the voter's identity and the privacy of the ballot. In the authentication phase, if a voter is authenticated by *AC*, he can get a voting identification ID_i that is unrelated to his identity. *AC* can only confirm the legitimacy of the voter, but cannot know who he is. During the voting phase, voters encrypt their ballots, so that no one can know the contents of the ballots or link them to the real identities of voters. In the format checking, the *ECA* and *CC* are assumed not to be colluded, they also cannot know the plaintext of the ballot. The privacy of the voter and the ballot content can be preserved.

Legitimacy. The *AC* will authenticate the voter's identity. Only if the voter's identity information exists in the $List^{V_i}$, the voter will be regarded as a legal voter, and will be issued a unique voting identification ID_i unrelated to his identity. So no illegal person can obtain the authorization.

Uniqueness. During the authentication phase, only the voter who has been authenticated by the *AC* can obtain the unique ID_i, and the $List^{AC}$ can guarantee that the voter cannot be authenticated repeatedly. Throughout the voting process, the voter cannot vote repeatedly, the $List^{CC}$ can check if the voter has voted before.

Fairness. Throughout the scheme, the ballot of each voter can only be known by himself, and no one including the *ECA*, *AC* or *CC* can know the content of the ballot. Any intermediate voting result cannot be revealed during the scheme until the final results are published.

Integrity. Throughout the voting, all authentication information and counting information are verified and counted by the *AC* and *CC* respectively. The relevant information is published in the $List^{AC}$ and $List^{CC}$. Each voter can check whether his information is correctly recorded in the above lists according to his ID_i.

Accuracy. Each voter has to pass the authentication of the *AC* before casting a ballot, and the *CC* only receives ballots from the legal voters. Therefore, the final counting result is legal, and the addictive homomorphism ensures the correctness of the final counting results.

5.2 Performance Comparison

In this section, the scheme is mainly analyzed in comparison with the schemes proposed in literature [7–12]. The literature [7] proposes an end-to-end verifiable scheme using identity-based blind signature. As mentioned in Sect. 1, the scheme cannot preserve the privacy of each voting, and voters can only vote for one candidate at a time. The e-voting scheme based on secret sharing in [8] lacks a rigorous authentication process and format check for ballots. The DRE-ip based voting system such as [9], depends on the recording machine. The blockchain based voting scheme such as [10] cannot be used to vote for multiple candidates, besides the latency in verification. The e-voting scheme in [11] and [12] also lacks of the format check for ballots, besides the voters' identities authentication and ballots' uniqueness. The comparison results are shown in Table 3.

Table 3. Performance comparison of different e-voting schemes

Scheme	Legitimacy	Privacy	Uniqueness	Format check	Multi choice
[7]	√	$\not\surd$	√	×	×
[8]	$\not\surd$	√	√	×	√
[9]	√	√	√	×	√
[10]	√	√	√	×	×
[11]	√	√	√	×	√
[12]	√	√	×	×	√
Ours	√	√	√	√	√

6 Conclusion

In this paper, we propose a privacy-preserving e-voting scheme that can check the format of the ballot while protecting the privacy of the content. The scheme implements a *k*-out-of-*m* choice for candidates. Voters generate ballots according to their wishes and encrypt the ballots by Pallier encryption. The counting center uses the *n*-ary conversion protocol and the ComparePro protocol to check the formats of ballots, and only those ballots that pass the check will be counted. The counting result is obtained through only one additive homomorphic operation by the counting center. On the one hand, this scheme does not disclose the privacy of voters and ballots. On the other hand, it guarantees the correct formats of the ballots. This scheme satisfies the basic properties of privacy, legality, accuracy, and other requirements of e-voting systems. It is more useful and practical than existing e-voting systems. However, in the proposed scheme, it is the counter center, and not the voter, undertakes more computation cost for the format check. It may result the inefficiency with too many voters in reality and this is the future work of this paper.

References

1. Pu, H.Q., Cui, Z., Liu, T.: A review of research on secure e-voting schemes. Comput. Sci. **47**(9), 8 (2020)
2. Mursi, M.F.M., Assassa, G.M.R., Abdelhafez, A.: On the development of electronic voting: a survey. Int. J. Comput. Appl. **61**(16), 1–11 (2013)
3. Peng, K.: A general and efficient countermeasure to relation attacks in mix-based e-voting. Int. J. Inf. Secur. **10**(1), 49–60 (2011)
4. Ku, W.C., Wang, S.D.: A secure and practical electronic voting scheme. Comput. Commun. **22**(3), 279–286 (1999)
5. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_32
6. He, Q., Shen, W.: Homomorphic encryption-based multi-candidate e-voting scheme. Comput. Syst. Appl. **28**(2), 146–151 (2019)
7. Kumar, M., Chand, S., Katti, C.P.: A secure end-to-end verifiable internet-voting system using identity-based blind signature. IEEE Syst. J. **14**(2), 2032–2041 (2020)
8. Liu, Y., Zhao, Q.: E-voting scheme using secret sharing and K-anonmity. World Wide Web **22**(4), 1657–1667 (2019)
9. Shahandashti, S.F., Hao, F.: DRE-ip: a verifiable E-voting scheme without tallying authorities. In: Askoxylakis, I., Ioannidis, S., Katsikas, S., Meadows, C. (eds.) ESORICS 2016. LNCS, vol. 9879, pp. 223–240. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45741-3_12
10. McCorry, P., Shahandashti, S.F., Hao, F.: A smart contract for boardroom voting with maximum voter privacy. In: Kiayias, A. (ed.) FC 2017. LNCS, vol. 10322, pp. 357–375. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70972-7_20
11. Chaieb, M., Koscina, M., Yousfi, S., Lafourcade, P., Robbana, R.: Dabsters: a privacy preserving e-voting protocol for permissioned blockchain. In: Hierons, R.M., Mosbah, M. (eds.) ICTAC 2019. LNCS, vol. 11884, pp. 292–312. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32505-3_17
12. Waheed, A., Din, N., Umar, A.I.: Novel blind signcryption scheme for e-voting system based on elliptic curves. Mehran Univ. Res. J. Eng. Technol. **40**(2), 314–322 (2021)
13. Alam, K., Tamura, S., Rahman, S.: An electronic voting scheme based on revised-SVRM and confirmation numbers. IEEE Trans. Dependable Secure Comput. **99**, 400–410 (2019)
14. Ajish, S., Anilkumar, K.S.: Secure mobile internet voting system using biometric authentication and wavelet based AES. J. Inf. Secur. Appl. **61**(14), 102908 (2021)

15. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) Advances in Cryptology, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
16. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_4
17. Samanthula, B.K.K., Chun, H., Jiang, W.: An efficient and probabilistic secure bit-decomposition. InL ACM SIGSAC Symposium on Information, pp. 541–546. ACM (2013)
18. Liu, X., Deng, R.H., Choo, K.: An efficient privacy-preserving outsourced calculation toolkit with multiple keys. IEEE Trans. Inf. Forensics Secur. **11**(11), 2401–2414 (2016)

LogLR: A Log Anomaly Detection Method Based on Logical Reasoning

Kehan Zhang[1,2], Xiaoqiang Di[1,2,3](✉), Xu Liu[1,2], Bo Li[1,2], Luyue Fang[1,2], Yiping Qin[1,2], and Jinhui Cao[1,2]

[1] School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, China
dixiaoqiang@cust.edu.cn

[2] Jilin Province Key Laboratory of Network and Information Security, Changchun, China

[3] Information Center, Changchun University of Science and Technology, Changchun 130022, China

Abstract. Logs are widespread in large and complex software-intensive systems. Log-based anomaly detection is used for system diagnosis and troubleshooting. Existing methods extract log sequences as temporal log vectors, preserving the timing information between logs. However, they lack a reasoning mechanism, which prevents the model from mining the logical relationship between logs and loses the logical association between logs. In this paper, we propose LogLR, a log anomaly detection method based on logical reasoning. LogLR extracts the logical relationship between temporal log vectors and improves detection accuracy by combining Logical Tensor Network (LTN) with LSTM. In order to overcome the problem of ignoring the logical relationship between logs in existing statistical methods for data annotation. LogLR uses LTN to capture the logical relationship between log sequences and obtains weak labels to train an LSTM model through the weak label estimation method, which saves time costs. We evaluate LogLR on two widely used public datasets and the results demonstrate the effectiveness of LogLR.

Keywords: Log anomaly detection · LTN · LSTM · Temporal log vectors · Weak label estimation

1 Introduction

Logs are important information that records system behavior. As more and more services appear, many attack behaviors and abnormal states of the system also

This work was supported in part by the Science and Technology Development Plan Project, Jilin, China, under Grant 20190302070GX, and in part by the Education Department of Jilin Province under Grant JJKH20220773KJ, and Cernet Next Generation IT Innovation project under Grant NGIICS20190503, and the National Natural Science Foundation of China under grant No. U21A20451.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 489–500, 2022.
https://doi.org/10.1007/978-3-031-19214-2_41

increase. The log records the information generated when the system is running, and analyzing the log can help the system administrator to find the abnormal behavior of the system. An accurate and efficient anomaly detection method is the key to maintaining the normal operation of the system.

A structured log is called a log event, and multiple log events within a period of time are called a log sequence. The log sequence could reflect the order of task execution. Early PCA [24], IM [12], DT [8] and LogCluster [10], methods methods detect log sequence anomalies. Among them, DT [8] uses event count vectors and their labels to build decision trees. While achieving commendable detection results, the method relies on labeled data. In contrast, LogCluster [10] performs anomaly detection through clustering of unlabeled data, which gets rid of the time cost of obtaining labeled data, but the detection result is lower than that of supervised learning methods. PLELog [25] proposes a semi-supervised anomaly detection method, which saves time cost while ensuring detection accuracy. However, existing semi-supervised methods based on statistical methods ignore the logical relationship between logs, resulting in a high error rate of data annotation.

Methods in the field of natural language processing (NLP) extract timing information in time series, and since log sequences are time series, many methods in the field of NLP are used for anomaly detection. DeepLog extracts the timing information between log events by inputting each log event into LSTM Cell at different time steps. PLELog uses the GRU model to observe the temporal log vectors of log sequences and detect anomalies by binary classification. These methods improve detection accuracy by obtaining timing information of log sequences. Although the existing log-based anomaly detection models are effective, they lack an inference mechanism, which leads to the loss of the logical relationship between log sequences and the reduction of detection accuracy.

To overcome the above challenges, this paper proposes LogLR, a log anomaly detection method based on logical reasoning. LogLR adds a reasoning mechanism by introducing LTN, captures the logical relationship between log sequences, and uses weak labels to assign probability values to log sequences, which not only saves time, but also maintains the effectiveness of supervised learning.

The main contributions of this paper are as follows:

1) We point out the problem that the existing data annotation methods based on statistical methods cannot extract the logical relationship between log sequences, and extract the logical information of log sequences through the weak label estimation method, which improves the accuracy of data annotation.
2) We propose LogLR, a log anomaly detection method based on logical reasoning. LogLR introduces a reasoning mechanism, and simultaneously extracts the timing information and logical information between log sequences for the first time, which improves the detection accuracy.
3) We evaluate the effectiveness of LogLR on two publicly available datasets, and the results confirm that our method outperforms existing state-of-the-art methods.

2 Related Work

Supervised Learning: Supervised learning methods use labeled data to assist in anomaly detection, so supervised learning methods perform well in detecting anomalies. Statistical models such as LR [21], DT [8], SVM [9], etc. are widely used in classification tasks and are trained using event count vectors and their labels to distinguish normal and abnormal log events. Inspired by SVM [9], methods such as OC-SVM [18], SVDD [19], etc. obtain spherical boundaries around the dataset to distinguish normal and abnormal log events. Considering the temporal relationship between log events, many RNN-based methods are used to extract temporal information between log events. LogRobust [26] extracts the semantic information of log events and detects anomalies using an attention-based Bi-LSTM model, capturing the contextual information of log sequences. OC4Seq [20] jointly detects anomalies using a local representative RNN model and a global representative RNN model, focusing on local and global information in the sequence, respectively. Methods [6,22] use the inference mechanism of Bayesian network for anomaly detection, LogGAN [23] uses GAN network to infer data to infer similar data. However, there is currently no method to effectively combine the inference mechanism with the temporal characteristics of log sequences. We propose a logical reasoning log anomaly detection method LogLR, which effectively combines the reasoning mechanism and the time-series characteristics of log sequences to improve the detection accuracy.

Unsupervised Learning and Semi-supervised Learning: Unsupervised learning and semi-supervised learning methods use unlabeled or a small amount of labeled data to assist in anomaly detection, which is more in line with practical application production environments. Methods such as PCA [24], IM [12], and LogCluster [10] perform anomaly detection by mining the similarity or linear relationship between data of the same category. Different from the widely used TFIDF [17] method, LogClass [14] proposes a new feature representation method, TFILF, and verifies the effectiveness of this method using classical machine learning methods. DeepLog [3] uses an LSTM model to preserve timing information between log events and detect anomalies when log patterns deviate from models trained under normal log execution. LogAnomaly [15] combines sequential and quantitative detection for the first time to improve detection performance. PLELog [25] uses some labeled data to label the remaining training data through probabilistic label estimation, using only a small amount of labeled data to take advantage of supervised learning. In contrast, LogLR adopts the weak label estimation method based on logical reasoning, and applies LTN to data annotation, which improves the accuracy of data annotation.

3 Methodology

In order to overcome the problem of the lack of reasoning mechanism in existing anomaly detection methods, which prevents the model from mining the logical relationship between logs. We propose LogLR, a log anomaly detection method based on logical reasoning. Figure 1 shows the overview of LogLR. LogLR consists of the following four parts: log parsing, vectorization, weak label estimation and anomaly detection.

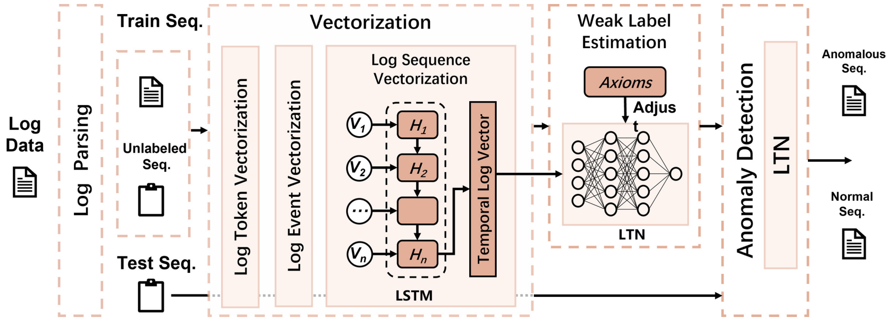


Fig. 1. Overview of LogLR

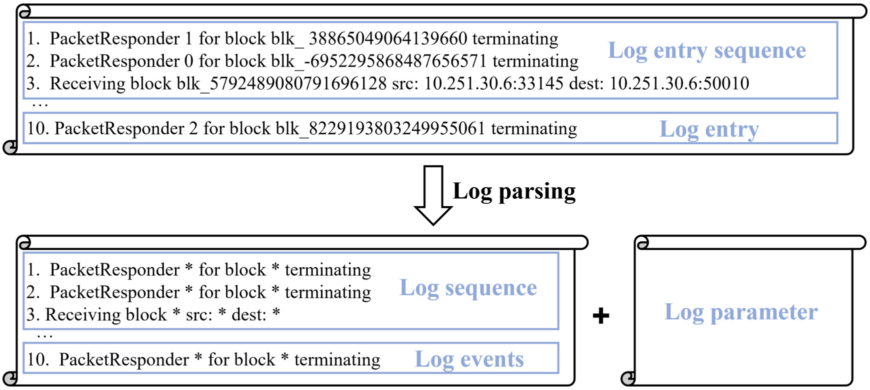
3.1 Log Parsing

Since logs are unstructured data, they contain a lot of special information (e.g., IP addresses, file names, etc.) that prevents the model from automatically detecting. It is necessary to extract this special information before using the raw logs as input to an anomaly detection model. We call the processed raw logs log events, and the step of extracting special new ones is log parsing. In this paper, we use Drain [7], which can parse logs in a streaming and timely manner. To accelerate the parsing process, Drain uses a fixed depth parse tree, which encodes specially designed rules for parsing. For example, in Fig. 2, the first log entry *"Receiving block blk_5792489080791696128 src: 10.251.30.6:33145 dest: 10.251.30.6:50010"* is parsed into the log event *"Receiving block * src: * dest: *"*. Through log parsing, unstructured raw log are transformed into structured log events.

3.2 Vectorization

The vectorization step converts the structured log events into digital vectors. Since the anomaly detection model requires an input of numeric vectors, log events need to be vectorized before being fed into the anomaly detection model. It consists of three parts: log token vectorization, log event vectorization and log sequence vectorization.

Log Token Vectorization. Treat log events as natural language sentences, each word in the sentence is called a log token, and the context between log tokens can better describe the sentence. To extract semantic information between log tokens, LogLR first splits matching words in log events into separate words according to Camel Case [2], and removes non-character tokens and stop words in



```
1.  PacketResponder 1 for block blk_ 38865049064139660 terminating
2.  PacketResponder 0 for block blk_-6952295868487656571 terminating
3.  Receiving block blk_5792489080791696128 src: 10.251.30.6:33145 dest: 10.251.30.6:50010
 …
10. PacketResponder 2 for block blk_8229193803249955061 terminating
```

```
1.  PacketResponder * for block * terminating
2.  PacketResponder * for block * terminating
3. Receiving block * src: * dest: *
 …
10.  PacketResponder * for block * terminating
```

Fig. 2. Overview of log parsing

log events to preprocess log events. LogLR then uses the FastText algorithm [16] to vectorize each log token in the log event. FastText performs word vectorization through the context of each log token to obtain a log token vector. After the log token vectorization, each log token is converted into a fixed-dimensional vector.

Log Event Vectorization. To extract semantic information between log events, LogLR performs weighted summation of the token vector in the log event to obtain the log event vector, The log event vector V be calculated by Eq. 1:

$$V = \frac{1}{N}\sum_{i=1}^{N} w_i \cdot v_i \tag{1}$$

where N is the number of log tokens in the log event, v_i is the log token vector, and w_i is the weight of each log token.

LogLR uses TF-IDF, a weighting technique commonly used in information retrieval and data mining, to calculate the weight w_i for each log token, where TF is the term frequency and IDF is the inverse text frequency index. TF is the frequency of occurrence of each log token in log events, calculated as $\frac{\#w}{\#N}$, and IDF is a measure of the general importance of a word, calculated as $log(\frac{\#L}{\#L_w})$, where #w is the number of log token w in log events, #N is the total number of log tokens in log events, #L is the total number of different log events, and $\#L_w$ is the log containing log token w number of events. The weight ω is calculated as $TF \times IDF$. Through weighted summation, LogLR obtains a log event vector containing semantic information.

Log Sequence Vectorization. After obtaining the log event vector containing semantic information, LogLR uses LSTM to solve the problem of gradient disappearance and explosion during long-sequence training, and extract the log sequence vector. Figure 3 shows the overview of An LSTM Cell.

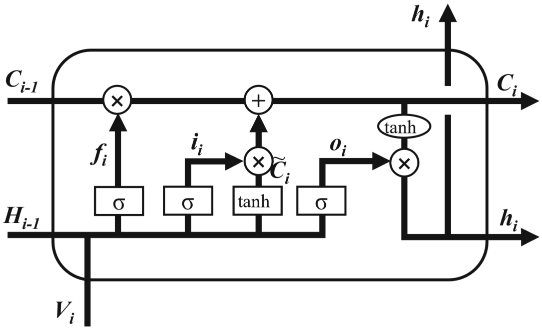


Fig. 3. Overview of An LSTM Cell

LSTM uses the gating unit to combine the LSTM state of the previous time step with the input data of this time step to generate the LSTM state of this time step. The gating unit is calculated as the Eq. 2:

$$\begin{aligned} f_t &= \delta(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \delta(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \widetilde{C}_t &= tanh(W_c \cdot [h_{t-1}, x] + b_c) \\ o_t &= \delta(W_o \cdot [h_{t-1}, x_t] + b_o) \end{aligned} \tag{2}$$

The LSTM state at this time step is calculated as Eq. 3:

$$\begin{aligned} C_t &= f_t * C_{t-1} + i_t * \widetilde{C}_t \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{3}$$

In order to extract the timing information of the log sequence, LogLR connects multiple LSTM Cell, inputs the log events in the log sequence into different LSTM Cell in turn, and uses the final hidden h_t as the log sequence vector of the sequence, which is called the temporal log vector.

3.3 Weak Label Estimation

After log-vectorization, LogLR uses LTN [1], a framework that combines tensor networks with first-order multivalued logical inference, to label unlabeled data. The structure of LTN is shown in Fig. 4.

Some objects are associated with a set of quantitative properties, represented by a real-valued n-tuple $G(o_i) \in R_n$, which we call grounding, where o_i belongs to an infinite set of objects $O = \{o_1, o_2, ...\}$. LogLR uses the vectorization process as the ground, x_+ are the normal examples, x_- are the abnormal examples input into $G(A|\theta) : x \rightarrow sigmoid(MLP(x))$, where MLP is a multilayer perceptron with one output neuron whose parameter θ needs to be learned. Through $G_\theta(A)$, LogLR obtains a probability value as the label of the input example, and labels

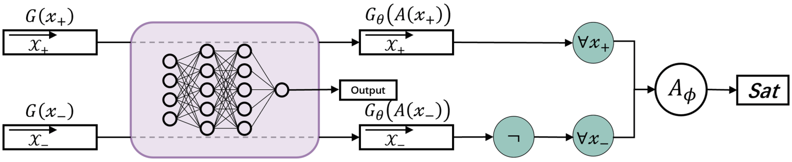


Fig. 4. Overview of LTN

weak labels for the examples at the boundary of normal examples and abnormal examples, reducing the impact of data annotation errors on the model.

LTN introduces the inference mechanism by setting the axioms, and in the back-propagation stage, the model parameters are adjusted by setting the loss function using the axioms. The axioms are set as shown in Eq. 4:

$$\begin{gathered} \forall x_+ A(x_+) \\ \forall x_- \neg A(x_-) \end{gathered} \tag{4}$$

K is a set of closed first-order logic formulas. The objective function with $K = \{\forall x_+ A(x_+), \forall x_- \neg A(x_-)\}$ is denoted as $SatAgg_{\phi \in K} G_{\theta, x \leftarrow D}(\phi)$. The value of the objective function represents the satisfaction of the knowledge base and the confidence that all examples are correctly classified. The loss function is calculated as 1 minus the value of the objective function. The objective function of LTN is calculated as Eq. 5:

$$\begin{aligned} & \underset{\phi \in K}{SatAgg} G_\theta(\phi) = \\ & 1 - \frac{1}{2}(1 - (1 - (\frac{1}{|G(x_+)|} \sum_{v \in G(x_+)} (1 - sigmoid(MLP_\theta(v)))^2)^{\frac{1}{2} \cdot 2}) \\ & + 1 - (1 - (\frac{1}{|G(x_-)|} \sum_{v \in G(x_-)} (sigmoid(MLP_\theta(v)))^2)^{\frac{1}{2} \cdot 2}))^{\frac{1}{2}} \end{aligned} \tag{5}$$

The notation $G_{x \leftarrow D}(\phi(x))$ means that the variable x is grounded with the data D when grounding $\phi(x)$.

In the weak label estimation stage, LogLR obtains a true value for the input sample in the interval [0, 1] as the label value of the unlabeled data. The samples at the classification boundary are easily mislabeled, and directly classifying the samples with a large label error rate will change the distribution of the samples. LogLR uses probability values as weak labels to increase training data and improve the detection accuracy of the model without changing the overall distribution of samples as much as possible.

3.4 Anomaly Detection

In the anomaly detection step, we use the session as the basic unit of classification of the anomaly detection model. A session is a process of information exchange between a client and a server. A session is established within a period of time, during which multiple information transfers are involved. We use two hyperparameters to divide sessions into log sequences, which are fed into the log anomaly detection model. A session is considered normal when all log sequences in the session are classified as normal by the anomaly detection model, but is considered abnormal when at least one log sequence in it is detected as abnormal.

LogLR detects anomalies using the LTN detection model. After the weak label estimation stage, LogLR retrains the LTN model with weak labels. LogLR uses log sequences marked with 0 or 1, where 0 indicates that the log sequence is abnormal and 1 indicates that the log sequence is normal. Different from the weak label estimation stage, the anomaly detection stage compares the true value between [0, 1] obtained by the LTN model with the preset threshold. LogLR detects the log sequence as normal when the true value of the output is greater than the threshold, otherwise it is detected as abnormal.

Overall, LogLR processes unstructured logs and converts them into structured log events. Secondly, construct the log event sequence, and extract the timing information and logical information in the log sequence and convert it into a log vector. Then, the training data is increased by the weak label estimation method, and finally the LTN model is used to extract the logical relationship between log sequences to improve the accuracy of anomaly detection.

4 Evaluation

4.1 Datasets

We evaluate our approach on two publicly available log datasets, including the HDFS dataset and the BGL dataset.

HDFS dataset: It is generated through running Hadoop-based map-reduce jobs on more than 200 Amazon's EC2 nodes, and labeled by Hadoop domain experts [3]. HDFS dataset has 11197954 log entries, according to the identifiers, the log sequences are divided into 575061 identifiers. Each identifier is annotated by domain experts. Among them, 4855 normal log sequences and 1638 abnormal sequences are selected as the training dataset, and the rest are used as the test dataset for testing.

BGL dataset: It is generated by the Blue Gene/L supercomputer, which consisted of 128K processors and was deployed at Lawrence Livermore National Laboratory(LLNL) [15]. BGL dataset has 4747963 log entries, each log entry is labeled by domain experts as normal or abnormal, and 348460 logs are labeled as abnormal. Divide log entries into log sequences, 44054 normal log sequences and 4050 abnormal sequences are selected as the training dataset, and the rest are used as the test dataset for testing.

To execute anomaly detection approaches, we group log entries into different sessions by an identifier field which for HDFS log is block_id and for BGL log is the sliding window. We divide each dataset into training, validation, and test sets with a ratio of 6:1:3 to evaluate the performance of log-based anomaly detection methods. To evaluate the annotation accuracy of the semi-supervised method LogLR, we sample 50% of the training data as known log sequences and the remaining log sequences in the training data as unlabeled log sequences to simulate a semi-supervised scenario.

4.2 Measurements

In this paper, we use Precision, Recall and F1-score scores to measure the effectiveness of abnormal detection based on log-based abnormal detection. Precision, Recall and F1-score is calculated as $\frac{TP}{TP+FP}$, $\frac{TP}{TP+FP}$, $\frac{2\cdot(Precision\cdot Recall)}{Precision+Recall}$, where TP, FP, and FN refer to the number of true positives(An abnormal log sequence is detected as an abnormal sequence), false positives(A normal log sequence is detected as an abnormal sequence), and false negatives(An abnormal log sequence is detected as a normal sequence), respectively.

4.3 Results and Analysis

Comparison with Statistical Methods. Figure 5 shows the superiority of using LTN for data annotation. Compared with statistical methods, LTN gradually regulates the classification boundary between normal and abnormal log sequences by automatic learning, and has improved the accuracy of annotation through the reasoning mechanism to reason log sequences. The experimental results show that the accuracy of using LTN for data annotation reaches 97.1%, which is higher than the existing statistical method PCA [4], K-Means [11], MST [5] and HDBSCAN [13]. LogLR uses weak label estimation method to provide a probability value for the log sequence labeled by error, thereby reducing the impact of error annotation on the detection model.

Comparison with Anomaly Detection Methods. Figure 6 shows the superiority of LogLR over other semi-supervised and unsupervised learning methods. LogLR captures the logical relationship of temporal log vectors through preset axioms, extracts the logical information of log sequences, and achieves better detection results. DeepLog and LogAnomaly outperform BGL on HDFS dataset, This is because there are more unstable data in BGL due to its longer time span compared with HDFS. More specifically, the BGL dataset is unstable, there are a lot of data in the test data that did not appear during training. DeepLog and LogAnomaly predict log events based on log sequences, are sensitive to unseen log events, and detect unseen log events as anomalies. PLELog only performs simple binary classification processing on the time log vector, ignoring the logical relationship between log events. Compared with LogGAN, LogLR achieves better results by pre-extracting the temporal characteristics of log sequences.

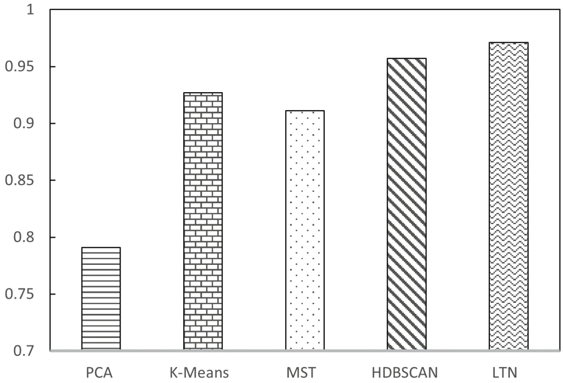
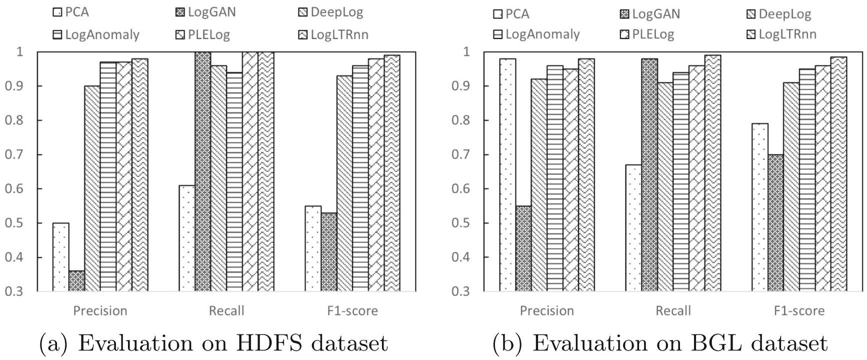


Fig. 5. Experimental results of HDFS dataset data label accuracy

LogLR outperforms existing state-of-the-art unsupervised and semi-supervised learning methods. Table. 1 shows the comparison of LogLR with the state-of-the-art supervised learning methods. Although there is a gap between LogLR and LogRobust, the gap between the three metrics is very small. This shows that LogLR combines the advantages of supervised learning well with weak label estimation methods. Moreover, as LogRobust depends on a large amount of manually labeled training data, LogLR has greater usability in practice.



(a) Evaluation on HDFS dataset

(b) Evaluation on BGL dataset

Fig. 6. Evaluation on two datasets

Table 1. Comparison with supervised learning method

	LogRobust-HDFS	LogRobust-BGL	LogLR-HDFS	LogLR-BGL
Precision	0.98	1.00	0.98	0.98
Recall	1.00	1.00	0.99	1.00
F1-score	0.99	1.00	0.99	0.99

5 Conclusion

Over the years, many log-based anomaly detection methods have been proposed, but they lack inference mechanisms that prevent models from mining logical relationships between logs. In this paper, we propose LogLR, a log anomaly detection method based on logical reasoning. LogLR extracts the temporal and logical information of log sequences by effectively combining LTN and LSTM. LogLR uses LTN to detect anomalies while applying LTN to data annotation, which not only saves time costs, but also maintains the accuracy of supervised learning. Finally, we demonstrate the effectiveness of LogLR on the two most widely used public datasets, demonstrating that LogLR outperforms current state-of-the-art methods.

References

1. Badreddine, S., Garcez, A.d., Serafini, L., Spranger, M.: Logic tensor networks. Artif. Intell. **303**, 103649 (2022)
2. Dit, B., Guerrouj, L., Poshyvanyk, D., Antoniol, G.: Can better identifier splitting techniques help feature location? In: 2011 IEEE 19th International Conference on Program Comprehension, pp. 11–20. IEEE (2011)
3. Du, M., Li, F., Zheng, G., Srikumar, DeepLog: anomaly detection and diagnosis from system logs through deep learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1285–1298 (2017)
4. Dunia, R., Qin, S.J.: Multi-dimensional fault diagnosis using a subspace approach. In: American Control Conference. Citeseer (1997)
5. Gower, J.C., Ross, G.J.: Minimum spanning trees and single linkage cluster analysis. J. Roy. Stat. Soc.: Ser. C (Appl. Stat.) **18**(1), 54–64 (1969)
6. Gu, J., Lu, S.: An effective intrusion detection approach using SVM with naïve bayes feature embedding. Comput. Secur. **103**, 102158 (2021)
7. He, P., Zhu, J., Zheng, Z., Lyu, M.R.: Drain: an online log parsing approach with fixed depth tree. In: 2017 IEEE International Conference on Web Services (ICWS), pp. 33–40. IEEE (2017)
8. He, S., Zhu, J., He, P., Lyu, M.R.: Experience report: system log analysis for anomaly detection. In: 2016 IEEE 27th international symposium on software reliability engineering (ISSRE), pp. 207–218. IEEE (2016)
9. Liang, Y., Zhang, Y., Xiong, H., Sahoo, R.: Failure prediction in IBM bluegene/l event logs. In: Seventh IEEE International Conference on Data Mining (ICDM 2007), pp. 583–588. IEEE (2007)

10. Lin, Q., Zhang, H., Lou, J.G., Zhang, Y., Chen, X.: Log clustering based problem identification for online service systems. In: Proceedings of the 38th International Conference on Software Engineering Companion, pp. 102–111 (2016)
11. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
12. Lou, J.G., Fu, Q., Yang, S., Xu, Y., Li, J.: Mining invariants from console logs for system problem detection. In: USENIX Annual Technical Conference, pp. 1–14 (2010)
13. McInnes, L., Healy, J.: Accelerated hierarchical density based clustering. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 33–42. IEEE (2017)
14. Meng, et al.: LogClass: anomalous log identification and classification with partial labels. IEEE Trans. Netw. Serv. Manage. **18**(2), 1870–1884 (2021)
15. Meng, W., et al.: LogAnomaly: unsupervised detection of sequential and quantitative anomalies in unstructured logs. In: IJCAI, vol. 19, pp. 4739–4745 (2019)
16. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
17. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. **24**(5), 513–523 (1988)
18. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001)
19. Tax, D.M., Duin, R.P.: Support vector data description. Mach. Learn. **54**(1), 45–66 (2004)
20. Wang, Z., Chen, Z., Ni, J., Liu, H., Chen, H., Tang, J.: Multi-scale one-class recurrent neural networks for discrete event sequence anomaly detection. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 3726–3734 (2021)
21. Wright, R.E.: Logistic regression. (1995)
22. Wu, D., et al.: LSTM learning with Bayesian and Gaussian processing for anomaly detection in industrial Iot. IEEE Trans. Industr. Inf. **16**(8), 5244–5253 (2019)
23. Xia, B., Bai, Y., Yin, J., Li, Y., Xu, J.: LogGAN: a log-level generative adversarial network for anomaly detection using permutation event modeling. Inf. Syst. Front. **23**(2), 285–298 (2021)
24. Xu, W., Huang, L., Fox, A., Patterson, D., Jordan, M.: Largescale system problem detection by mining console logs. In: Proceedings of SOSP 2009 (2009)
25. Yang, L., et al.: Semi-supervised log-based anomaly detection via probabilistic label estimation. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), pp. 1448–1460. IEEE (2021)
26. Zhang, X., et al.: Robust log-based anomaly detection on unstable log data. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 807–817 (2019)

A Software Security Entity Relationships Prediction Framework Based on Knowledge Graph Embedding Using Sentence-Bert

Yan Wang[1], Xiaowei Hou[1,2], Xiu Ma[1,2], and Qiujian Lv[1](✉)

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{wangyan,houxiaowei,maxiu,lvqiujian}@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Science, Beijing, China

Abstract. Recently, the need for complex cyber attack knowledge is increasing with the rising risk of software vulnerabilities and weaknesses on the internet. To spread knowledge and strengthen software security defense, researchers record software vulnerabilities, weaknesses, and attack patterns through software databases, including CVE, CWE, and CAPEC, etc. However, software security databases are time delayed and thus miss unobserved facts. Attackers can take advantage of this problem to execute an attack successfully. Therefore, the reasoning task of predicting software security entity relation is critical to supplementing software security data. This paper constructs a software security knowledge graph and proposes a knowledge graph representation learning method combining Sentence-Bert and GAT. The way can implement link prediction and classification tasks for knowledge graph completion. We finally designed a large number of experiments to evaluate the effectiveness of our model in knowledge graph completion and knowledge graph classification. The experimental results demonstrate that the proposed method can effectively improve the effectiveness of prediction.

Keywords: Software security entity · Entity relation prediction · Sentence-bert · Knowledge graph embedding · Graph attention network

1 Introduction

Software security data is stored in knowledge databases, a rich set of relationships between the same type of entities or different types of entities. Researchers take advantage of these security databases to manage the information on software vulnerabilities, weaknesses, and attack patterns to protect security and share security knowledge with the attacked organization. Common Vulnerabilities and Exposures (CVE) list publicly identified security vulnerabilities. For example, CVE-2017-0144 is a security vulnerability that uses the SMB protocol of Windows to obtain the highest permissions of the system so that attackers

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 501–513, 2022.
https://doi.org/10.1007/978-3-031-19214-2_42

can control the compromised computer. The Common Weakness Enumeration (CWE) presents software weaknesses developed by the security community, such as CWE-506 is embedded malicious code. Common Attack Pattern Enumeration and Classification (CAPEC) defines specific attack patterns and relevant solutions to defense. The CAPEC-185 denotes its attack pattern with "Malicious Software Download".

Entities in the three security databases have a large number of relations. For instance, the CAPEC-112, which summarizes the attack pattern of brute force, can associate the CAPEC-664 with the relation "CanPrecede". Meanwhile, ParentOf and ChildOf relationships between CWE (or CAPEC) reveal similar weaknesses (or attack patterns) that may exist in superior-subordinate relationships such as <CWE-330 ParentOf CWE-804>. Security entity relations give plenty of security information that benefits experts for security analysis and vulnerability repair. However, the CVE, CWE, and CAPEC databases inevitably miss software security entity-relationship. We need speed time to wait for software security databases to be updated, so an attacker can take advantage of this time gap to implement an attack successfully.

We construct a software security knowledge graph and propose software security entity-relationship prediction based on graph representation learning to address the above limitations of security databases. Facing the dynamic complex network environments, internal and external network threat intelligence, and the increasing need for network defense, knowledge graphs show excellent application potential in the network security area because of their capabilities in knowledge aggregation, representation, management, and reasoning [7]. In our study, the software security knowledge graph construction can associate multi-source security knowledge based on software security. Therefore, implementing software security entity-relationship prediction helps achieve threat intelligence reasoning to improve the ability to protect software security.

We summarize the main contributions as follows:

(1) We design a model based on sentence-Bert representation learning, which learns the paragraph features an entity and relationship description statements and has better prediction performance.
(2) We enhance the representation vector of our software security knowledge graph by introducing an auxiliary relationship of multi-hop neighbors between two entities to acquire knowledge from the distant neighbors of an entity to obtain information about the relationships between triples.
(3) Experiments on scenarios such as relationship prediction, entity prediction, triple classification, and multi-classification of vulnerability severity are designed to demonstrate their effectiveness on software security knowledge graph reasoning tasks.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 presents the details of the proposed model. Section 4 describes the detailed setup of the experiments. The experimental results are described in Sect. 5. Finally, Sect. 6 concludes our research.

2 Related Work

Firstly, this section introduces three software security databases we used in our research. Then, we focus on the software security entity prediction models based on knowledge graph representation learning. Finally, we review some current models of knowledge graph representation learning.

Knowledge graph representation learning attempts to represent entity and relation into a low-dimensional dense vector. It is widely used in various domains and has proven to predict the implied security entity relationships from software security databases.

CVE, CWE, and CAPEC consist of rich software security knowledge. Common Vulnerabilities and Exposures (CVE) [3] provides a unique identification number for each public security vulnerability or exposure. Common Weakness Enumeration (CWE) [4] is a list of software weakness types developed by the security community. Common Attack Pattern Enumeration and Classification (CAPEC) [2] provides a comprehensive dictionary of known attack patterns that attackers employ to exploit the software weaknesses in the applications and systems.

In recent work, Han [5] constructed a knowledge graph of common software weaknesses and proposed a knowledge graph embedding approach to embed the structural and textual knowledge of CWE into vector representations. Xiao [13] proposed a knowledge graph embedding approach to embed the knowledge of security concepts and instances based on the model of CNN. Yuan [15] designed a text-enhanced GAT model to represent the structural and textual knowledge from the security knowledge graph.

Knowledge graph embedding can project a large-scale knowledge graph into a continuous low-dimension vector space. There are many methods for modeling entities and relations of triples in the knowledge graph. TransE [1], TransH [12], and TransR [6] are translation-based methods that could achieve the task of link prediction in the knowledge graph. Among them, TransE is a basic knowledge graph embedding model, and it interprets the relations as translating operations between head and tail entities on the low-dimensional vector space. The above translation-based models focus on the structural information between entities, regardless of rich information encoded in entity descriptions. There are several methods using textual information to help knowledge graph representation learning. Socher [11] represented the new neural tensor network, which represents each entity as the average of its word vectors. Xie [14] explored the continuous bag-of-words and deep convolutional neural models to encode the semantics of entity description. Reimers [10] introduced the SBERT model, which can create sentence embedding that outperforms other embedding methods significantly. However, these models consider the triple independently so that they can not capture the potential relations in the adjacent neighbors of each entity. To solve this problem, Nathani [8] introduced the graph attention network represents each entity with the potential knowledge from its multi-hop neighbors.

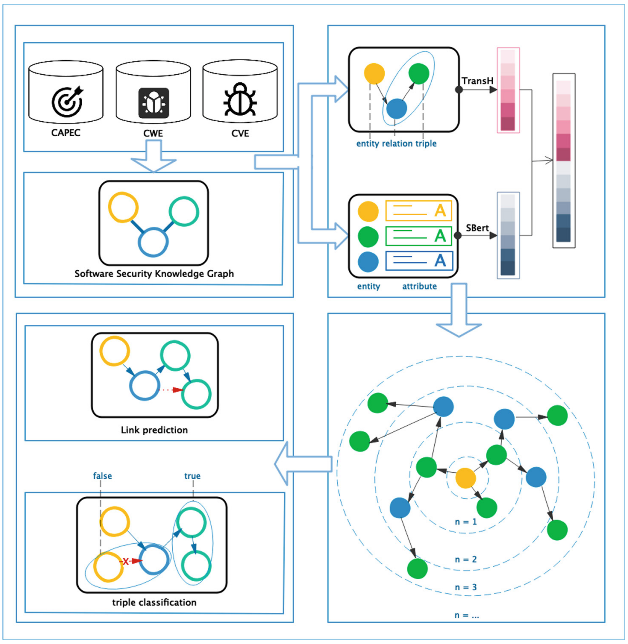


Fig. 1. The architecture of our model.

3 Model Architecture

This section describes the model architecture and the algorithm details behind our model. Figure 1 illustrates the workflow of our method. We design a method for reasoning about software security entity relationships based on knowledge graph representation learning. Our model consists of four parts, and the detailed description of each step is as follows.

3.1 Knowledge Graph Construction

To aggregate scattered software security information, we construct the software security knowledge graph using three semi-structured databases, including CVE, CWE, and CAPEC. We show an example of the software security knowledge graph in Fig. 2. And it can be typically defined as

$$G = (E, R, T) \tag{1}$$

where G is a labeled and directed multi-graph, E, R, and T are sets of entities, relations, and triples, respectively. In the knowledge graph, we construct each triple that is denoted as

$$T = \{(h, r, t)\,|h, t \in E, r \in R\} \tag{2}$$

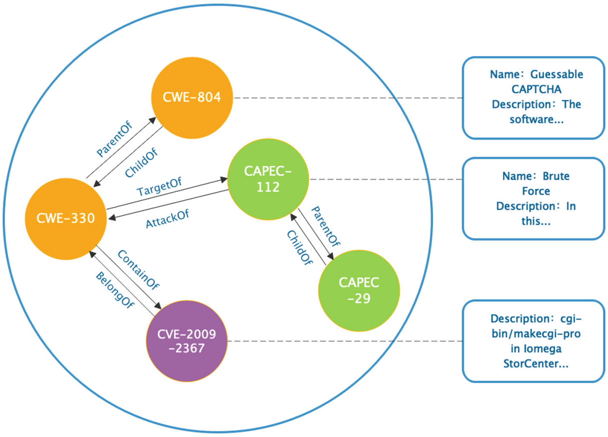

CWE-804
CWE-330
CAPEC-112
CAPEC-29
CVE-2009-2367
ParentOf
ChildOf
TargetOf
AttackOf
ContainOf
BelongOf
ParentOf
ChildOf
Name: Guessable CAPTCHA Description: The software...
Name: Brute Force Description: In this...
Description: cgi-bin/makecgi-pro in Iomega StorCenter...

Fig. 2. An example of software security knowledge graph.

where h, r, and t are the head entity, relation, and tail entity, respectively. This formula represents a fact, i.e., a relationship between the head and tail entities. For example, (CWE-330, ContainOf, CVE-2009-2367) is a triple shown in Fig. 2. CWE-330 is the head entity in this triple, ContainOf is the relation, and CVE-2009-2367 is the tail entity.

Triples are the generic representation of a knowledge graph. There are 16746 triples in the entire knowledge graph. The entities in all triples have three types, including CVE ID, CWE ID, and CAPEC ID, because they come from three heterogeneous databases, respectively. And the number of entities is 4144, of which 2677 are the entities in CVE, 924 are the entities in CWE, and 544 are the entities in CAPEC. Moreover, relations contain twelve types of relationships, such as ChildOf, PeerOf, etc. It is worth noting that we construct a two-way relationship between entities.

3.2 Structure Embedding Generation

Structure embedding generation, a usual method of knowledge graph trained by TransH, is the second step of our models. This step aims to mine the potential relationships of triples in the knowledge graph. To improve the model's efficiency and consider the one-to-many, many-to-one, and many-to-many characteristics of entities in our knowledge graph, we select the TransH model to extract the structural features. The TransH models a relation as a hyperplane and a translation operation on it. Then, the score function is

$$f_r(\boldsymbol{h}, \boldsymbol{t}) = \|(\boldsymbol{h} - \boldsymbol{w}_r^\top \boldsymbol{h} \boldsymbol{w}_r) + \boldsymbol{d}_r - (\boldsymbol{t} - \boldsymbol{w}_r^\top \boldsymbol{t} \boldsymbol{w}_r)\|_2^2 \tag{3}$$

In this way, we can preserve the mapping properties of the given relations. Meanwhile, the training efficiency of the model exceeds that of the TransE model. Firstly, we input the training set constructed in the previous step into the TransH model, which maps the triple to a continuous vector space. We set the structure-embedding vector with a dimension of 128 for obtaining adequate information based on the structure of the knowledge graph. Then, the output is the structure vector of each security entity.

3.3 Description Embedding Generation

However, limited by the sparseness of the knowledge graph structure, representation learning only from a structural perspective can no longer meet our research needs. We introduced the auxiliary information to improve the performance. Moreover, we regard that the two entities with similar descriptive sentences are more likely to be related in this phase. Similar ones will obtain a high similarity vector when description statements are mapped into the vector space. We adopt the Sentence-Bert model to embed the description information of security entities and relationships equally into a continuous vector space, thus providing a representation-enhancing effect on the structural embedding. Sentence-BERT is a modification of the BERT network using triplet networks that can derive semantically meaningful sentence embedding. Adding a pooling layer to the pre-trained BERT model produces a high accuracy of sentence embedding. The role of pooling is to combine these vectors into a fixed-length sentence vector.

Our research extracts the name and description field in three databases as texture information. For example, the name of CAPEC-112 is brute force. Its description is The attacker attempts to gain access to this asset by using trial-and-error to exhaustively explore all the possible secret values to find the secret that will unlock the asset. We input the texture information of all security entities in the knowledge graph. Then the pre-trained Sentence-Bert model is loaded to obtain the sentence vector. And finally, we receive the sentence vector representation of the descriptive statements.

The joint embedding vector is generated by concatenating the structure embedding vector trained by TransH and the texture embedding vector trained by SBERT. Therefore, the joint embedding vector combines two vector representations and has a better prediction performance than the vector representation only based on the structure of the knowledge graph.

3.4 Graph Attention Layer

There are two parts to this section. Firstly, the joint embedding vector is generated by concatenating the structure embedding vector trained by TransH and the texture embedding vector trained by SBERT. Therefore, the joint embedding vector combines two vector representations and has a better prediction performance than the vector representation only based on the structure of the knowledge graph.

Secondly, we introduce the model of the graph attention. Our software security knowledge graph represents structural, and texture features as a global vector. However, we observe that every entity has at least one neighbor who is an entity in other triples. And then we can find out neighbors of the entities and even more. Thus, we can discover the multi-hop neighbors of entities and then discover the local relationships. Meanwhile, the model of graph attention can consider the information of neighboring nodes in the knowledge graph to reflect the diversity of roles of the same security entity in different relationships.

In detail, Our model achieves these objectives by assigning different weights to different neighborhood nodes and by propagating attention via layers in an iterative fashion. We introduce an auxiliary edge between multi-hop neighbors, which allows the flow of knowledge between entities. And then, our model can allow the flow of knowledge between entities to decrease the negative influence of the distance between entities.

3.5 Model Training

The objective of our model in this step is to minimize the score function, i.e., the loss function:

$$L = \sum_{(h,r,t)\in T} \sum_{(h',r,t')\in T'} \max(\gamma + E(h,r,t) - E(h^{'},r,t^{'}),0) \tag{4}$$

where γ is a margin hyperparameter that is limited to be greater than 0. $E(h,r,t)$ is the energy function, which is defined as

$$E = E_s \oplus E_d \tag{5}$$

where E_s is the energy function of the structure-based representation, which is defined as

$$E_s = \|h_s + r_s - t_s\| \tag{6}$$

and E_d is the energy function of the texture-based representation, which is defined as

$$E_d = \|h_d + r_s - t_s\| \tag{7}$$

In triple T, head or tail entities are replaced by another entity. And then, an invalid triple that does not exist in our SSKG is formed, which is defined as

$$T^{'} = \left\{(h^{'},r,t)|h^{'} \in E \cup (h,r,t^{'})|t^{'} \in E\right\} \tag{8}$$

Finally, we adopt the ConvKB algorithm [9] as a decoder to compute the final score of the triple. And it is trained using soft-margin loss.

4 Experiments Setup

4.1 Datasets

Our study constructs a software security knowledge graph based on CWE, CVE, and CAPEC databases. The experimental database was collected from the version of November 5, 2021. We extract CWE ID, CVE ID, and CAPEC ID as entities, the titles and descriptions of CVE, CWE, and CAPEC as texture information of the entities, and extract the relations between different entities to form triples. The type of relation in SSKG is shown in Table 1. We create 12 types of relations and then complement the complementary relationships in the datasets. We implement this step to help entities as much as possible to find out their neighbor's information by the operation of multi-hop.

We remove the entities without descriptions in the pre-processed datasets to ensure that each entity has a description statement, and we also remove all the triples containing these entities. There are 16748 triples in the dataset, of which 85% are in the training set, and 15% are in the test set.

4.2 Evaluation Protocols

We implement TransE, TransH, CNN, Sentence-Bert, GAT, and other models for comparison. The specific experimental settings are: structural embedding dimension of entities and relations is 128d, textual embedding dimension is 384 d, GAT embedding dimension is 512 d, final output dimension is 200 d, the learning rate is set to $1e^{-3}$, and margin is set to 1.0.

4.3 Baseline Protocols

We utilize six baseline models on the dataset we construct above. And the baseline methods adopt the same training set and hyper-parameter settings as our approach.

Baseline1: Baseline 1 is designed to extract structure-based features in CVE, CWE, CAPEC entities, and relations using the primary knowledge graph embedding method TransE.

Baseline2: Baseline 2 extracts structure-based features in CVE, CWE, CAPEC entities, and relations using the baseline knowledge graph embedding method TransH.

Baseline3: Baseline 3 is designed for extracting texture-based and structure-based features of entities and relations in triples. It extracts structure-based features and texture-based features using TransE and Sentence-Bert, respectively.

Baseline4: Baseline 4 extracts structure-based and texture-based features using TransH and Sentence-Bert, respectively.

Baseline5: Baseline 5 is proposed by Xiao [13]. A new embedding method is to create a knowledge graph embedding model based on the translation model, CNN textual encoding, and word embedding by joint training.

Baseline6: Baseline 6 is proposed by Yuan [15]. It is a textual-enhanced GAT model for better representing and learning structural and textual knowledge from the software security knowledge graph, which integrates CVE, CWE, and CAPEC.

Table 1. The type of relation in software security knowledge graph.

Type	ChildOf	ParentOf	PeerOf	CanPrecede	Requires	CanAlsoBe
Number	1649	1649	132	293	13	30
Type	StartsWith	CanFollow	BelongOf	ContainOf	TargetOf	CanAttack
Number	3	293	5191	5191	1152	1152

5 Experiments Results

We validate the effectiveness of our model through multi-hop and ablation studies. And then, we conduct two application experiments on link prediction and triple classification tasks.

5.1 Multi-hop Study

This section focuses on the impact of neighbor hop times on entity predictive performance. For example, when n is equal to 1, our model finds out the most relevant nodes of the graph, namely the node features in a 1-hop neighborhood. And then, our model extends the attention mechanism in a multi-hop neighborhood of a given node.

The results of evaluating the overall performance on entity predicting are shown in Table 2. We can see that the 3-hop has the best performance in the task. The Hits@10 of our model outperforms the 1-hop, 2-hop, 3-hop, 4-hop, and 5-hop by 0.091, 0.029, 0.063, and 0.086, respectively. And the Mean Rank of our model outperforms the 1-hop, 2-hop, 3-hop, 4-hop, and 5-hop by 45.5, 3.9, 22.9, and 27.5, respectively. The multi-hop has better performance than the 1-hop in that multi-hop encapsulates more rich semantics and potential relationships. However, it does not mean that the model's predictive performance is better with more hops. When $n = 4$, the model's performance begins to decline due to over-fitting.

Table 2. Evaluation results on multi-hop study.

Metric	Hits@10	Mean rank
1-hop	0.831	72.4
2-hop	0.893	30.8
3-hop	0.922	26.9
4-hop	0.859	49.8
5-hop	0.836	54.4

5.2 Application Studies

Knowledge Graph Link Prediction. In this step, we want to study how our model can improve the accuracy of knowledge graph link prediction. The task of knowledge graph link prediction is to minimize the scoring function $T(h, r, t) = \|h + r - t\|$ and to complement a triple (h, r, t) when one of h, t, r is missing.

We treat the knowledge graph link prediction task as two sub-tasks: entity prediction and relation prediction. And then, we experiment with our method and several baseline methods with the embedding representations. We use two metrics to evaluate our experimental results:

(1) The proportion of correct entities ranked in the top 10 of the predicted entities(Hits@10), and the proportion of correct relations ranked in the top 3 of the predicted relations(Hits@3).
(2) The average ranking of correct entities or relations(Mean Rank).

According to the results in Table 2, our model achieves the best performance in both evaluation metrics compared with other baseline methods. The Hits10 of our model outperforms the Baseline1, Baseline2, Baseline3, Baseline4, Baseline5, Baseline6 by 0.596, 0.436, 0.586, 0.410, 0.262, and 0.058, respectively. Similar conclusions can be drawn on mean rank. Our model outperforms the Baseline1, Baseline2, Baseline3, Baseline4, Baseline5, and Baseline6 on mean rank by 533.9, 502.7, 505.1, 403.8, 84.3, and 34.0, respectively.

Meanwhile, the results on relation ranking shown in Table 3 are similar to the effects on entity ranking. Our model also achieves the best performance compared with the two baseline methods. The Hits@3 of our model outperforms the Baseline1 and Baseline3 by 0.160 and 0.176, respectively. And the Mean Rank of our model exceeds the Baseline1 and Baseline3 by 1.68 and 1.71, respectively.

Our model achieves the best performance on the knowledge graph link prediction task compared with the baseline approaches. And concatenation of structure-based and texture-based representation outperforms the model of only structure-based representation. Therefore, structure-based representation helps improve link prediction results, but much textual information could lead to better performance. Meanwhile, we can infer that our method based on the SBERT model and GAT can represent the textual information of an entity and extract more auxiliary relations from the multi-hop neighbors (Table 4).

Table 3. Evaluation results on entity prediction.

Metric	Hits@10	Mean rank
Baseline 1	0.326	559.8
Baseline 3	0.336	531.0
Baseline 2	0.486	528.6
Baseline 4	0.512	429.7
Baseline 5	0.660	110.2
Baseline 6	0.864	59.9
Our model	0.922	25.9

Table 4. Evaluation results on relation prediction.

Metric	Hits@3	Mean rank
Baseline 1	0.784	3.05
Baseline 3	0.768	3.08
Our model	0.944	1.37

Triple Classification. This section aims to investigate whether and to what extent our model can correctly classify the heterogeneous triples compared with the Baseline2 and Baseline4. And the goal of the triple classification task is to reach a given triple(h, r, t) with the correct triple in the test set and then determine whether it is a positive triple or not. This problem can be viewed as a binary classification problem. And similar to link prediction, this experiment applies only to our model, the Baseline2 and Baseline4.

The evaluation of this task requires setting up negative triples. Because our dataset contains only correct triples, we need to set some negative triples. The experimental setup is done in the same way as entity prediction. The head or tail entities of the triples are replaced by entities from a random list of entities, denoted as (h', r, t) or (h, r, t'). And then, a binary classifier is trained. The purpose of this classifier by the energy function is less than this threshold. The triple is the positive triple. Otherwise, it is the negative triple. In this task, we use four metrics, including Accuracy, Precision, Recall, and F1, for evaluating the overall performance of the triple classification task shown in Table 5.

Therefore, our model can support triples reasoning tasks and achieve the best performance compared with the baseline models. And Table 5 presents the results of the triple classification task compared with Baseline2 and Baseline4.

Table 5. Evaluation results on triple classification.

Metric	Accuracy	Precision	Recall	F1
Baseline 2	0.749	0.864	0.294	0.438
Baseline 4	0.760	0.836	0.347	0.490
Our model	0.794	0.933	0.394	0.552

Results claim that our model based on knowledge graph embedding can successfully support the triple classification task. And our model could slightly outperform the only structure-based baseline2 and the concatenation representation of structure-based and texture-based Baseline4.

6 Conclusion

This paper proposes a model to predict software security entity relation in the software security knowledge graph we build, and a series of experiments are conducted on real datasets. As a result, our model can predict entity relation and classify triples as a general model. It behaves better than the state-of-the-art methods. Especially, it can learn the paragraph features and the information of multi-hop neighbors to help models perform better predictions.

In the future, it is hoped that this method can be validated in network security problems from various domains. Our model has been tested on real datasets using software security knowledge graph, which lays a foundation for application to the problems of predicting knowledge graph entity relation in other domains. The efforts here may motivate the necessity and encourage further research into predicting software security and vulnerability detection.

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
2. CAPEC: Common attack pattern enumeration and classification (2022). https://capec.mitre.org/
3. CVE: Common vulnerabilities and exposures (2022). https://www.cve.org/
4. CWE: Common weakness enumeration (2022). https://cwe.mitre.org/
5. Han, Z., Li, X., Liu, H., Xing, Z., Feng, Z.: Deepweak: reasoning common software weaknesses via knowledge graph embedding. In: 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 456–466. IEEE (2018)
6. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
7. Liu, K., Wang, F., Ding, Z., Liang, S., Yu, Z., Zhou, Y.: A review of knowledge graph application scenarios in cyber security. arXiv preprint arXiv:2204.04769 (2022)
8. Nathani, D., Chauhan, J., Sharma, C., Kaul, M.: Learning attention-based embeddings for relation prediction in knowledge graphs. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4710–4723 (2019)
9. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: A novel embedding model for knowledge base completion based on convolutional neural network. In: NAACL HLT 2018: 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies-Proceedings of the Conference, pp. 327–333. Association for Computational Linguistics (2018)
10. Reimers, N., et al.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, pp. 671–688. Association for Computational Linguistics (2019)
11. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
12. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
13. Xiao, H., Xing, Z., Li, X., Guo, H.: Embedding and predicting software security entity relationships: a knowledge graph based approach. In: Gedeon, T., Wong, K.W., Lee, M. (eds.) ICONIP 2019. LNCS, vol. 11955, pp. 50–63. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36718-3_5

14. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)
15. Yuan, L., Bai, Y., Xing, Z., Chen, S., Li, X., Deng, Z.: Predicting entity relations across different security databases by using graph attention network. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 834–843. IEEE (2021)

A Secure Task Matching Scheme in Crowdsourcing Based on Blockchain

Di Jiang[1], Jiajun Chen[1], Chunqiang Hu[1,2](✉), Yan Lei[1], and Haibo Hu[1]

[1] School of Big Data and Software Engineering, Chongqing University,
Chongqing 400044, China
{dijiang,jiajunchen,chu,yanlei,haibo.hu}@cqu.edu.cn

[2] Joint Laboratory on Cyberspace Security, China Southern Power Grid,
Guangzhou, China

Abstract. As crowdsourcing continues to evolve, researchers explored task matching in crowdsourcing extensively. However, the privacy issues such as task content of publishers and ability or interest of worker in task matching are often overlooked. Also, the identity of the task publisher/worker needs to be protected. To address the above issues, we propose a secure task matching scheme in crowdsourcing based on blockchain in this paper. Firstly, we implement multi-publisher/multi-worker task matching in the scheme while protecting task content privacy. Meanwhile, we take advantage of the immutability of the blockchain to ensure the reliability of publishing/matching results. We utilize the smart contract for task publishing/matching without human intervention. Finally, the scheme is shown to be secure and feasible through theoretical and comprehensive performance evaluations.

Keywords: Task matching · Multi-publisher/Multi-worker · Blockchain · Smart contract

1 Introduction

Crowdsourcing is a new distributed paradigm that adopts the idea of gathering wisdom to accomplish greater tasks and is mainly used for high computational or wide coverage tasks. With the rise of crowdsourcing in the past few years, lots of researchers are interested in it [1–3]. At the same time, many companies have introduced crowdsourcing into their work, such as Amazon Mechanical, MTurk and CrowdFlower.

Task matching is an integral part of the crowdsourcing system, allowing workers to find the tasks they are interested in. What is noted that privacy protection in task matching in crowdsourcing is often overlooked. In fact, privacy-preserving crowdsourcing schemes also exist in existing research, particularly spatial crowdsourcing. In schemes [4,5], they protect the worker's for location services. However, it is also important to protect the privacy of task content. Due to the lack of

Supported by National Key Research and Development Project under grant 2020YFB1711900, and the National Natural Science Foundation of China under grant 62072065.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 514–525, 2022.
https://doi.org/10.1007/978-3-031-19214-2_43

task content, there is thus the possibility of malicious servers inferring the sensitive information of task publishers/workers. Therefore, how to protect the privacy of task content is a challenge.

In order to protect the privacy of task content, it is necessary to encrypt the task content before publish it, and the ciphertext needs to be matched in the crowdsourcing system. Searchable encryption will provide a good solution for task matching in crowdsourcing system. Searchable confidential schemes such as [6,7] are mainly for single users, users generate their own ciphertexts that can only be matched by themselves. Nonetheless, with the existence of multi-publisher/multi-worker in the crowdsourcing system, the above scheme is clearly not sufficient. Therefore, it is necessary to implement multi-publisher/multi-worker task matching with the privacy of the task content protected.

When using searchable encryption in task matching, it is important to understand that the crowdsourcing system may be some dishonst behavior [8]. It wants to explore sensitive information of task publisher/worker, and even gives wrong matching results to the worker. Therefore, how to avoid the wrong results of crowdsourcing system is a challenge. With the rise of Bitcoin [9], blockchain is gradually gaining the attention of researchers [10,11]. As a distributed ledger, blockchain combines consensus algorithms, cryptography, etc. to achieve the consistency and credibility of multi-party nodes. Smart contracts are codes that run on the blockchain and use the characteristics of the blockchain to ensure correct and automatic execution without trusted third parties.

In this paper, considering the above problems of task matching in crowdsourcing, we design a multi-publisher/multi-worker, privacy-preserving task matching in crowdsourcing system based on the blockchain with smart contract. The main contributions of this paper can be summarized as follows:

- We design multi-publisher/multi-worker task matching scheme and implement privacy protection for task content.
- We guarantee correct task matching result adopting smart contracts in the blockchain. Meanwhile, due to the anonymity of the blockchain, the identity of the task publisher/task worker is also protected.
- Compared with other schemes, our scheme has advantages in time cost while ensuring security.

The remainder of this paper is organized as follows. Section 2 introduces related works. Section 3 describes related preliminaries contained in our scheme. Section 4 presents the system model, threat model and design goals. Section 5 shows the details of our scheme. Sections 6 and 7 present the security and performance analyses, respectively. Finally, conclusions are drawn in Sect. 8.

2 Related Works

With the rise of crowdsourcing, task matching is becoming increasingly popular among researchers. At the same time, the privacy protection issue of task matching in crowdsourcing is gradually coming into the view of researchers.

To *et al.* [12] utilized differential privacy techniques to propose a privacy-aware framework that effectively protected the spatial privacy of workers. Xu *et al.* [13] protected the privacy of publishers and workers with inner-product encryption and one-time anonymous authentication mechanisms, while allowing for detection of multiple task submissions.

Searchable encryption can be a way to protect the privacy of task content. Song *et al.* [6] proposed the first symmetric searchable secrecy scheme in order to solve the search problem against encrypted data. Boneh *et al.* [14] constructed the first public-key searchable encryption algorithm in order to solve the search problem of public-key encrypted data and with an email sending and receiving scenario. However, multi-publisher/multi-worker need to be satisfied in task matching in crowdsourcing. Kiayias *et al.* [15] utilized the master key to derive multiple private keys for the purpose of multiple users, but the scheme needed huge consumption. Furthermore, Shu *et al.* [16,17] utilized searchable encryption to protect the privacy of task content, while also devising effective staff revocation, but the centralized servers that they used may be dishonest.

With the development of the blockchain, some researchers have introduced the blockchain into the crowdsourcing system. Zhu *et al.* [18] built a hybrid blockchain crowdsourcing platform which ensured communication security, verified transactions and privacy protection through dual ledgers, dual consensus and smart contracts. Zhang *et al.* [19] utilized smart contracts, rewritable deterministic hashing and searchable encryption to implement an agent-free privacy-preserving and federated crowdsourcing system.

3 Preliminaries

3.1 Bilinear Pairings

G_1, G_2 and G_T are all multiplicative cyclic groups of prime order p. Let g_i denotes a generator of G_i. A bilinear map $e : G_1 \times G_2 \rightarrow G_T$ has the following properties:

- *Bilinearity*: For all $\mu \in G_1, \nu \in G_2$, and $a, b \in Z_p^*, e(\mu^a, \nu^b) = e(\mu, \nu)^{ab}$.
- $Nondegeneracy$: $e(g_1, g_2) \neq 1$.
- *Computability*: It is efficient to compute e for any input.

3.2 Shamir Threshold Secret Sharing

Shamir threshold secret sharing is a (t, n) threshold secret distribution scheme. Suppose we exist a secret S. The secret S is subjected to a specific operation to obtain n secret fragments $S_i(0 < i <= n)$, which are given to n individuals to keep. The original secret S is restored when at least t individuals simultaneously take out the secret fragments S_i they have.

Shamir threshold secret sharing employs Lagrangian interpolation to restore secret. When the secret S needs to be shared to the n-parties, it picks $t-1$ random numbers $(a_1, a_2, ..., a_{t-1})$ and constructs a secret polynomial $f(x)$ as follows:

$$f(x) = S + a_1 x + a_2 x^2 + ... + a_{t-1} x^{t-1}$$

Then, each of the n parties i will obtain the secret $f(i)$. When the secret S needs to be restored, the secret set Γ consisting of at least t parties is obtained. For each party i in Γ, we compute the Lagrangian polynomial:

$$L_{i,\Gamma}(x) = \prod_{j \in \Gamma, j \neq i} \frac{x-j}{i-j}$$

Hence, secret S will be restored as follows:

$$S = f(0) = \sum_{i \in \Gamma} (f(i) \cdot L_{i,\Gamma}(0))$$

3.3 Blockchain and Smart Contract

Blockchain essentially works as a distributed ledger, which combines cryptographic algorithms, consensus mechanisms, distributed storage and P2P transmission as a solution to the trust and security problems of multiple nodes. Blockchain is mainly divided into public chain, private chain and federated chain. The blockchain consists of many blocks linked together, and the data structure of the blockchain is shown in the Fig. 1.

Smart contract was originally proposed by Szabo and have since been used in blockchain. It represents a piece of code that is deployed on the blockchain and cannot be changed once it is deployed. When the conditions are met, the smart contract will be executed automatically.

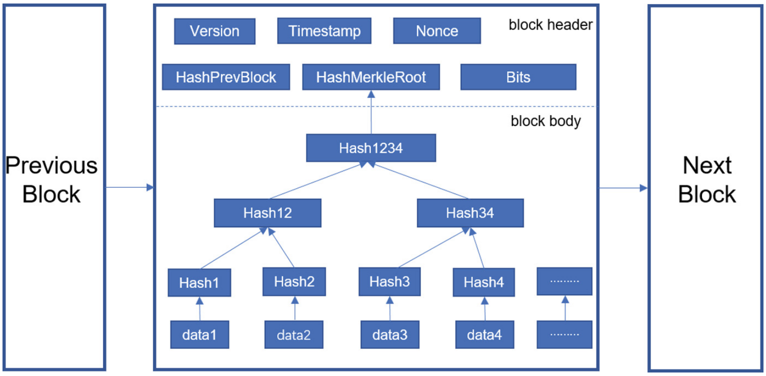


Fig. 1. Block structure

4 System Model, Threat Model, and Design Goals

In this section, we mainly formalize system model, threat model and design goals.

4.1 System Model

In our scheme, we provide a scheme for task matching in crowdsourcing based on blockchain. This scheme mainly includes four entities: task publisher, task worker, key management center (KMC) and blockchain system as shown in Fig. 2

- *KMC*:KMC mainly performs the initialization of the system, including the generation and distribution of keys. Meanwhile, KMC is responsible for the initialization of the blockchain and deployment of smart contracts.
- *Blockchain System*: The blockchain system contains a lot of nodes, which execute smart contracts and return the results of the smart contracts after consensus and upload them to the chain.
- *Task Publisher*: When a task publisher needs to publish a task, he encrypts his task content into ciphertext and publishes it using the task publishing smart contract (TPSC).
- *Task Worker*: When a task worker needs to match tasks, he generates trapdoors for the tasks he is interested in. Matching is performed utilizing task matching smart contract (TMSC).

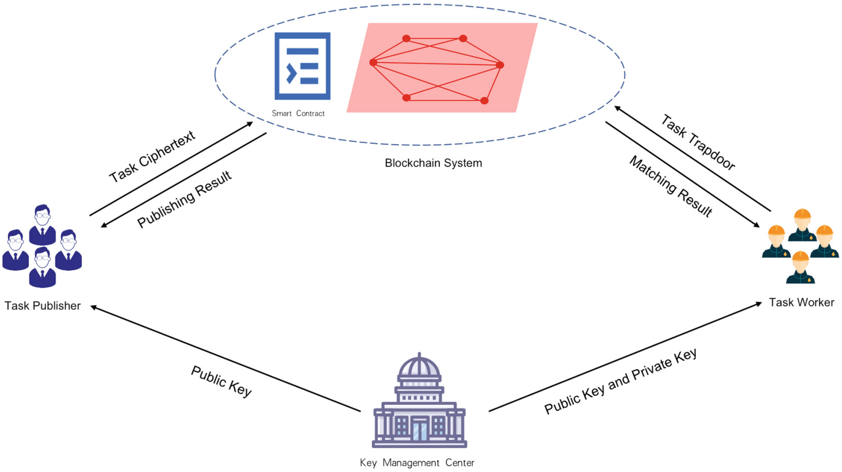


Fig. 2. System model

4.2 Threat Model

From the above system model, we consider the KMC as honest.

External Attack. An external attacker may intercept the ciphertext/trapdoor of the task publisher/task worker and guess his sensitive information.

Internal Attack. Nodes in the blockchain are curious about the sensitive information from ciphertext/trapdoor, and there may also be malicious nodes that wish to return false publishing or matching results.

4.3 Design Goals

Based on the above threat model, the design goals of our scheme are as follows:

- **Task confidentiality.** After obtaining the ciphertext/trapdoor of the task publisher/worker, the adversary cannot get information about the task or distinguish between any two tasks.
- **Identity anonymity.** The identity anonymity of the task publisher/worker in the scheme means that the identity and behavior of the publisher/worker cannot be linked.
- **Multi-publisher/Multi-worker matching.** While achieving privacy protection, each task worker can use their own private key for trapdoor generation.
- **Results dependability.** The task publisher/worker can get the correct publishing/matching results.

5 The Proposed Scheme

In this section, we'll employ blockchain to build a multi-publisher/multi-worker task matching. And we will describe the details of its build.

5.1 Summary of Model Notations

Table 1 provides a summary of the symbols used in our scheme.

Table 1. Summary of model notations

Notations	Description
$SMSK$	System key
SPK	Public key
SSK_i	Private key for each task worker
C, T	Task ciphertext/trapdoor for task publisher/worker
f	A secret polynomial function
H	A hash function
a, b, t, r_1, r_2	Random numbers
G_1, G_2	Multiplicative cyclic group
TPSC, TMSC	Task publishing/matching smart contract

5.2 System Initialization

In this process, KMC calls **KeyGen** and **SKeyGen** to complete the initialization of the keys, and distributes the keys. Then, blockchain is initialized and task publishing smart contract (TPSC) and task matching smart contract (TMSC) are deployed.

- **KeyGen**$(1^\lambda) \rightarrow (SMSK, SPK)$. This algorithm is executed by KMC, and it is applied to initialize the system key, public key. Taking as input the security parameter λ, it selects two multiplicative cyclic groups G_1 with g_1 as the generator and G_2 which have the same prime order p as G_1. And it defines a bilinear map $e : G_1 \times G_1 \rightarrow G_2$ and a hash function $H : \{0,1\}^* \rightarrow G_1$. Then, it chooses some random numbers $a, b, t \in Z_p^*$ and gets a secret polynomial function $f(x) = a + bx$. Thus, it outputs the system key SPK and the public key $SMSK$ as follows.

$$SPK = (g, g^{\frac{f(t)}{a}}), \qquad SMSK = (a, b, t)$$

- **SSKeyGen**$(SMSK, SPK) \rightarrow SSK_i$. This algorithm is executed by KMC, and it is applied to initialize the private keys for each worker. It selects a random number $t_i \in Z_p^*$ for each worker and sets $\varGamma_i = \{t, t_i\}$. After that, it selects a random number $s \in Z_p^*$ and outputs the private key SSK_i of each worker which can be obtained according to the Lagrangian interpolation polynomial as follows.

$$SSK_i = ((L_{t,\varGamma_i}(0) + \frac{f(t_i)}{f(t)} L_{t_i,\varGamma_i}(0))s, g^s)$$

5.3 Task Publishing

The task publisher who need for task publishing can call the **Enc** to get the ciphertext C based on his task content w and the public key SPK.

- **Enc**$(SPK, w) \rightarrow C$. It chooses a random number $r_1 \in Z_p^*$ and computses the ciphertext $C = (C_1, C_2)$ for the keyword w as:

$$C = (C_1, C_2) = (H(w)^{r_1}, g^{\frac{f(t)}{a} r_1})$$

The nodes on the blockchain invokes the smart contract TPSC to return the publishing results to the task publisher after consensus. In this paper the task ciphertext C is stored in the smart contract TPSC.

5.4 Task Matching

The task worker who need for task matching can call the **Trap** to get the matching trapdoor T based on his matching keyword q and his own private key SSK_i.

- **Trap**$(SSK_i, q) \rightarrow T$. It chooses a random number $r_2 \in Z_p^*$ and computes the trapdoor $T = (T_1, T_2)$ for the keyword q as:

$$T = (T_1, T_2) = (H(q)^{(L_{t,\Gamma_i}(0)+\frac{f(t_i)}{f(t)}L_{t_i,\Gamma_i}(0))sr_2}, g^{sr_2})$$

The nodes on the blockchain invokes the smart contract TMSC to return the matching results to the task worker after consensus. The smart contract TPSC stores all the tasks published by the task publisher. The smart contract TMSC achieve the task matching by calling $mathbfMatch$.

- **Match**$(C, T) \rightarrow 0/1$. This algorithm will be executed automatically by the smart contract TMSC. TMSC obtains the task ciphertext $C = (C_1, C_2)$ in TPSC and matches with the trapdoor $T = (T_1, T_2)$. It will check if $e(C_1, T_2) \overset{?}{=} e(T_1, C_2)$. If equal, it returns 1, otherwise it returns 0.

Theorem 1. *Matching Correctness. The task matching process is correct. That is, if* **Match**$(C, T) \rightarrow 1$*, then the keyword w of the task publisher is equal to the keyword q of the worker.*

Proof. Suppose **Match**$(C, T) \rightarrow 1$, and we have

$$\begin{aligned} e(C_1, T_2) &= e(T_1, C_2) \\ \Leftrightarrow e(H(w)^{r_1}, g^{sr_2}) &= e(H(q)^{(L_{t,\Gamma_i}(0)+\frac{f(t_i)}{f(t)}L_{t_i,\Gamma_i}(0))sr_2}, g^{\frac{f(t)}{a}r_1}) \\ \Leftrightarrow e(H(w)^{r_1}, g^{sr_2}) &= e(H(q), g)^{\frac{r_1 r_2 s(f(t)L_{t,\Gamma_i}(0)+f(t_i)L_{t_i,\Gamma_i}(0))}{a}} \\ \Leftrightarrow e(H(w)^{r_1}, g^{sr_2}) &= e(H(q), g)^{r_1 r_2 s} \\ \Leftrightarrow e(H(w), g)^{r_1 r_2 s} &= e(H(q), g)^{r_1 r_2 s} \\ \Rightarrow H(w) &= H(q) \end{aligned}$$

Since H is a hash function, according to its collision-proofness, we can get $w = q$. This is the complete proof process.

6 Security Analysis

6.1 External Attack

In task matching, the task publisher/worker generates the ciphertext/trapdoor by adding random numbers, so that even the same keyword still gets different results and cannot be distinguished. In other words, the adversary will not be able to distinguish the ciphertext/trapdoor of any keyword and obtain the sensitive information.

6.2 Internal Attack

Similar to the external attack, nodes can not obtain the sensitive information. At the same time, since the publishing/matching results will be returned after consensus, it is guaranteed that the results received by the publishers/workers are correct. Furthermore, random addresses in the blockchain guarantee the anonymity of task publisher/worker identity information.

7 Performance Evaluation

In this section, we will implement the proposed scheme, evaluate it and compare it with related schemes (pMatch [16] and SEMEKS [15]). Table 2 provides a summary of the symbols used in the evaluation.

Table 2. Summary of evaluation notations

Notations	Description
E_1, E_2	Group exponentiation on G_1, G_2
P	Pairing on $\langle G_1, G_1 \rangle$
H_1	Hash operation $\{0,1\}^* \rightarrow G_1$
n	Number of publishers/workers
k_1, k_2	Number of keywords in requirement/query

7.1 Experimental Setting

In order to test our scheme, we implemented the relevant algorithms in the following environment.

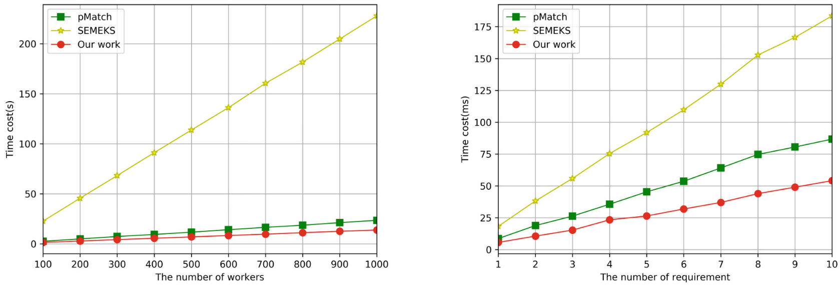
- Ubuntu 20.04.2 virtual machine is built under Windows 10 with 8 GB of memory.
- PBC v0.5.14, Go1.16 and PBC-Go-Wrapper.

In the comparison of each schemes, our implementation of pMatch, SEMEKS and our work are based on the elliptic curve SS512($|G_1| = 512\,bits$, $|G_2| = 1024\,bits$), which is a symmetric elliptic curve with base field 512-bit and embedding degree 2. The hash function H is in the PBC library.

7.2 Experimental Results

In our experiments, we compare with related schemes in the task matching algorithm.

System Initialization. System initialization is done by the key management center (KMC). It mainly includes keys generation. We set different numbers of registered workers n from 1000 to 10000 and measure the time consumption of the different schemes. The computational complexity of SEMEKS, pMatch and our work are $(8E_1 + n \cdot 16E_1)$, $(3E_1 + n \cdot 2E_1)$ and $((n+1) \cdot E_1)$, respectively. From Fig. 3(a), our scheme is much less time cost than pMatch and SEMKES. For example, when the number of workers is 10000, the different schemes are approximately 228 s, 24 s and 13 s, respectively.

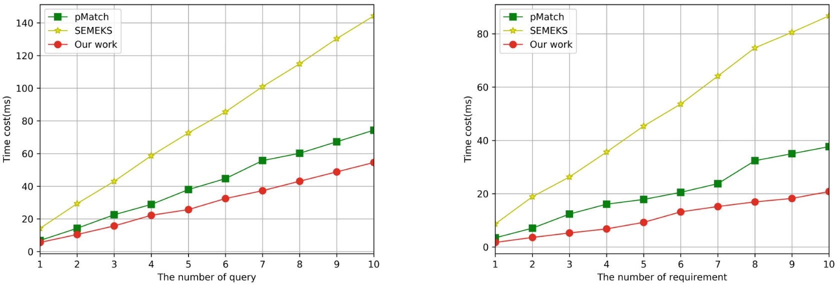


(a) system initialization (b) task publishing

Fig. 3. System initialization and task publishing

Task Publishing. In the task publishing, the task publisher calls **Enc** to complete the ciphertext generation. To measure the time cost of ciphertext generation, we set different numbers of keywords in requirement k_1 from 1 to 10. The computational complexity of SEMEKS, pMatch and our work are $(k_1 \cdot 9E_1)$, $(k_1 \cdot (5E_1 + H_1))$ and $(k_1 \cdot (2E_1 + H_1))$, respectively. We give the time cost of different schemes on ciphertext generation in Fig. 3(b). The time cost of our scheme is lower compared to the other schemes. For example, when the number of requirement is 10, the time cost of our scheme is about 54ms, while the time cost of other schemes are 184 ms and 87 ms.

Task Matching. In the task matching, task workers call **Trap** to generate trapdoors for task matching. Similar to the ciphertext generation, we also set different number of keywords in query k_2 from 1 to 10 for trapdoor generation. The computational complexity of SEMEKS, pMatch and our work are $(k_2 \cdot 12E_1)$, $(k_2 \cdot (4E_1 + H_1))$ and $(k_2 \cdot (2E_1 + H_1))$, respectively. We give the time cost of different schemes on trapdoor generation in Fig. 4(a). From the figure we can observe that our scheme has lower time cost than SEMEKS and pMatch. For example, when the number of keywords in query is 10, the time cost of our scheme is 54 ms and the other schemes are about 145 ms and 75 ms. Task matching is mainly done by the smart contract TMSC on the blockchain. Our experiments focus on time cost tests by executing **Match**. We set a different number of keywords in requirement from 1 to 10 k_1 under the trapdoor of being given a single keyword in query($k_2 = 1$). The computational complexity of SEMEKS, pMatch and our work are $(k_1 \cdot k_2 \cdot 5P)$, $(k_1 \cdot k_2 \cdot 4P)$ and $(k_1 \cdot k_2 \cdot 2P)$, respectively. Figure 4(b) shows that the time cost of our scheme is lower compared to SEMEKS and pMatch. For example, when the number of keywords in requirement is 10, the time cost of our scheme is about 20 ms, and the other schemes are about 110 ms and 38 ms.



(a) trapdoor generation

(b) task matching

Fig. 4. Task matching

8 Conclusion

In this paper, we design a multi-publisher/multi-worker task matching scheme in crowdsourcing based on the blockchain, in which task content is protected while enabling multi-publisher/multi-worker task matching. We analyzed the performance of our scheme from both theoretical and practical aspects. From the results, it is clear that the scheme is feasible.

References

1. Duan, Z., Li, W., Cai, Z.: Distributed auctions for task assignment and scheduling in mobile crowdsensing systems. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp. 635–644 (2017)
2. Duan, Z., Li, W., Zheng, X., Cai, Z.: Mutual-preference driven truthful auction mechanism in mobile crowdsensing. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 1233–1242 (2019)
3. Cai, Z., Duan, Z., Li, W.: Exploiting multi-dimensional task diversity in distributed auctions for mobile crowdsensing. IEEE Trans. Mob. Comput. **20**(8), 2576–2591 (2021)
4. Han, S., et al.: Location privacy-preserving distance computation for spatial crowdsourcing. IEEE Internet Things J. **7**(8), 7550–7563 (2020)
5. Chunqiang, H., Cheng, X., Tian, Z., Yu, J., Lv, W.: Achieving privacy preservation and billing via delayed information release. IEEE/ACM Trans. Netw. **29**(3), 1376–1390 (2021)
6. Song, D.X., Wagner, D.A., Perrig, A.: Practical techniques for searches on encrypted data, pp. 44–55 (2000)
7. Wang, C., Cao, N., Li, J., Ren, K., Lou, W.: Secure ranked keyword search over encrypted cloud data, pp. 253–262 (2010)
8. Yiming, W., Tang, S., Zhao, B., Peng, Z.: BPTM: blockchain-based privacy-preserving task matching in crowdsourcing. IEEE Access **7**, 45605–45617 (2019)
9. Bitcoin: A peer-to-peer electronic cash system. Social Science Electronic Publishing

10. Zhu, S., Li, W., Li, H., Tian, L., Luo, G., Cai, Z.: Coin hopping attack in blockchain-based IoT. IEEE Internet Things J. **6**(3), 4614–4626 (2019)
11. Yuwen, P., Chunqiang, H., Deng, S., Alrawais, A.: R^{2}PEDS: a recoverable and revocable privacy-preserving edge data sharing scheme. IEEE Internet Things J. **7**(9), 8077–8089 (2020)
12. To, H., Ghinita, G., Fan, L., Shahabi, C.: Differentially private location protection for worker datasets in spatial crowdsourcing. IEEE Trans. Mob. Comput. **16**(4), 934–949 (2017)
13. Jie, X., Lin, Z., Jun, W.: Privacy-preserving task-matching and multiple-submissions detection in crowdsourcing. Sensors **21**(9), 3036 (2021)
14. Boneh, D., Crescenzo, G.D., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. IACR Cryptology ePrint Archive, p. 195 (2003)
15. Kiayias, A., Oksuz, O., Russell, A., Tang, Q., Wang, B.: Efficient encrypted keyword search for multi-user data sharing. In: Askoxylakis, I., Ioannidis, S., Katsikas, S., Meadows, C. (eds.) ESORICS 2016. LNCS, vol. 9878, pp. 173–195. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45744-4_9
16. Shu, J., Yang, K., Jia, X., Liu, X., Wang, C., Deng, R.H.: Proxy-free privacy-preserving task matching with efficient revocation in crowdsourcing. IEEE Trans. Dependable Secur. Comput. **18**(1), 117–130 (2021)
17. Shu, J., Liu, X., Jia, X., Yang, K., Deng, R.H.: Anonymous privacy-preserving task matching in crowdsourcing. IEEE Internet Things J. **5**(4), 3068–3078 (2018)
18. Zhu, S., Cai, Z., Huafu, H., Li, Y., Li, W.: zkCrowd: a hybrid blockchain-based crowdsourcing platform. IEEE Trans. Industr. Inf. **16**(6), 4196–4205 (2020)
19. Chen Zhang, Yu., Guo, X.J., Wang, C., Hongwei, D.: Enabling proxy-free privacy-preserving and federated crowdsourcing by using blockchain. IEEE Internet Things J. **8**(8), 6624–6636 (2021)

Dropout-Based Ensemble Dual Discriminator for Cross-Domain Sentiment Classification

Xing Wei[1,3,4](✉), Xiuxiu Wang[1], Li Zhang[1], Lei Chen[2], Hui Luo[3], Di Wu[6], and Chong Zhao[5]

[1] School of Computer and Information, Hefei University of Technology, Hefei, China
weixing@hfut.edu.cn
[2] Institute of Intelligent Machines, HFIPS, Chinese Academy of Sciences, Beijing, China
[3] Intelligent Manufacturing Institute of Hefei University of Technology, Hefei, China
[4] Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Hefei, China
[5] Engineering Quality Education Center of Undergraduate School, Hefei University of Technology, Hefei, China
[6] School of Computer Science, Hefei Normal University, Hefei, China

Abstract. The main task of Cross-Domain Sentiment Classification is to train a well-performing classification model by using labeled source domain data, and then transfer the model to unlabeled target domain data, thereby solving the expensive labor consumption and domain shift caused by a large number of labels resulting performance degradation. Most mainstream adversarial domain adaptation methods are based on a single discriminator, which ignores the uneven distribution of labels between domains and multiple modalities of data and tends to cause negative transfer and poor generalization performance. We propose a Dropout-based ensemble dual discriminator for Cross-Domain Sentiment Classification. We functionally decouple the single discriminator by using two forms of text data, and replace it with a positive sentiment discriminator and a negative sentiment discriminator. A dynamic set of discriminators will be obtained by random deactivation of the discriminator network neurons, then the feature extractor has to extract richer and more realistic domain-invariant features to fool the discriminator and mitigate the mode collapse phenomenon. To solve the problem of class imbalance in a large number of unlabeled data samples, we use mutual information maximization to train sentiment classifiers to ensure that label predictions are distributed in a reasonably balanced state. We conduct full experiments on the Amazon and Airlines datasets. Experiments showed that our proposed model achieves state-of-the-art cross-domain sentiment classification performance.

Keywords: Cross-domain sentiment classification · Dual discriminator · Dropout · Mutual information maximization

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 526–538, 2022.
https://doi.org/10.1007/978-3-031-19214-2_44

1 Introduction

The textual comment data generated in various fields such as shopping platforms, forums, and social software have grown exponentially with the development of the internet and social media. Emotional categories are people's emotional cognition and evaluation of products, services, social public opinion, etc., and are generally divided into two categories: positive and negative. Sentiment classification methods based on machine learning are generally supervised learning methods. Sentiment classification models are trained on labeled data, and then use the model to automatically classify sentiment categories of review data [14].

Generally, the domain with labeled data is called the source domain, and the new domain to be classified is called the target domain. In traditional text sentiment classification, the data between the training and testing domains is independent and identically distributed. However, there are distribution differences between different domains under realistic conditions, and the classifier trained in the source domain is directly used for the sentiment classification task of the target domain, resulting in a sharp drop in performance. Recalibrating a large amount of training data for a new domain is undoubtedly time-consuming, labor-intensive, and even impractical. Cross-domain sentiment classification aims to train a source domain model with good classification performance using labeled source domain data and transfer the model to the unlabeled target domain so that it performs well on the target domain as well [4,6].

Cross-domain sentiment classification tasks mainly revolve around reducing the domain offset between the source and target domains. It is currently divided into two categories, the first category is the method based on feature selection [5,11]. According to the discrete characteristics of text data, the pivot is used to establish a bridge between source and target domains. The method requires manual selection of shared feature words between domains, and its use of linear classifiers will bring huge labor consumption and cannot achieve good classification results. The second category is the methods based on domain adversarial learning [1,2,20,21]. It constructs an adversarial relationship between the discriminator and the feature extractor to reduce the distribution difference between domains. After iterative training, the discriminator cannot discriminate its real domain origin for the input data, and the feature extractor captures the largest common features between domains [7]. However, there are still two challenges:1) The design of discriminator does not take into account the impact of multiple modalities of the data and a large number of network parameters on the model speed, which confuse the domain origin of the data and reduce the discriminative power and computational speed of the discriminator. It also poses a greater risk of negative transfer and modal collapse to the model. 2) Unstable label prediction. In practical application scenarios, we often only have a small amount of labeled data and a large number of unlabeled data samples also have the problem of category imbalance. It increases the negative migration of the model, poses a great challenge to our classifier performance, and severely limits the generalization of adversarial domain adaptation methods for this class of natural language processing tasks (NLP).

The performance of any adversarial network depends heavily on the capabilities of the discriminator. To resolve the above problems, we propose a Dropout-based ensemble dual discriminator for Cross-Domain Sentiment Classification. The model uses Dropout-based ensemble dual discriminator (positive sentiment discriminator and negative sentiment discriminator), utilizing a feature extraction module for adversarial learning with the discriminator. Then using gradient reversal (GRL) [7] for parameter updates, the feature extractor learns a continuous gradient distribution of the data. The sentiment classifier uses standard cross-entropy and mutual information maximization to train the labeled data and unlabeled data respectively and then obtains a sentiment classifier with more accurate label prediction.

Our main contributions in this work are summarized as follows:

1. We use a dual discriminator to decompose the burden of a single discriminator by setting a positive sentiment discriminator and a negative sentiment discriminator to be responsible for discriminating the domain origin (source or target domain) of the data respectively. It achieves high cohesiveness and low coupling of discriminators and avoids overconfidence of a single discriminator in judging the source of the sample domain. Fine-grained alignment is performed on the same classes in different domains to maximize distribution matching and increasing positive transfer.
2. We incorporate Dropout in a multi-adversarial network, by dropping out a certain number of neurons from our discriminator to get a dynamic ensemble discriminator. It enables our model to obtain a continuous gradient distribution of data without increasing model parameters, reducing prediction variance, eliminating overfitting, mitigating mode collapse, and capturing fine-grained domain common features with richer corresponding polarities.
3. We use mutual information maximization techniques to constrain the sentiment classifier's label predictions on unlabeled data so that they do not unreasonably biased towards a certain class. We only need a small amount of labeled source domain data.

2 Related Work

2.1 Text Sentiment Classification

With the development of deep learning in the field of NLP, the mainstream models of text sentiment classification are Transformer [16] and BERT [3]. Transformer uses self-attention and fully connected layers for parallel computing, which can train all words simultaneously and improve the training speed. BERT is a pre-trained language model trained with a large amount of unlabeled text in an unsupervised manner. Its architecture is an encoder built by Transformer, which can directly handle various NLP tasks. These classification models have excellent performance for sentiment classification in labeled domains, but we cannot directly use these models for cross-domain sentiment classification tasks.

2.2 Cross-Domain Sentiment Classification

In the early research on cross-domain sentiment classification, the typical way is to use some statistical measures to quantify the difference in distribution, such as KL divergence [15], maximum mean discrepancy (MMD) [13], etc. The approach is suitable for cross-domain tasks with significant differences in the edge distributions of the two domains.

Adversarial-based domain adaptation has made good progress in domain adaptation tasks [18,20]. Ganin et al. [7] proposed Adversarial Domain Adaptive Neural Network (DANN) to apply generative adversarial ideas to transfer learning for the first time. It constructs the adversarial training of the discriminator and the feature extractor, and cooperates with the gradient flip layer in the process of backpropagation, so that the discriminator cannot identify the domain source of the sample, so as to extract the domain invariant. Volpi et al. [17] augmented the feature representation in a noisy manner to make the training process more robust and shared a feature extractor for both source and target domains. The feature extractor only learned the point distribution of the data. A single discriminator only pays attention to the coarse-grained information of sentiment polarity and easily leads to overconfident reasoning and decision-making, which brings challenges for the feature extractor to learn domain-invariant features. Saito et al. [12] used two independent classifiers to indirectly optimize the decision boundary of the domain by iteratively maximizing and minimizing the discriminative conflicting regions of the two classifiers. Although the above method decomposes the discriminative pressure of a single discriminator, what our feature extractor learns is still not a continuous gradient distribution but two gradient values. Meanwhile, it also involves an increase in the number of network parameters, which makes the model complex and computational overhead severely limits the generalization performance of adversarial domain adaptation methods for cross-domain sentiment classification tasks.

3 Methods

In the cross-domain sentiment classification task, we define N_{s1} labeled samples from a source domain $D_{s1} = \left\{x_i^{s1}, y_i^{s1}\right\}_{i=1}^{N_{s1}}$, and $y_i^{s1} = \begin{cases} 0, x_i^{s1} \in neg \\ 1, x_i^{s1} \in pos \end{cases}$, *neg* represents a collection of negative emotional comments, *pos* represents a collection of positive emotional comments. N_{s2} unlabeled samples from a source domain $D_{s2} = \left\{x_i^{s2}\right\}_{i=1}^{N_{s2}}$. N_t unlabeled samples from a target domain $D_t = \left\{x_j^{t}\right\}_{j=1}^{N_t}$. $D_s = D_{s1} \cup D_{s2}$, the distributions of D_s and D_t are different due to the domain discrepancy.

As shown in Fig. 1, we concatenate the word2vec [10] word embedding and the BERT context embedding to obtain the input vector e_i. The feature vector h_i is obtained by using BERT for feature extraction on the input vector e_i. The sentiment classifier network C performs label prediction on the feature vector h_i to obtain $\hat{y}$ ($\hat{y}^0$ and $\hat{y}^1$ represent the sentiment category as negative and positive). According to the prediction $\hat{y}$ of the sentiment classifier, the sample data is

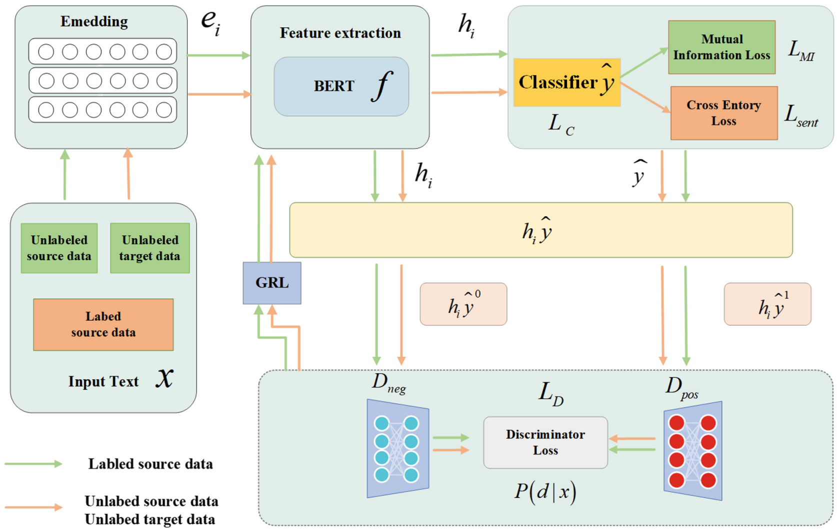


Fig. 1. Dropout-based ensemble dual discriminator for Cross-Domain Sentiment Classification. The feature extraction module f conducts adversarial training with the Dropout ensemble discriminator, and the parameters are updated by gradient reversal (GRL). The sentiment classifier uses standard cross-entropy and mutual information maximization to constrain training on labeled and unlabeled data.

respectively sent to the specified Dropout-based ensemble discriminator (D_{neg} represents the negative sentiment discriminator, D_{pos} represents the positive sentiment discriminator) combined with the corresponding feature vector d_i for domain discrimination and gradient reversal for parameter update of the feature extractor.$P\left(d|x\right)$ is the predicted distribution of the discriminator for the domain to which the input data belongs.

3.1 Sentiment Classifier Module

We use mutual information technology to constrain the relationship between unlabeled samples and predicted labels, as shown in (Eq. 1).

$$I\left(X,Y\right) = H\left(Y\right) - H\left(Y|X\right) \tag{1}$$

X is the unlabeled sample, and Y is the label prediction result of our sentiment classifier. The larger $H\left(Y\right)$, the richer the label prediction; The smaller $H\left(Y|X\right)$, more accurate the prediction for the unlabeled sample label. Mutual information maximization $I(X,Y)_{\max}$ is to maximize $H\left(Y\right)$ while minimizing $H\left(Y|X\right)$.

The unlabeled data loss function L_{MI} is shown in (Eq. 2), θ_f is the parameter of the feature extractor, and θ_c is the parameter of the sentiment classifier:

$$L_{MI}\left(\theta_f,\theta_c\right) = E_y\left[\log p_\theta\left(y\right)\right] - E_x\left[\sum_y p_\theta\left(y|x\right)\log p_\theta\left(y|x\right)\right] \tag{2}$$

$p_\theta(y|x)$ denotes the logits of input, $p_\theta(Y)$ represents the distribution of predicted target labels. When calculating we use the mean $p_\theta(y|x)$ as an approximation $p_\theta(Y)$. In the first iteration of training $y \sim p_\theta(Y)$, from the second iteration of training $x \sim p_\theta(X)$.

For labeled data, we use the standard cross-entropy to calculate the classification loss L_{sent}, as shown in (Eq. 3).

$$L_{sent}(\theta_f, \theta_c) = \frac{1}{N_{s1}} \sum_{x_i \in D_{s1}} L_1(C(f(x_i)), y_i) \tag{3}$$

f is the feature extraction network, C is the sentiment classifier network, and L_1 is a standard cross-entropy function.

In summary, the loss L_C of our sentiment classifier is (Eq. 4).

$$L_C(\theta_f, \theta_c) = L_{sent}(\theta_f, \theta_c) + L_{MI}(\theta_f, \theta_c) \tag{4}$$

3.2 Dropout-Based Ensemble Dual Discriminator Module

Set up a group of Dropout-based ensemble discriminator (D_{neg} and D_{pos}), each discriminator focuses on learning more fine-grained and richer sentiment features in a specified category. The relationship between $\theta_f, \theta_c, \theta_d$ and the loss of the discriminator are shown in (Eq. 5).

$$\begin{aligned} L_D(\theta_f, \theta_c, \theta_d) = &-\lambda \sum_{x_i \in D_s \cup D_t} L_d(D_{neg}(f(x_i)), d_i) \\ &-\lambda \sum_{x_i \in D_s \cup D_t} L_d(D_{pos}(f(x_i)), d_i) \end{aligned} \tag{5}$$

$$\left(\hat{\theta}_f, \hat{\theta}_c\right) = \arg\min_{\theta_f, \theta_c} L\left(\theta_f, \theta_c, \hat{\theta}_d\right) \quad \left(\hat{\theta}_d\right) = \arg\max_{\theta_d} L\left(\hat{\theta}_f, \hat{\theta}_c, \theta_d\right)$$

L_d is the binary cross-entropy loss between the discriminator output and the desired output, θ_d is the network parameter of the discriminator, $d_i = 0$ represents the sample comes from the source domain, $d_i = 1$ represents the sample comes from the target domain, λ is the trade-off parameter between the two objectives.

As shown in Fig. 2, we adopt the Bernoulli dropout method [8] to deactivate a certain number of neurons from the discriminator network with a certain probability, and then the discriminator for each class becomes a dynamic ensemble model. Randomly deactivating neurons ensures that the output of our discriminator does not depend on connections between specific neurons, and each of our neurons depends on the aggregation of several other neurons, facilitating model generalization.

3.3 Training

In the cross-domain sentiment classification task, our ultimate goal is that the sentiment classifier can classify the target domain well.

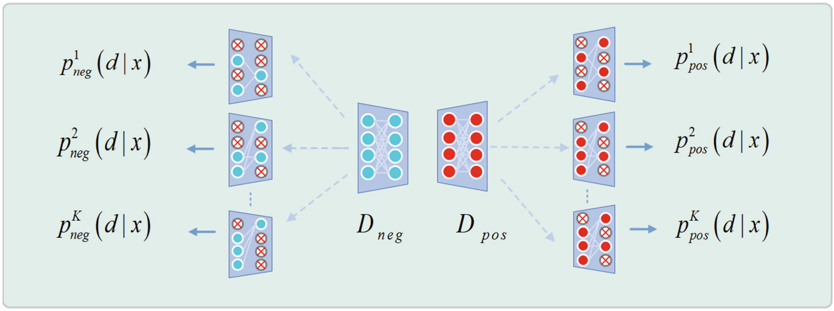


Fig. 2. Dropout-based ensemble dual dropout discriminator structure. $p_{neg}^{K}(d|x)$ and $p_{pos}^{K}(d|x)$ are the predicted distribution of the kth discriminator for the domain to which the input data belongs in the dynamic ensemble discriminator D_{neg} and D_{pos}.

We minimize $\Pr_{(x,y)\sim q}[C(f(x)) \neq y]$ to reduce distribution variance and increase positive transfer through multi-adversarial training. The training process is shown in Algorithm 1 ,our objective function is:

$$L_{\text{total}} = L_C + L_D \tag{6}$$

4 Experiment

4.1 Dataset

To evaluate the effectiveness of the proposed model, we conduct experiments on the Amazon review dataset which has been widely used for cross-domain sentiment classification task. [5] and the Airlines review dataset (https://github.com/quankiquanki/skytrax-reviews-dataset). As shown in Table 1, the Amazon dataset contains reviews in four domains, books (B), DVDs (D), electronics(E), and kitchen products (K). The Airlines dataset contains reviews of airlines. It is far from the domain in the Amazon dataset, which makes the cross-domain problem more challenging. we randomly sample 1000 positive reviews and 1000 negative reviews from the aviation dataset as labeled data, and the remaining data are de-labeled as unlabeled data.

Table 1. Statistics of Amazon review dataset and Airlines review dataset. "pos : neg" denotes the ratio of unlabeled positive samples over unlabeled negative samples on that domain.

Domains	Labeled	Unlabeled	pos : neg
Books	2000	6000	6:1
DVD	2000	30000	7:1
Electronics	2000	10000	3:1
Kitchen	2000	10000	4:1
Airlines	2000	30000	1:1

Algorithm 1 Training strategy of cross-domain sentiment classification with Dropout-based ensemble dual discriminator

Input: Labeled source data $D_{s1} = \left\{x_i^{s1}, y_i^{s1}\right\}_{i=1}^{N_{s1}}$, unlabeled source data $D_{s2} = \left\{x_i^{s2}\right\}_{i=1}^{N_{s2}}$, unlabeled target data $D_t = \{x_i^t\}_{i=1}^{N_{s2}}$, minibatch size m, training step n;

Output: Target domain sentiment label y_t;

```
repeat
    Sample minibatch $\left\{x_{s1}^i, y_{s1}^i\right\}_{i=1}^m$ ,$\left\{x_{s2}^i\right\}_{i=1}^m$, from $D_{s1}$,$D_{s2}$,$D_t$;
    $e_i = [word2vec\,(s_i)\,; BERT_{s_i}]$ ▷ input vector;
    for j = 1, ... , n do
        $h_s, h_t = BERT\,(e_s, e_t)$ ▷ feature vectors;
        update parameters $\theta_c$ to minimize $L_C\,(\theta_f, \theta_c)$    ▷ (Equation4);
        if $\hat{y} = 0$ then
            $\hat{d} = D_{neg}\,(f\,(x_i))$;
        else
            $\hat{d} = D_{pos}\,(f\,(x_i))$;
        end if
        update parameters $\theta_d$ to minimize $L_d\,(\theta_f, \theta_c, \theta_d)$    ▷ (Equation5);
        update parameters$\theta_f \xleftarrow{GRL}$ ◁ (Equation5);
        update parameters$\theta_f, \theta_c, \theta_d$    ▷ (Equation6);
    end for
until $\theta_f, \theta_c, \theta_d$ converge;
$y_t = f_C\,(d_t)$
```

Given the computation cost and the scale of the datasets, we pick 12 pairs of domains as source and target respectively to evaluate the proposed method, namely B->D,B->E,B->K,D->B,D->E,D->K,E->B,E->D,E->K,K->B,K->D and K->E. In each of the settings, we randomly choose 500 samples of the source labeled data for development (dev set), and use the rest of 1500 source labeled data, along with unlabeled data from both domains for training. All labeled data from the target domain are used for testing.

4.2 Baselines

We mainly compare with EADA [21] and DAAT* [4] methods based on BERT as a feature extractor. The traditional methods DANN [2], HAGAN-C [12], and HATN [19] are also included. We train BERT [3] using labeled data from the source domain and then test it directly on the target domain data.

4.3 Experimental Parameter Settings

As for training strategy, we train the models for 10 epochs, using the AdamW [9], optimizer with learning rate 2×10^{-5}. The value of Dropout is set according to different cross-domain pairs, where B->E, D->K, K->E is 0.5, E->K is 0.4, and the rest are 0.45. In the gradient reversal layer of our model, we define the training progress as $p = \frac{t}{T}$, where t and T are the current training step and the maximum training step respectively, and the adaptation rate λ is increased following $\lambda = \frac{2}{1+\exp(-\gamma p)}$. We choose γ from 0.25, 0.5, 0.75, 1.0 on different domain settings.

4.4 Experimental Results and Analysis

Comparison with Baselines. The performance index of the evaluation model is the accuracy of text sentiment classification. Table 2 presents the accuracy of each method on the 12 DA tasks. Our proposed method can outperform the previous state-of-the-art method DAAT* by 0.81% on average. Relying on the strong ability of BERT to extract universal high-quality features, greatly improves overall cross-domain sentiment classification performance. Despite the high baseline accuracy given by BERT and DAAT*, our model outperforms them by 2.68% and 0.81% on average.

Table 2. Shows the performance of our model as well as the baselines on the benchmarks.

S>T	DANN	AMN	HAGAN-C	EADA	BERT	DAAT*	Ours
B->D	83.25	81.32	84.60	86.05	88.98	89.70	**90.85**
B->E	77.42	80.07	80.12	89.35	86.15	89.57	**91.50**
B->K	78.50	81.00	82.00	89.65	89.05	90.75	**92.50**
D->B	81.60	81.52	81.69	88.10	89.40	90.86	**91.00**
D->E	79.80	80.00	80.99	87.15	86.55	89.30	**90.25**
D->K	80.80	83.88	81.50	89.20	87.53	90.50	**91.65**
E->B	77.60	77.80	79.23	85.25	86.50	88.91	**89.75**
E->D	77.90	77.51	80.65	85.35	87.95	**90.13**	88.15
E->K	83.95	87.10	84.99	90.50	91.6	93.18	**94.60**
K->B	76.52	79.37	78.99	81.20	87.55	87.98	**89.00**
K->D	78.65	80.03	80.91	80.35	87.30	**88.81**	88.60
K->E	85.26	81.97	85.23	85.11	90.45	91.72	**93.30**
Avg	80.10	80.96	81.56	86.44	88.25	90.12	**90.93**

To demonstrate the better robustness of our model, we additionally select the Airlines reviews dataset with a large distribution differs from the electronics (E). In this experiment, we choose typical DANN and BERT models to be

Table 3. Cross-domain sentiment classification accuracy by different models on the Airlines review dataset.

S>T	DANN	BERT	Ours
E>A	78.00	86.50	**87.50**
A>E	80.40	88.65	**89.08**

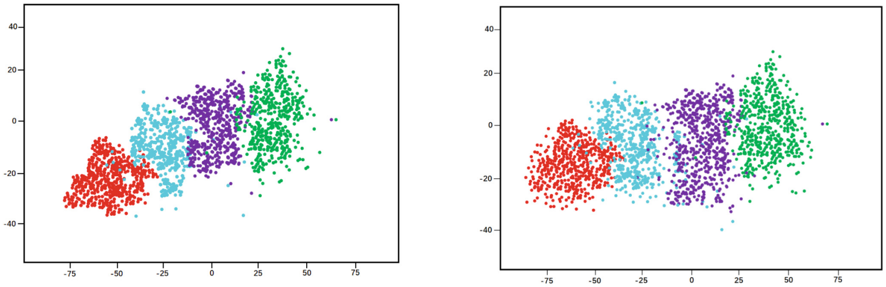
experimentally validated together with our model. The experimental results are shown in Table 3. We can find that our model is better than other models in A->E or E->A, which also verifies the robustness of our model.

Ablation Study. To better demonstrate the impact of improvements made in our model on model performance, we conduct ablation experiments on the type and number of discriminators. According to Table 4, we can see that the accuracy of sentiment classification is gradually improving from top to bottom. It illustrates the Dropout-based ensemble single discriminator achieves 5.35% and 2.85% higher scores than the traditional single discriminator in B->E and K->E.

Table 4. Evaluation results on different discriminator

Single	Dual	Dropout-based - single	Dropout-based - dual	Accuracy	
				B->E	K->E
✓				86.15	90.45
	✓			89.89	91.25
		✓		90.90	92.15
			✓	91.50	93.30

It is worth noting that the classification accuracy using the Dropout-based ensemble dual discriminator is all higher than other type discriminators. It suggests that we set the discriminator of a specific sentiment category according to the two modalities of the data, which can fine-grained alignment of the same category in different domains, maximize distribution matching, and increase positive transfer. The classification accuracy of the Dropout-based ensemble single discriminator is 1.01% higher than the ordinary dual discriminator. It illustrates that the Dropout-based ensemble discriminator does not rely too much on local features when discriminating the domain source of the data, so our feature extractor does not rely on a specific type of discriminator to learn the trick of deceiving the discriminator and learns that the data is more real rich domain-invariant features.



(a) Basline-BERT.

(b) Ours.

Fig. 3. t-SNE projection of (a) BERT-base hidden feature, (b) our model hidden feature. Note that the margin between the positive cluster and negative cluster on the target domain becomes clearer from left to right. The red and purple dots respectively represent the feature of positive and negative samples on the source domain, and the blue and green dots represent those on the target domain. (Color figure online)

Visualization. To visualize the effect of our model, we use different models to visualize the distribution of positive and negative samples in the source and target domains by t-SNE in B->K. As shown in Fig. 3. Figure 3(a) uses the BERT model for adaptation. We can find that the positive and negative polarity classification in the source domain is better, but the positive and negative samples in the target domain are confused, and the boundaries between domains are confused. Figure 3(b) uses our model for adaptation, not only the boundaries of positive and negative samples in the target domain are clear, but also the domains are separable. This shows that we use the unlabeled data of the mutual information maximization technique for constraint training to make our sentiment classification model prediction distribution more balanced, expand the boundary of our sentiment polarity classification, and make the classes better separated. The training effect of the Dropout-based ensemble dual discriminator allows the feature extractor to capture more abundant common features between domains, and also makes the domains separable.

5 Conclusion

In this paper, we propose use of a Dropout-based ensemble dual discriminator promotes domain positive transfer. The use of a sampling-based ensemble results in an improved discriminator without increasing the number of parameters. Our proposed model beats strong baselines and visualization results also show the efficacy of our model. Extensive ablation studies unveil how to label distribution shift may interact with our model. Furthermore, we will extend our work to larger real-world datasets on new emerging domains with less labeled data.

Acknowledgements. This work was supported by Anhui Provincial Key R&D Program (202004a05020004), Joint Fund of Natural Science Foundation of Anhui Province in 2020 (2008085UD08), Intelligent Networking and New Energy Vehicle Special Project of Intelligent Manufacturing Institute of HFUT (IMIWL2019003, IMIDC2019002).

References

1. Cao, Y., Xu, H.: SATNet: symmetric adversarial transfer network based on two-level alignment strategy towards cross-domain sentiment classification (student abstract). In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 13763–13764 (2020)
2. Chen, X., Sun, Y., Athiwaratkun, B., Cardie, C., Weinberger, K.: Adversarial deep averaging networks for cross-lingual sentiment classification. Trans. Assoc. Comput. Linguist. **6**, 557–570 (2018)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Du, C., Sun, H., Wang, J., Qi, Q., Liao, J.: Adversarial and domain-aware BERT for cross-domain sentiment analysis. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4019–4028 (2020)
5. Fang, F., Dutta, K., Datta, A.: Domain adaptation for sentiment classification in light of multiple sources. INFORMS J. Comput. **26**(3), 586–598 (2014)
6. Fei, H., Li, P.: Cross-lingual unsupervised sentiment classification with multi-view transfer learning. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5759–5771 (2020)
7. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning, pp. 1180–1189. PMLR (2015)
8. Hara, K., Saitoh, D., Shouno, H.: Analysis of dropout learning regarded as ensemble learning. In: Villa, A.E.P., Masulli, P., Pons Rivero, A.J. (eds.) ICANN 2016. LNCS, vol. 9887, pp. 72–79. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44781-0_9
9. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam (2018)
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
11. Pan, S.J., Ni, X., Sun, J.T., Yang, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: Proceedings of the 19th International Conference on World Wide Web, pp. 751–760 (2010)
12. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3723–3732 (2018)
13. Sejdinovic, D., Sriperumbudur, B., Gretton, A., Fukumizu, K.: Equivalence of distance-based and RKHS-based statistics in hypothesis testing. Ann. Stat. 2263–2291 (2013)
14. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1422–1432 (2015)
15. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)

16. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
17. Volpi, R., Morerio, P., Savarese, S., Murino, V.: Adversarial feature augmentation for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5495–5504 (2018)
18. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2517–2526 (2019)
19. Zhang, K., Zhang, H., Liu, Q., Zhao, H., Zhu, H., Chen, E.: Interactive attention transfer network for cross-domain sentiment classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5773–5780 (2019)
20. Zhang, Y., Miao, D., Wang, J.: Hierarchical attention generative adversarial networks for cross-domain sentiment classification. arXiv preprint arXiv:1903.11334 (2019)
21. Zou, H., Yang, J., Wu, X.: Unsupervised energy-based adversarial domain adaptation for cross-domain text classification. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 1208–1218 (2021)

CNsum: Automatic Summarization for Chinese News Text

Yu Zhao[1], Songping Huang[1], Dongsheng Zhou[2], Zhaoyun Ding[1](✉), Fei Wang[1], and Aixin Nian[1]

[1] Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410000, China
zyding@nudt.edu.cn

[2] National and Local Joint Engineering Laboratory of Computer Aided Design, School of Software Engineering, Dalian University, Dalian 116000, China

Abstract. Obtaining valuable information from massive data efficiently has become our research goal in the era of Big Data. Text summarization technology has been continuously developed to meet this demand. Recent work has also shown that transformer-based pre-trained language models have achieved great success on various tasks in NLP. Aiming at the problem of Chinese news text summary generation and the application of Transformer structure on Chinese texts, this paper proposes a Chinese news headline generation model CNsum based on Transformer structure, and tests it on Chinese datasets such as THUCNews. The results of the conducted experiments show that CNsum achieves better ROUGE, BLEU and BERTScore scores than the baseline models, which verifies the outperformance of the model.

Keywords: Abstractive summarization · Pre-trained language model · Seq2Seq · Chinese news headlines

1 Introduction

In recent years, the rapid development of information technology has led to an exponential increase in the amount of data, and more and more data appear on the Internet in various forms. But not all data can directly provide the information we want to obtain, which requires us to invest a lot of time and energy to understand massive text data. How to quickly filter and obtain information from massive texts has become an urgent problem to be solved. The task of the text summarization is to condense long documents into short summaries while preserving the important information and meaning of the documents [1]. Generating news headlines is a typical application field of text summarization techniques. Simple news classification has been difficult to meet the needs of news reading because of misleading headlines. The application of automatic text summarization technology has provided solutions, which could greatly improve the efficiency of news reading and selecting valuable information from the massive news.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 539–547, 2022.
https://doi.org/10.1007/978-3-031-19214-2_45

2 Related Work

At present, the research on automatic text summarization technology in academia and industry has made great progress, and good results have also been achieved in practical applications. The research on automatic text summarization technology begins with the method based on word frequency statistics, trying to find the key words and central sentences in the text through statistical technology, and then combine them to achieve text summarization [2]. H. P. Luhn [3] calculated the relative importance through the statistical information obtained from word frequency and distribution, becoming the first automatic text summarization system.

Automatic text summarization technology mainly includes two types: extractive text summarization and adstractive text summarization [4]. The former mainly extracts keywords and sentences in text data through some scoring criteria to generate summaries. The abstractive summarization method is closer to the understanding of text summarization and it mainly generates a new abstract based on the learning of the source text. At present, most abstractive summarization models are based on the Seq2seq framework, which was first proposed by Sutskever et al. [5] and Cho et al. [6]. It is mainly composed of an encoder and a decoder and it performed well in abstractive summarization tasks.

The application of deep learning methods, especially Transformers [7], has greatly improved the performance of automatic text summarization. And related researchs and comparisons proved that the use of neural networks had better performances [4]. However, almost all pre-trained language models currently are based on English, and these pre-trained language models' performances in Chinese are not as effective as their in English [8]. Moreover, research based on Chinese text summaries has developed slowly in recent years. Based on this and inspired by GPT2-chinese [21], this paper studies the performance of the Transformer-based pre-trained language model in generating Chinese news headlines.

3 Model

We proposed CNsum in this paper and it mainly includes two stages: the first stage is to realize the encoder output based on Bert's preprocessing method of Chinese news texts, that is, to encode the Chinese text and convert the Chinese news text into processable sequences; the second-stage decoder is composed of GPT-2 parts. With the help of its powerful text generation ability, it can generate Chinese news headlines. Figure 1 shows the construction of CNsum.

3.1 BERT-Based Encoder

Language model (LM) is a basic concept in natural language processing. The language model task is also the core problem in the field of NLP. After using the language model to process the text data related to the natural language, a

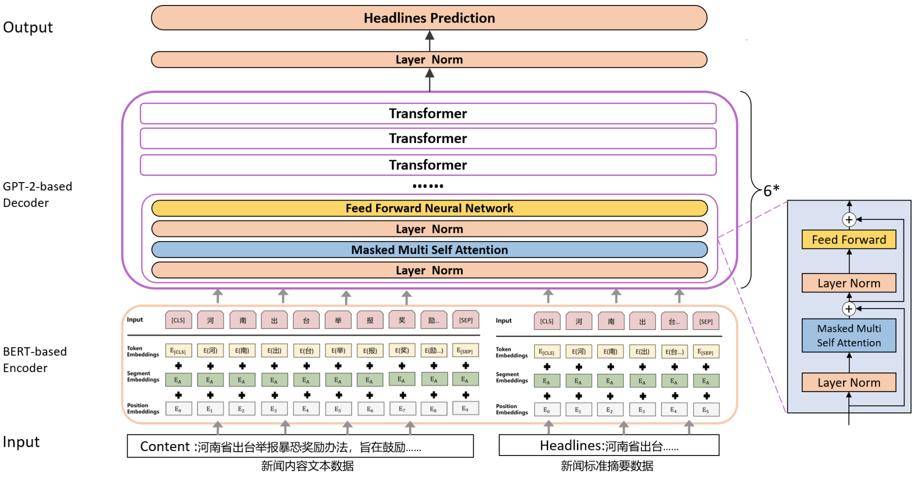

Output
Headlines Prediction
Layer Norm
GPT-2-based Decoder
Transformer
Transformer
Transformer
••••••
Feed Forward Neural Network
Layer Norm
Masked Multi Self Attention
Layer Norm
6*
Feed Forward
Layer Norm
Masked Multi Self Attention
Layer Norm
BERT-based Encoder
Input
[CLS] 河 南 出 台 举 报 奖 励... [SEP]
Token Embeddings
E[CLS] E(河) E(南) E(出) E(台) E(举) E(报) E(奖) E(励...) E[SEP]
Segment Embeddings
EA
Position Embeddings
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9
Input
[CLS] 河 南 出 台... [SEP]
Token Embeddings
E[CLS] E(河) E(南) E(出) E(台...) E[SEP]
Segment Embeddings
EA
Position Embeddings
E0 E1 E2 E3 E4 E5
Input
Content :河南省出台举报暴恐奖励办法，旨在鼓励......
新闻内容文本数据
Headlines:河南省出台......
新闻标准摘要数据

Fig. 1. The basic architecture of CNsum is a Seq2Seq model composed of encoder-decoder. The encoder mainly uses the BERT pre-trained language model to complete the encoding. The decoder is based on the GPT-2 model and is composed of 6 layers of Transformer-decoder stacked to realize the generations. The CNsum model is trained on the NLPCC2017 dataset.

language representation that can be processed by a computer can be obtained, which is convenient for the processing of the text data. The calculation of probability of the sequence occurrence is shown in Eq. (1).

$$p(S) = p(w_1, w_2, w_3, \ldots, w_n) = \prod\nolimits_i^n p(w_i | w_(i - n + 1), \ldots, w_(i - 1)) \qquad (1)$$

The BERT pre-training language model uses a bidirectional Transformer as an encoder for feature extraction, and based on the attention mechanism, it can better extract text information features. The two-way language representation of the system provides high-quality textual data information for downstream tasks. The Transformer coding unit mainly includes two parts, namely the self-attention mechanism and the feed-forward neural network. The self-attention mechanism can pay attention to the internal correlation of data or features, and is less dependent on external information, which is an effective capture method. The feedforward neural network is composed of two fully connected layers, which are mainly used to strengthen the nonlinear ability, learn more abstract features, and enhance the performance of the model.

$$Attention(Q, K, V) = softmax((QK^T)/\sqrt{d_k}))V \qquad (2)$$

As Eq. (2) shows, the input part of the self-attention mechanism includes Query vector (Q), Key vector (K) and Value vector (V). The Query vector and Key vector of text data are multiplied to obtain QK^T, and then $\sqrt{d_k}$ is used to ensure that the obtained results meet the specifications, then input the results to the softmax layer for normalization to obtain the probability of the text data, then obtain the word vector representation in the text data. Therefore, the BERT-based encoder can better complete the processing of Chinese news text data, and input the sequence information which contains embedding vector sequence and position encoding into CNsum's decoder.

3.2 GPT-2-Based Decoder

The GPT-2 model has outperformance on text generation. Based on the powerful text generation capability of GPT-2, the model proposed in this paper is composed of the decoder part of the multi-layer one-way Transformer, so the model only considers the influence of the words on the left side of the position of the word to be predicted on the predicted word when processing text data, and calculates self-attention based on these. The researchers [7] found that in Transformer, the Query vector (Q), the Key vector (K) and the Value vector (V) were first linearly transformed, and then input to the scaled dot product Attention. After multiple calculations, the results can be stitched together to achieve better results. The effect, that is, the realization of multi-head self-attention allows the model to learn relevant information in different dimensions. So similar to BERT, it is multi-heads self-attention that the model need to calculate when processing text information to Self-attention mechanism.

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O \tag{4}$$

As Eq. (3) and Eq. (4) show, $head_i$ represents NO.i self-attention, $W_i^Q \in R^(d_{model} \times d_k)$, $W_i^K \in R^(d_{model} \times d_k)$, $W_i^V \in R^(d_{model} \times d_v)$, $W^O \in R^{(hd_v \times d_{model})}$. The model need to calculate 8 self-attentions, which means the number of heads is 8. For each self-attention calculation, $d_k = d_v = d_model/h = 64$.

$$FFN(x) = max(0, xW_1, +b_1)W_2 + b_2 \tag{5}$$

Using the Mask can mask the information of all the data to the right of the current calculation position when calculating the self-attention, which can enhance the model's attention to the information of the current position. Using the Layer Norm layer can stabilize the distribution of data features, thereby accelerating the convergence of model training. In the Feed Forward Neural Network layer, the RELU activation function is used. It calculates the loss between news headlines and generated headlines to train the model. And in view of resource consumption, we only use a 6-layer network based on the existing model.

4 Experiments

4.1 Datasets

NLPCC2017: Provided by the 2017 CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC2017), including 49500 news texts in the training set and 500 news texts in the test and validation set. Articles obtained from news websites such as Sina and summaries of experts. It is used for model training and testing in the experiment.

SogouCS: Compiled by Sogou Lab and came from Sohu News. In this experiment, 500 samples are randomly selected to be used to test the performance of the model.

LCSTS: It was organized by Hu et al. [10] and created the dataset based on news abstracts published by news media on Weibo. It has the characteristics of short length and more noise. 300 samples are randomly selected as the test dataset.

THUCNews: Tsinghua News (THUCNews) is compiled by the NLP Lab of Tsinghua University. It is filtered and generated according to the historical data of the Sina News. It is used for model training and testing in the experiment.

4.2 Evaluation Metrics

ROUGE: ROUGE [19] is a summary evaluation tool based on recall statistics. It can reflect the coverage of the abstract to the news text.

$$ROUGE - N = \frac{\sum_{S \in RS} \sum_{g_n \in S} Count_m\left(g_n\right)}{\sum_{S \in RS} \sum_{g_n \in S} Count\left(g_n\right)} \tag{6}$$

As Eq. (6) shows, RS represents the artificial standard abstract, that is, the reference abstract, $Count_m(g_n)$ represents the maximum number of the same n-grams that appear between the generated abstract and the reference abstract, and $Count(g_n)$ represents the number of n-grams in standard summary.

$$ROUGE - L = F_{LCS} = \frac{\left(1+\beta^2\right) R_{LCS} P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}} \tag{7}$$

As Eq. (7) shows, $LCS(S, RS)$ represents the length of the longest common subsequence, $len(S)$ and $len(RS)$are the lengths of the summary, R_LCS, P_LCS are respectively Recall and Precision.

BLEU: Proposed by researchers at IBM in 2002 [20], BLEU can calculate the similarity between the summaries computed by the model and the reference summaries.

BERTScore: Tianyi Zhang et al. [11] proposed BERTScore, an evaluation metric for calculating the similarity. Compared with ROUGE and BLEU, this indicator is closer to human understanding of similarity. BERTScore uses contextual embeddings to compute token similarity.

4.3 Baseline Models

The experiments in this paper select 7 models as baseline models:

LexPageRank: LexPageRank system defines sentence centrality based on graph-based prestige [15]. It Applies the PageRank algorithm to the textual sentence relational representation and the extractive summarization.

MEAD: MEAD system is a extractive summarization baseline system. Text summaries are selected by scoring the importance of sentences by considering their centroids, positions, common sequences, and keywords [16].

SuBmodular: SuBmodular system treat the text summarization problem as maximizing a submodular function under a budget constraint [16].

UniAttention: UniAttention system is the basic sequence-to-sequence model. The model takes into account the attention mechanism in the process of text summarization [18].

NLPONE: NLPONE proposes to add an new attention mechanism on output sequence and uses the subword method. And it gets a significant improvement [12].

PGN: PGN: Proposed in ACL2017, which used pointer networks and attention-based Seq2Seq models to get improvements [13].

TKF: TFK system is a multi-attention sequence-to-sequence model that pays attention to topic keyword information [14].

4.4 Results

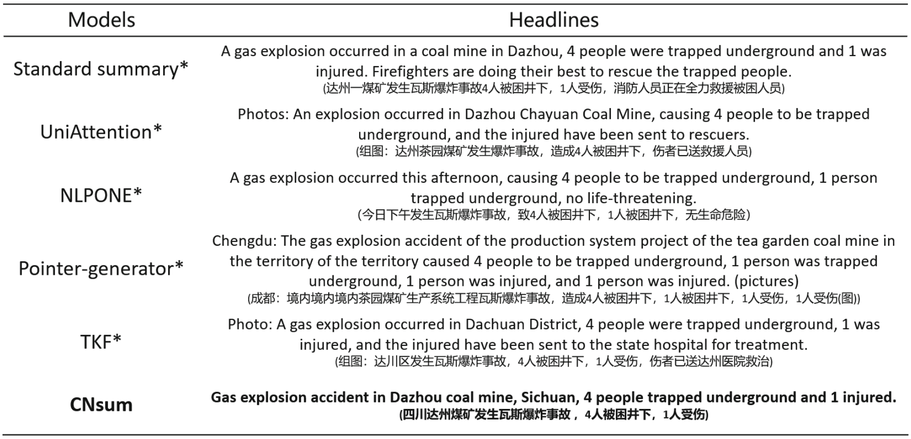
We use the Adam optimizer [22] and set the learning rate to be 1e-8, with a batch size of 8 and 10 epochs. Hugging Face's Transformers library [23] was used in all our experiments. In this paper, based on comparing with the running tests of the baseline models, the results are shown in Table 1. In order to verify the reliability of the model, we tested it on several different Chinese corpus, and the results are shown in Table 2. Figure 2 shows the summaries automatically generated by different models based on the same news article. It can be concluded that CNsum pays more attention to the generalization of the news content and the complete expression of the key information of the source text. It can be concluded that the model proposed in this paper has a good effect in Chinese news headline generation task.

Table 1. Results comparison of overall summarization. The index data results of the models marked with * come from the reference to the literature [21]

Models	Rouge-1	Rouge-2	Rouge-L
LexPageRank*	0.23634	0.10884	0.17578
MEAD*	0.28674	0.14872	0.22365
SuBmodular*	0.29704	0.15283	0.21668
UniAttention*	0.33752	0.20067	0.29462
NLPONE*	0.34983	0.21181	0.30686
PGN*	0.36022	0.21978	0.29888
TKF*	0.37667	0.24077	0.32886
CNsum	**0.38021**	**0.24083**	**0.34764**

Table 2. Evaluations of CNsum on different datasets.

Metrics	nlpcc2017	SogouCS	LCSTS	THUCNews
ROUGE-1	0.38021	0.33659	0.36493	0.29897
ROUGE-2	0.24083	0.21353	0.17257	0.15616
ROUGE-L	0.34764	0.24964	0.33244	0.17600
BLEU	0.45976	0.28173	0.38521	0.22807
BERTScore	**0.50149**	**0.51382**	**0.53468**	**0.50031**



Models	Headlines
Standard summary*	A gas explosion occurred in a coal mine in Dazhou, 4 people were trapped underground and 1 was injured. Firefighters are doing their best to rescue the trapped people. (达州一煤矿发生瓦斯爆炸事故4人被困井下，1人受伤，消防人员正在全力救援被困人员)
UniAttention*	Photos: An explosion occurred in Dazhou Chayuan Coal Mine, causing 4 people to be trapped underground, and the injured have been sent to rescuers. (组图：达州茶园煤矿发生爆炸事故，造成4人被困井下，伤者已送救援人员)
NLPONE*	A gas explosion occurred this afternoon, causing 4 people to be trapped underground, 1 person trapped underground, no life-threatening. (今日下午发生瓦斯爆炸事故，致4人被困井下，1人被困井下，无生命危险)
Pointer-generator*	Chengdu: The gas explosion accident of the production system project of the tea garden coal mine in the territory of the territory caused 4 people to be trapped underground, 1 person was trapped underground, 1 person was injured, and 1 person was injured. (pictures) (成都：境内境内境内茶园煤矿生产系统工程瓦斯爆炸事故，造成4人被困井下，1人被困井下，1人受伤，1人受伤(图))
TKF*	Photo: A gas explosion occurred in Dachuan District, 4 people were trapped underground, 1 was injured, and the injured have been sent to the state hospital for treatment. (组图：达川区发生瓦斯爆炸事故，4人被困井下，1人受伤，伤者已送达州医院救治)
CNsum	**Gas explosion accident in Dazhou coal mine, Sichuan, 4 people trapped underground and 1 injured.** **(四川达州煤矿发生瓦斯爆炸事故，4人被困井下，1人受伤)**

Fig. 2. Comparison of summaries generated by the models on the same news article. The results of the Chinese news headlines generated by the model marked * refer to the literature [21]

5 Conclusion

In this work, we proposed CNsum, a Seq2Seq model for abstractive text summarization on Chinese News texts. Experiments show that in the task of generating Chinese news headlines, CNsum generates a summary that is closer to the standard summary and contains key information of the source text. According to the indicators ROUGE, BLEU and BERTScore of model effect evaluation, the model has achieved good results. The results show that the model's BERTScore scores are always greater than 0.5, which verifies the model's outperformance on headlines generations. After training on the news corpus, the model has achieved better performance than the baseline models in the task of generating news headlines, which has certain application values. Tests on similar Chinese news corpus shows that the model has good generalization ability. The experiments' results also have certain reference value for applying the pre-trained language model trained in English to other languages. Compared with the benchmark model, it is improved by several percentage points. We will continue to pay attention to the development of Chinese text summarization techniques to further improve the accuracy and objectivity of Chinese text summarization generation.

References

1. Allahyari, M., et al.: Text summarization techniques: a brief survey. Int. J. Adv. Comput. Sci. Appl. (IJACSA) **8**, 397–405 (2017)
2. Iboi, H., Chua, S., Ranaivo-Malançon, B., Kulathuramaiyer, N.: Performance of opinion summarization towards extractive summarization. J. Telecommun. Electron. Comput. Eng. **9**, 57–64 (2017)
3. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. **2**(2), 157–165 (1958)
4. Syed, A., Gaol, F., Matsuo, T.: A survey of the state-of-the-art models in neural abstractive text summarization. IEEE Access **9**, 13248–13265 (2021)
5. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS 2014, pp. 3104–3112. MIT Press, Cambridge (2014)
6. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 1724–1734. Association for Computational Linguistics, October 2014. https://aclanthology.org/D14-1179
7. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates Inc. (2017)
8. Li, J., Tang, T., Zhao, W.X., Wen, J.-R.: Pretrained language model for text generation: a survey. In: Zhou, Z.-H. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, International Joint Conferences on Artificial Intelligence Organization, pp. 4492–4499, Survey Track (2021)
9. Lin, C.-Y., Hovy, E.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pp. 150–157 (2003). https://aclanthology.org/N03-1020

10. Hu, B., Chen, Q., Zhu, F.: LCSTS: a large scale Chinese short text summarization dataset (2015)
11. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: evaluating text generation with BERT, arXiv, vol. abs/1904.09675 (2020)
12. Hua, L., Wan, X., Li, L.: Overview of the NLPCC 2017 shared task: single document summarization. In: Huang, X., Jiang, J., Zhao, D., Feng, Y., Hong, Yu. (eds.) NLPCC 2017. LNCS (LNAI), vol. 10619, pp. 942–947. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73618-1_84
13. See, A., Liu, P.J., Manning, C.D.: Get to the point: summarization with pointer-generator networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, Canada, pp. 1073–1083. Association for Computational Linguistics, July 2017. https://aclanthology.org/P17-1099
14. Hou, L.-W., Hu, P., Cao, W.-L.: Automatic Chinese abstractive summarization with topical keywords fusion. IEEE/CAA J. Automatica Sinica **45**, 530–539 (2019)
15. Güneş, E., Radev, D.: Lexpagerank: prestige in multi-document text summarization. In: The 20th International Joint Conference on Artificial Intelligence, pp. 365–371 (2014)
16. Radev, D.R., et al.: MEAD - a platform for multidocument multilingual text summarization. In: LREC (2004)
17. Lin, H., Bilmes, J.: Multi-document summarization via budgeted maximization of submodular functions. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, California, pp. 912–920. Association for Computational Linguistics, June 2010
18. Chopra, S., Auli, M., Rush, A.M.: Abstractive sentence summarization with attentive recurrent neural networks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, pp. 93–98. Association for Computational Linguistics, June 2016
19. Lin, C.-Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, Barcelona, Spain, pp. 74–81. Association for Computational Linguistics, July 2004
20. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pp. 311–318. Association for Computational Linguistics, July 2002
21. Zhu, Q., Li, L., Bai, L., Hu, F.: Chinese text summarization based on fine-tuned GPT2. In: Mohiddin, M.K., Chen, S., EL-Zoghdy, S.F. (eds.) Third International Conference on Electronics and Communication; Network and Computer Technology (ECNCT 2021), SPIE 2022, vol. 12167, pp. 304–309. International Society for Optics and Photonics (2022). https://doi-org-s.libyc.nudt.edu.cn:443/10.1117/12.2629132
22. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. Computer Science (2014)
23. Wolf, T., et al.: Huggingface's transformers: state-of-the-art natural language processing, arXiv e-prints (2019)

Higher Layers, Better Results: Application Layer Feature Engineering in Encrypted Traffic Classification

Zihan Chen[1,2,3], Guang Cheng[1,2,3](✉), Zijun Wei[1,2,3], Ziheng Xu[1,2,3],
Nan Fu[1,2,3], and Yuyang Zhou[1,2,3]

[1] School of Cyber Science and Engineering, Southeast University,
Nanjing 211189, China
zhchen@njnet.edu.cn,
{chengguang,220215369,220205229,220205076,yyzhou}@seu.edu.cn
[2] International Governance Research Base of Cyberspace (Southeast University),
Nanjing 211189, China
[3] Jiangsu Provincial Engineering Research Center of Security for Ubiquitous
Network, Nanjing 211189, China

Abstract. Encrypted traffic has become the primary carrier of network transmission, and encrypted traffic classification is essential for advanced network management and security protection. Existing studies mainly focus on encrypted traffic feature engineering and classification model design, aiming to select more expressive features from encrypted traffic and achieve high-performance classification. The most commonly used features in the feature engineering process are statistical features and sequence features obtained in network or transport layers, which are more inclined to represent the factors of network transmission rather than the data attributes of applications or services. As a result, the relevance of the features and application or services is not strong, leading to unsatisfactory performance. To solve this problem, we introduce the Application Data Unit (ADU) and put forward the application layer feature engineering, which uses the features of the highest protocol level - the application layer to achieve better HTTPS classification. In order to compare the classification effects of features of different layers, we carried out experiments on traditional machine learning models based on statistical features and deep learning models based on sequence features, respectively. The results show that the proposed ADU features are better than the segment granularity features of the TLS layer and far better than the packet granularity features both in statistical and length sequence features. The average F1-score increase in the encrypted traffic application classification scenario is more than 10%.

Keywords: Encrypted traffic classification · Application layer feature engineering · Application data unit · HTTPS

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 548–556, 2022.
https://doi.org/10.1007/978-3-031-19214-2_46

1 Introduction

As the main form of Internet traffic, full encryption of Web traffic has become an inevitable trend. According to the Google Transparency report "HTTPS Encryption on the Web" [4], 99% of Chrome loading web pages are encrypted, and the protocol most used in web traffic is the HTTPS. HTTPS combines HTTP application protocol and TLS encryption protocol, using TCP as the transport layer protocol to form the TCP-TLS-HTTP protocol stack.

Traffic management and security protection used to rely on the Deep Packet Inspection (DPI) [10] approach, but encryption has rendered this approach obsolete. As an alternative, machine learning methods were proposed, using statistical or sequence features [1], which are not hidden by encryption.

The existing research on encrypted traffic classification can be divided into two categories. Most research tries to improve the classification efficiency from the perspective of classifiers (e.g., introducing new models [7,8], combining or transferring from the prior models [5,6]). The other part of the study focuses on the in-depth mining of more expressive complex features from the perspective of feature engineering [2].

However, the features considered are basically at the network layer or the transport layer, and only a very few studies [3] investigate higher-level protocol features for classification. Different layers and protocols have their own unique Protocol Data Unit (PDU). It can be seen from the protocol stack pattern that the PDU sequence of the higher layer protocol is more similar to the data. As the highest layer in the current widely used TCP/IP protocol stack, the application layer is the protocol layer closest to the data.

Therefore, this paper proposes the concept of application layer feature engineering. First, we introduce the concept of Protocol Unit (PU), PDU, and Application Data Unit (ADU) under HTTPS encrypted traffic. Then, we propose different traffic feature engineering schemas in HTTPS, especially the application layer feature engineering. In addition, we use a variety of existing models to conduct sufficient and meticulous experiments of encrypted traffic classification on the statistical features and length sequence features of three different PDUs.

Our main contributions are as follows:

- We propose the concept of application layer feature engineering. It breaks the limitation that existing encrypted traffic classification methods mainly utilize statistical or sequence features of packet or flow. According to the current usage of PDU length sequence, PDU is extended to statistical features, and the ADU is proposed as a new unit to construct statistical features and length sequence features for encrypted traffic classification.
- Experiment results show that in machine learning and ensemble learning methods consistent with the statistical features, the performance of ADU features is better than TLS segment features and TCP packet features, both in full flow and the fixed data size. In the state-of-the-art deep learning methods consistent with length sequence features, the performance of ADU features is still satisfactory and better than other PDUs. It demonstrates the advantage of application layer feature engineering.

2 Related Works

The mainstream research on encrypted traffic classification includes the model selection and feature engineering optimization, and these two research fields are also complementary and promote each other.

2.1 Encrypted Traffic Classification Models

Traditional machine learning methods were first applied in encrypted traffic classification. The most typical ones are the decision tree C4.5 and the SVM [1]. However, these methods lack experimental verification in an open world environment and have preconditions of feature engineering.

Later, several ensemble learning models were proposed. The most representative one is random forest [7]. Ensemble learning achieves better classification results, but it requires considerable computing costs and is prone to over-fitting.

With the generation of neural networks, deep learning is applied to encrypted traffic classification. The specialty of automatic feature selection can learn directly from the original input without prior expert knowledge of features. 1D-CNN was first used [8]. Then, the LSTM model, as the most widely used variant of RNN, is presented to depict the sequence relations between packets [11].

With the development of semi-supervised learning and hardware computing capability, some studies superimpose various models to improve the classification effect, such as Deep Packet [6], and SAM [9].

2.2 Encrypted Traffic Feature Engineering

Statistical features are considered first for the classification of encrypted traffic. However, they can not reflect the local bursts. Moreover, deep learning methods specializing in automatic feature selection are convenient, but the features selected are not necessarily superior, and the procedure is not interpretable.

Therefore, some researchers are dedicated to select more suitable features by feature engineering to depict the local sequential Markov properties effectively. The most common one is the length sequence, which is less affected by the network environment and encryption. The FS-Net [5] model is based on the GRU model and representation learning, which takes a full-flow length sequence as input.

Due to the layered design of network protocols, data needs to be segmented during actual transmission. As a result, the packet length obtained at the network layer or transport layer differs significantly from the actual application layer length. The PDU [2] was proposed to reduce the influence caused by protocol segmentation to solve this problem. In this field, the most representative one is LSCDL architecture [3], which takes PDU length sequence as input with N-gram hyper length sequence as the feature.

The current encrypted traffic feature engineering ignores the impact of network protocol engineering on encrypted traffic data segmentation. Therefore, this paper studies the application layer feature engineering of encrypted traffic to improve the expression ability of the selected features.

3 Preliminaries

3.1 Protocol Unit and Protocol Data Unit

A PU is the smallest unit each protocol owns in each layer of the network protocol stack. It contains the protocol header and the corresponding data. A PDU is the smallest unique unit of transmission data the protocol owns. Protocol objectives lead to differences in PUs and PDUs between protocols even at the same level. Generally speaking, PU is the whole of the atomized protocol header and data body, while PDU is the atomized data body. It is important to note that not every PU contains a data body (such as TCP ACK packets), but all PUs have a protocol header.

In the network transmission, the size of PDU is limited. A larger PDU will increase the number of data that need to be scheduled at a single time, while a smaller PDU will increase the number of times that need to be scheduled with the same amount of data. Therefore, an appropriate size should be set for the PDU to meet the current protocol and network requirements.

3.2 Application Data Unit

ADU is the PDU of the application layer. As this paper focuses on classifying encrypted traffic over HTTPS, we will consider only ADUs of HTTP. We give the following definition: the ADU of HTTP refers to all data transmitted by the HTTP protocol body in a request and response process. Since the request and response are paired in a successful request, the ADUs are also paired, even though the request may not carry a data body.

If the classified unit is unidirectional, the ADUs in the two directions will be separated during feature extraction. If the classification element is bidirectional flow, it is considered whether to add direction identifiers for some features according to the selected feature space.

3.3 Encrypted Traffic Classification

In the network transmission, a flow refers to a packet set consisting of quintuples of IP addresses and ports of both parties with the transport layer protocols in a single transmission. It can be expressed as:

$$F =< src_{IP}, dst_{IP}, src_{port}, dst_{port}, t_{proto} > \tag{1}$$

Encrypted traffic classification is the process that classifies encrypted traffic to specific services or applications based on certain features when classification atoms (usually flows) and category labels are determined. Assuming there are N samples to be classified and C different categories, then the i-th sample (assuming the feature space size is m) $x_i = [f_1^{(i)}, f_2^{(i)}, \cdots, f_m^{(i)}]$, where $f_j^{(i)}$ refers to the j-th feature of the input. If the real category of x_i is S_i, the goal of the encrypted traffic classification is to build a model $\phi(x_i)$ to get a predicted label $\hat{S}_i$ which is expected to be the real label S_i.

4 Traffic Feature Engineering in Different Layers of HTTPS

In the HTTPS scenario, feature engineering can be divided explicitly into feature engineering at the TCP, TLS, and HTTP layers.

4.1 TCP Layer Feature Engineering

Feature engineering of the TCP layer refers to feature selection and extraction with TCP PDU as a feature unit. It is worth noting that although the TCP layer has its own maximum limit of TCP load length called MSS, the TCP is responsible for timeout and retransmission because there is no timeout retransmission mechanism at the IP layer. Therefore, in the actual network environment, the MSS negotiated by TCP handshake ensures that the length of IP packets does not exceed the MTU of the data link layer. This phenomenon makes the data fragment of TCP PDU and IP PDU the same as that of the packet.

4.2 TLS Layer Feature Engineering

The feature engineering of the TLS layer is TLS-oriented PDU, i.e., the segment of TLS. HTTPS traffic information obtained through passive measurement is expressed in packets in the actual network. To obtain the segment of TLS, we need to concatenate TCP packets. The splicing is produced by concatenating TCP packets with the same ACK number in the same direction, which is also how the TLS segment is acquired in the actual network.

TLS is a special protocol with two distinct phases, the handshake, and the record, implemented by nesting two layers of different headers. As TLS headers are also variable-length headers, PU and PDU of TLS need to consider the information in the handshake and record phases when constructing statistical features. In TLS layer feature engineering, length sequence is still data-oriented, so TLS header information is also not considered. Only the length sequence of the segments in the record phase is considered, that is, the data length of TLS Application Data PU.

4.3 Application Layer Feature Engineering

The feature engineering of the application layer is different from that of the other two layers. First, HTTP requests and responses are strictly corresponding, and a successful request can only get one response. In most cases, the request does not carry actual data; that is, the length of the ADU of the HTTP request is 0. In HTTPS, the application layer is encrypted, and the HTTP header and the HTTP ADU are bound together and encrypted. The application layer header and ADU cannot be directly distinguished without other methods.

However, this situation will not affect the application layer feature engineering. In the training stage of encrypted traffic classification, we can decrypt the obtained samples to obtain their actual header and ADU lengths to realize feature calculation with ADU as the unit.

5 Evaluation

5.1 Dataset

We collected and labeled the traffic samples in the large-scale network environment of CERNET. The dataset covers ten kinds of currently the most popular HTTPS applications on the global Chinese Internet. The dataset is called CERNET-1.1-10 and is available at https://data.iptas.edu.cn/web/tbps. It is worth mentioning that all the flows we collected are complete TCP flows defined by Eq. (1), including handshake packets and end packets. The dataset is divided into balanced training and testing sets by a strict 8:2 ratio.

5.2 Experiments

In statistical features, traditional machine learning and ensemble learning methods are mainly used, including kNN, C4.5, SVM, AdaBoost, and RF. In length sequence features, the latest deep learning methods are mainly compared, including FS-Net [5], LS-CapsNet [2], and LS-LSTM [3].

Classification Experiments in Statistical Features. The experiments are divided into a full flow and a fixed data volume scenario. The full flow scenario represents the offline classification that can totally express the features. The fixed data volume scenario represents the near-real-time online classification, in which the total size of a flow cannot be sure.

The learning curves of five different models under three PDUs with the full flow scenario are shown in Fig. 1. It can be seen that RF stands out among the five models under any PDU. Moreover, the overall effect of ADU is better than the other two PDUs.

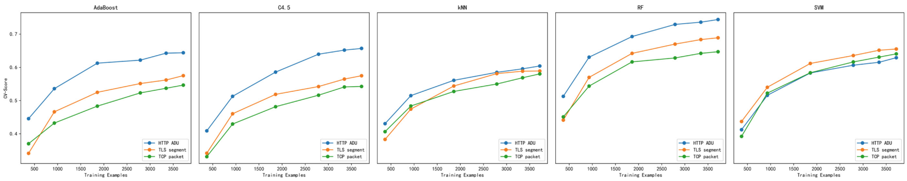


Fig. 1. Learning curves of five models in full flow scenario

Since it is not possible to directly determine the appropriate amount of data in a fixed data volume scenario, RF was used to conduct experiments under different total numbers of input packets, and the results are shown in Table 1. Because both segment and ADU are spliced, the number of these two PDUs changes and is much smaller than the number of packets. The results show that 300 packets are a suitable choice.

Table 1. Classification accuracy for fixed data sizes of RF

Fixed data sizes (Input packet counts)	100	200	300	400	500
HTTP ADU	0.7250	0.7458	**0.7539**	0.7378	0.7465
TLS segments	0.7033	0.7226	**0.7236**	0.7129	0.7079
TCP packets	0.6367	0.6168	0.6352	**0.6584**	0.6530

With fixed 300 packets as input, we further conducted ten experiments for each model and the F1-score box plots are shown in Fig. 2. The results show that ADU statistical features significantly improve the classification performance by more than 10% in the three valid models with fixed data volume. It is consistent with our experiment in the full flow scenario.

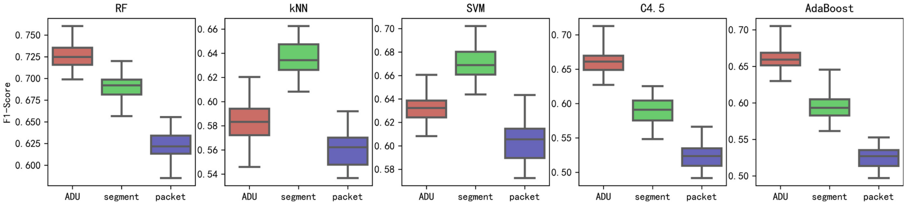


Fig. 2. Box-plot of five models in fixed 300 packets scenario with F1-score

Classification Experiments in Length Sequence Features. To further prove the advantages of application layer feature engineering, we conducted control experiments on three state-of-the-art deep learning methods on the length sequence features. However, given the input differentiae of the selected model, it is not suitable to conduct the experiment under full flow or fixed data.

The convergence curves of three different models under three PDUs using length sequence features are shown in Fig. 3. It can be seen that the performance of ADU length sequence features under the three models is better than that of the other two PDUs, and the convergence speed under the LS-LSTM and FS-Net models is faster than that of other PDUs.

The classification result of three PDUs in three methods using length sequence features is shown in Table 2. The classification effect of ADU length sequence features is far superior to segment and packet in both precision and recall rate.

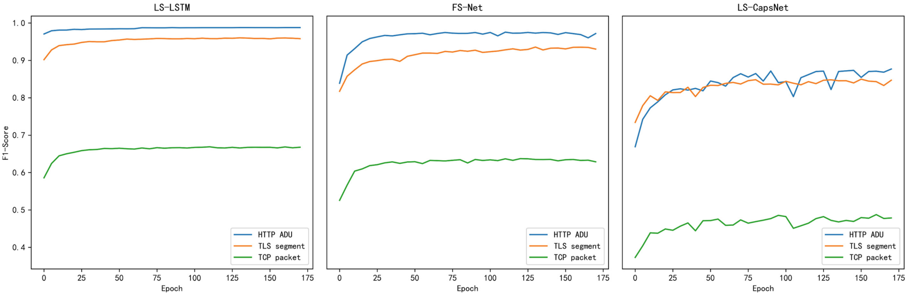


Fig. 3. Convergence curves of three PDUs in three methods using length sequence

Table 2. Classification precision and recall of three PDUs in three methods using length sequence

Methods	Packet			Segment			ADU		
	Pr	Rc	F1-score	Pr	Rc	F1-score	Pr	Rc	F1-score
LS-LSTM	0.6912	0.6232	0.6554	0.9372	0.9362	0.9367	**0.9761**	**0.9728**	**0.9744**
FS-Net	0.6649	0.6014	0.6316	0.8999	0.8964	0.8982	**0.9504**	**0.9477**	**0.9490**
LS-CapsNet	0.6664	0.4938	0.5673	0.8512	0.8505	0.8508	**0.8780**	**0.8771**	**0.8775**

6 Conclusions

In order to further improve the effect of encrypted traffic classification, we propose the concept of application layer feature engineering, which breaks the limitation of the features from the network or transport layer. We introduce the concepts of PU and PDU, combined with the particularity of the HTTPS protocol stack. The application layer feature engineering is instantiated as the feature engineering of HTTP ADU. We prove the effectiveness of application layer feature engineering in HTTPS encrypted traffic classification through detailed experiments. Experimental results show that ADU features outperform TLS segment and TCP packet in both statistical features and length sequence features.

However, HTTPS application layer feature engineering still faces two problems. The first is the error of the ADU length value caused by TLS encryption. The ADU length obtained directly contains the HTTP header length. Since HTTP headers are variable-length protocol headers, the ADU length deviation caused by the HTTP header needs to be reduced. Otherwise, the input will have interference. The second problem is caused by multiplexing and nested transport in the HTTP-2.0 scenario, where a PDU may contain multiple ADUs in different request-response pairs. Therefore, future research should focus on accurately restoring ADU to achieve encrypted traffic classification based on the ADU length sequence.

Acknowledgement. This paper is supported by the General Program of the National Natural Science Foundation of China under Grant No. 62172093.

References

1. Bagui, S., Fang, X., Kalaimannan, E., Bagui, S.C., Sheehan, J.: Comparison of machine-learning algorithms for classification of VPN network traffic flow using time-related features. J. Cyber Secur. Technol. **1**(2), 108–126 (2017)
2. Chen, Z., Cheng, G., Jiang, B., Tang, S., Guo, S., Zhou, Y.: Length matters: Fast internet encrypted traffic service classification based on multi-PDU lengths. In: 2020 16th International Conference on Mobility, Sensing and Networking (MSN), pp. 531–538 (2020)
3. Chen, Z., Cheng, G., Xu, Z., Guo, S., Zhou, Y., Zhao, Y.: Length matters: Scalable fast encrypted internet traffic service classification based on multiple protocol data unit length sequence with composite deep learning. Digit. Commun. Netw. **8**, 289–302 (2021)
4. Google: HTTPS encryption on the web - Google Transparency Report (2022). https://transparencyreport.google.com/https/overview
5. Liu, C., He, L., Xiong, G., Cao, Z., Li, Z.: FS-Net: a flow sequence network for encrypted traffic classification. In: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, pp. 1171–1179 (2019)
6. Lotfollahi, M., Siavoshani, M.J., Zade, R.S.H., Saberian, M.: Deep packet: A novel approach for encrypted traffic classification using deep learning. Soft. Comput. **24**(3), 1999–2012 (2020)
7. Shi, Y., Ross, A., Biswas, S.: Source identification of encrypted video traffic in the presence of heterogeneous network traffic. Comput. Commun. **129**(Sep.), 101–110 (2018)
8. Wang, W., Zhu, M., Wang, J., Zeng, X., Yang, Z.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 43–48. IEEE (2017)
9. Xie, G., Li, Q., Jiang, Y.: Self-attentive deep learning method for online traffic classification and its interpretability. Comput. Netw. **196**, 108267 (2021)
10. Yang, B., Liu, D.: Research on network traffic identification based on machine learning and deep packet inspection. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 1887–1891 (2019)
11. Yao, H., Liu, C., Zhang, P., Wu, S., Jiang, C., Yu, S.: Identification of encrypted traffic through attention mechanism based long short term memory. IEEE Trans. Big Data **8**, 241–252 (2019)

P-LFA: A Novel LFA-Based Percolation Fast Rerouting Mechanism

Minghao Xu, Tao Feng(✉), Xianming Gao, Shanqing Jiang, Shengyuan Qi, and Zhongyuan Yang

National Key Laboratory of Science and Technology on Information System Security, Institute of System Engineering, PLA Academy of Military Science, Beijing, China
xumingh_16@163.com

Abstract. Loop Free Alternate (LFA for short) is designed to avoid the interruption of transmission paths during routing convergence when a network suffers from single point failures (e.g. node failure, or link failure). Unfortunately, once more than one associated failure occurs simultaneously, the LFA protocol cannot strictly guarantee lossless packets. Therefore, we propose a LFA-based percolation routing mechanism (P-LFA for short). It establishes the shortest paths and their backup paths by using distributed routing protocol and LFA, and uses a percolation rerouting algorithm to calculate the percolation paths. Once the shortest path and corresponding backup path are interrupted, the network node immediately selects the percolation paths to continue transmitting packets without dropping any packets. And this mechanism allows each node to cache packets. This way can efficiently avoid packet loss when the path to the destination node is unreachable. The experimental results show that P-LFA can effectively guarantee zero-loss packets when some related failures occur. Meanwhile, even though there do not exist reachable paths to the destination node, the P-LFA mechanism can cache unreachable packets into the network node that is the closest to the destination node, and continue to re-forward packets when paths recover.

Keywords: Route convergence · Zero-loss · Percolation idea · Packet caching

1 Introduction

With the rapid development of networking technology, Internet has already become a vital information infrastructure carrying people's economic, social, communication, and other activities. People have increased expectations for network transmission performance, particularly for dependable transmission. However, several factors will lead to transmission failure, such as software failure, hardware failure and data failure. Since these failures are unpredictable and unavoidable, the network typically adopts distributed routing protocols to improve its robustness. Once the link is not reachable, the network restarts the process of routing convergence, by sending the latest link information to neighbors. And then other nodes recalculate the forwarding tables based on the received link information. Unfortunately, packets continue to transmit along the interrupted transmission path until the network calculates new paths, and these packets will be discarded.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 557–571, 2022.
https://doi.org/10.1007/978-3-031-19214-2_47

For a business with high real-time and time-sensitive services, the massive loss of packets caused by a single failure is intolerable. Therefore, how to construct a network of lossless transmission is a hot spot, which has attracted more attention from industry and academia.

In the early stage, researchers focused on shortening the network convergence time, mainly including two methods: fault detection efficiency and fault recovery efficiency: 1) fault detection efficiency. Bidirectional Forwarding Detection (BFD for short) is designed, which focuses on the fault detection of the forwarding plane. It can achieve the rapid fault detection in 30 ms and has been widely used [1]. 2) fault recovery efficiency. Researchers have improved the speed of transmission path recalculation by reducing the diffusion time of routing information and routing calculation time [2]. Although these methods effectively reduce network convergence time, they are still not ideal for high real-time and time-sensitive services. Researchers propose a better network fast recovery mechanism to reduce packet loss, which is called FRR [3]. FRR adopts the pre-calculation way to calculate the backup paths, which is used to avoid packets as far as possible when the network is affected by failure [4]. When the network occurs a single failure, the node can immediately use the pre-calculated backup path to transmit packets without waiting for network convergence. However, it is just suitable for a single failure. Once the network occurs multipoint failure, the backup path may be also interrupted, which results in the packet loss.

In this paper, we propose a LFA-based percolation fast rerouting mechanism, which is called Percolation Loop Free Alternates (P-LFA). It has three paths: 1) the shortest paths calculated using the SPF algorithm, 2) the backup paths calculated by TI-LFA, and 3) the percolation paths calculated using the percolation rerouting algorithm. When both the shortest paths and the backup paths are interrupted, the node selects the percolation paths to transmit packets. P-LFA also uses caching mechanism to prevent packet loss, and each node caches packet. Thus P-LFA ensures that packets are cached at the closest node to the destination when there is no path to the destination node, and the packets are sent firstly to the destination node when the failure is recovered. The effectiveness of the algorithm is verified by a simulation experiment. The experiment shows that the P-LFA mechanism inherits the advantages of TI-LFA and can achieve 100% coverage. It is concluded that the P-LFA mechanism can effectively improve the reachability of packets through qualitative and quantitative analysis.

The paper is structured as follows. Section 2 describes related work about FRR. Section 3 introduces the P-LFA framework including routing calculation, forwarding table, and forwarding processing. Then, in Sect. 4 we explain the percolation rerouting algorithm. Afterwards, we describe the packet forwarding process in Sect. 5. And we verified the effectiveness and feasibility of P-LFA through experiments in Sect. 6. Finally, Sect. 7 presents the conclusions and directions for future research work.

2 Related Work

In recent years, IP FRR has been deployed in almost all production networks. To adapt different networks, researchers put forward a variety of FRR mechanisms [5–8]. Based on the differences in implementation mechanisms, FRR mechanisms can be roughly divided into two methods: Unicast-based IP FRR and multicast-based IP FRR.

2.1 Unicast-Based IP FRR

Unicast-based IP FRR assumes link or node failure in advance and calculates the backup path at the source node. When the network node senses the failure, it switches to the backup path and continues to send packets. As a typical unicast-based IP FRR, LFA is a simple and feasible mechanism and is widely used in production networks [9]. It only needs to calculate the backup paths of the link or node according to the network topology at the source node according to the no loop criterion, without too much modification to the routing protocol of the network. When the shortest path is not reachable, the network node can achieve fast rerouting by using the pre-calculated backup path, and its convergence time is genera Chiesa M, Sedar R, Antichi G, et al. Fast ReRoute on Programmable Switches. IEEE/ACM Transactions on Networking, 2021, 29(2): 637–650.lly less than 50 ms.

However, LFA does not apply to all topologies. When the network node or link does not meet any loop criterion, that is, there is no potential next hop in a node or link, its fast rerouting mechanism invalidates. At the same time, the calculated backup path by the LFA mechanism is usually not the path after convergence. These drawbacks affect the scope of application of LFA. To expand the applicability of LFA, researchers have proposed a variety of improvement methods. Csikor L et al. propose the Remote Loop Free Alternate (RLFA) [10]. The P space and Q space are proposed to find the LFA path, and the LFA coverage is extended by the way of tunneling, but it introduces additional sessions and increases the complexity of the operation. Singh JA et al. proposed the Topology Independent Loop Free Alternate (TI-LFA) mechanism [11]. Based on the LFA and RLFA mechanisms, this mechanism uses Segment Routing (SR) to ensure that no additional sessions are introduced during the transmission process. It not only maintains the simplicity of operation but also achieves 100% coverage irrespective of network topology, and the calculated backup paths by the TI-LFA mechanism are the convergence paths.

2.2 Multicast-Based IP FRR

In order to improve the robustness and reachability of the network when faced with multiple failures at the same time, the multicast-based IP FRR is proposed. Multicast includes PIM, BIER, etc. Jozef Papan et al. proposed the Bit Repair Fast Reroute (B-REP) IP FRR Mechanism [12]. This mechanism uses the BIER header field (bit string) to use backup paths. The mechanism implements hop-by-hop routing using bit-string matching while allowing managers to define backup paths manually. It ignores link metrics when calculating paths, allowing any possible physical alternative to be selected, thus achieving 100% network coverage. However, B-REP only provides protection against a single fault. In order to provide advanced protection against multiple faults in a network domain, Jozef Papan et al. also proposed the new Enhanced Bit Repair (EB-REP) IP FRR Mechanism [13]. The mechanism improves on the shortcomings of B-REP, and provides protection for each port to support protection against multiple outages in a single network. By supporting multiple-network failure protection, the EB-REP mechanism is more robust and flexible in situations of unexpected network error conditions.

Papan J et al. proposed the Multicast Repair (M-REP) IP FRR Mechanism [14], which uses the independent multicast-dense mode (PIM-DM) multicast routing protocol to randomly generate backup paths. The biggest difference between this mechanism and others is that there is no need to pre-calculate alternative routes under the assumption of various network failures. When the path is not reachable, the network node encapsulates the protected packets into specific multicast packets, floods the packets in a specific multicast group, and forwards the packets to the destination node. As long as the network has a reachable path, the packets will be transmitted to the destination node. And this mechanism can be used in any network topology. In the same year, they expanded in dealing with multi-point failures on M-REP, and proposed the Enhanced Multicast Repair (EM-REP) [15]. It enhances the handling of multi-point failures and optimizes the multicast behavior of multi-area networks such as the IS-IS protocol.

We identify some problem areas that arise from the characteristics of IP FRR mechanisms through look at the issue of IP FRR mechanisms. Table 1 compares the indicators of the IP FRR mechanism mentioned above. Unicast-based IP FRR solves the problem of insufficient coverage, but it is only applicable to the scenario of a network single point of failure. Multicast-based IP FRR effectively improves the network robustness in case of multi-point failure, but it is difficult to realize considering the problems of complex multicast network configuration management, limited multicast address space, and low actual deployment rate of a multicast protocol.

Table 1. Comparison table of key indicators of IP FRR

	EB-REP	EM-REP	LFA	RLFA	TI-LFA
Coverage	100%	100%	70%–80%	99%	100%
Fault points	>1	>1	1	1	1
Packet status (no path)	Loss	Loss	Loss	Loss	Loss
Pre-calculation	Yes	No	Yes	Yes	Yes
Change packet	Yes	Yes	No	Yes	Yes
Backup path optimality	No	No	No	No	Yes
Tunnel	No	Yes	No	Yes	No

3 P-LFA Architecture

LFA just solves a single point of failure, such as link failure or node failure. Once the pre-calculated backup path also fails, the network still suffers from packet loss. In order to achieve lossless transmission, we propose P-LFA that provides percolation paths to ensure that packets are still transmitted to the destination node when the network has multiple points of failure. Besides, packets can be cached in the closest node to the destination even if there is no reachable way. So P-LFA can achieve a packet loss rate of 0%.

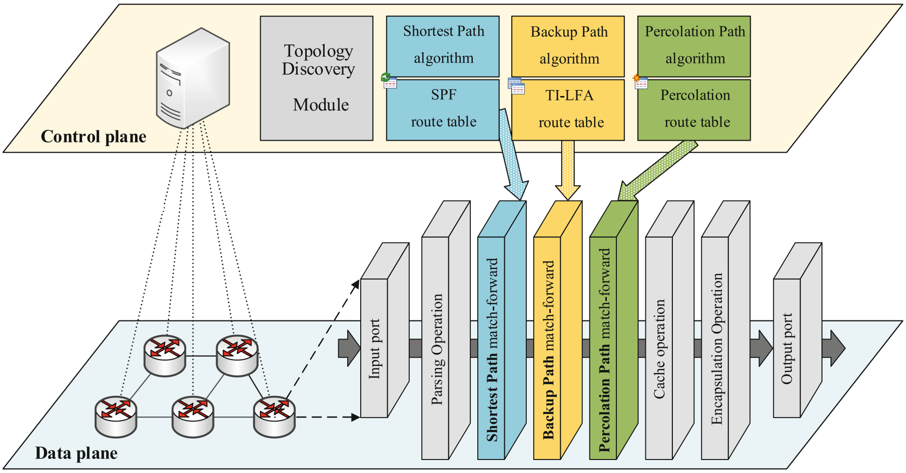


Fig. 1. P-LFA architecture

The P-LFA architecture consists of the control plane and data plane: the control plane generates routing tables based on topology information; the data plane processes packets based on the information sent from the control plane. The P-LFA architecture is shown in Fig. 1. The control plane is responsible for running routing protocols, which include three core modules: Topology Discovery Module, Path Calculation Module, and Routing Generation Module.

- Topology Discovery Module: It obtains the global topology G (V, E) by exchanging various network information among network nodes. In G (V, E), V denotes network nodes, and E denotes network links.
- Path Calculation Module: It calculates the paths by path generation algorithms. Path generation algorithms include three algorithms: shortest path generation algorithm, backup path generation algorithm, and percolation path generation algorithm.
- Routing Generation Module: It generates routing tables based on the results of the path generation module. Three algorithms generate three different routing tables.

The data plane is responsible for forwarding various types of packets, which include four core operations: Parsing Operation, Matching Operation, Cache Operation, and Encapsulation Operation.

- Parsing Operation: It extracts the header of the packet for subsequent matching of tables.
- Matching Operation: It is used to match forwarding entry to find out the next hop. Once it is successfully matched, it will stop the matching operation.
- Cache Operation: When the next-hop occurs failure, it caches the packet prevent packet loss.
- Encapsulation Operation: It encapsulates the header before sending the packet to the output port.

3.1 Path Calculation

P-LFA consists of three paths: shortest path, backup path, and percolation path. The three paths are computed by using different algorithms.

- Shortest Path: P-LFA uses a combination of Incremental Shortest Path First (ISPF for short) algorithm and partial route calculation algorithm to calculate the shortest paths. It algorithm uses Dijkstra to calculate the shortest paths. Each node generates a Shortest Path Tree (SPT for short) with itself as the root node and other routers as the leaf nodes. In contract to SPF, when the link states change, ISPF only calculates all the nodes in the first calculation, and the affected nodes need to be recalculated when the link states change. And the partial route calculation algorithm updates the route based on the SPT calculated by ISPF to complete the recalculation of the shortest path.
- Backup Path: P-LFA uses the TI-LFA mechanism to calculate the backup paths. TI-LFA defines P-space and Q-space: P-space is the space of the set of nodes whose shortest path from PLR to these nodes does not pass through the failure point; Q-space is the space of the set of nodes whose shortest path from the destination to these nodes does not pass through the failure point. The algorithm can provide backup paths for any topology.
- Percolation Path: we propose a percolation rerouting algorithm to calculate percolation paths in P-LFA. This algorithm is distributed, with each node calculating its percolation paths. The idea of percolation is similar to the idea of "water flowing downhill", where packets are expected to keep moving closer to the destination in case of multiple failures. The algorithm defines the node level (NL for short) and divides the network topology graph hierarchically. Each node is classified into different NL according to the distance from the destination. The closer the node to the destination, the higher the NL is. The algorithm also invokes the idea of backtrack, which allows packets to be sent from a high-level node to a low-level node. The node determines whether a packet has been backtracked by Δt. The node looks for conforming nodes to establish percolation paths. A neighboring node is a conforming node when its NL is greater than or equal to this node. When there are less than three eligible nodes, the node will take a backtrack operation. The packet will be sent to nodes with lower NL. When the packet reaches the other node, it will be matched and forwarded normally.

3.2 Forwarding Table

The node's forwarding tables provide the next-hop to determine the transmission of packets, which also includes three forwarding tables in the P-LFA.

Shortest Path Forwarding Table. The shortest path forwarding table is the default table. It has the highest priority. The packets choose the shortest path as forwarding at the source. This forwarding table indicates which is the best next hop for the packet. Its structure is shown in Table 2.

Table 2. Shortest path forwarding table

Destination address	Metric	Next hop	Output interface
8::	10	6::	Ens38

The shortest path forwarding table contains the destination address, metric, next hop, and output interface. When packets are transmitted using the shortest path forwarding table, the successful match is determined according to the output interface state. When the output interface state is up, the packet is encapsulated and sent to the output interface. If the interface state is down, the match is considered fail. The packet will change the header according to the pre-calculated Segment List to achieve the backup path switching.

Backup Path Forwarding Table. The backup path forwarding table is used for matching SRH headers. Once the local address of the node is the same as the destination address of the received packet, this node is called Endpoint. When Endpoint receives the packet, it will match the backup path forwarding table and modify the packet header. The table specifies that the packet ends the current label and performs the next label. The structure is shown in Table 3.

Table 3. Backup path forwarding table

Destination address	FuncType	Flavor	Next hop	Output interface
4::	End. X	--	5::	Ens38

The backup path forwarding table contains the destination address, functype, flavor, next hop, and output interface. The operation is divided into two steps when the packet is matched with the backup path forwarding table. First, the node subtracts one from Segments Left (SL) in SRH for the next label instruction. Secondly, the node transforms the destination address into the destination of the next Segment List. And the packet is forwarded according to the functype in the backup path forwarding table, such as End, End. X, etc. If the output interface is down, the packet will be matched with the percolation path forwarding table.

Percolation Path Forwarding Table. Except for node ID、network address, each node has its node level that is used for the Percolation Path. The percolation path forwarding table is used to transmit packets when both the shortest path and the backup path fail. The percolation path forwarding table provides multiple next hops to the destination for the packet. Once the packet matches the percolation path forwarding table in the node, this node modifies the next hop, NL, and Δt in the header. The structure is shown in Table 4.

Table 4. Percolation path forwarding table

NL	Destination address	Next hops	Δt
1	20::	R2, R3, R4	–
2	30::	R3, R5	–
2	30::	R1, R2	-1

The percolation path forwarding table contains the NL, destination address, next hop ID, and an operation of Δt. When the packet matches the percolation path forwarding table, the node first changes the packet header. Then the node selects the corresponding next hops according to the destination node and changes Δt according to the forwarding table instructions. If a node receives a packet carrying $\Delta t = 0$, it directly matches the percolation path forwarding table. The packet will be forwarded to the next hops without changing Δt, while the Δt of the header is changed to 1. When a node receives a packet carrying $\Delta t = 1$, it will no longer match the percolation path forwarding table.

3.3 Forwarding Process

When a packet is sent from the source to the destination, the packet is selected to use the shortest path for transmission at the source. It will be forwarded to the destination along the shortest path in normal network condition. When the network fails, i.e., the shortest path fails, the packet will use the backup path for transmission. The node types the Segment List into the packet header to guide the subsequent forwarding of the packet. When the node finds that the backup path has also failed, it will use the percolation paths for transmission. The node diffuses the packet through the percolation path forwarding table.

Once a node uses the percolation paths, it first checks whether the header contains Δt and NL. If the header does not contain Δt and NL, the node needs to type $\Delta t = 1$ and the NL of this node and the node ID into the packet header. Then the node diffuses the packet according to the percolation path forwarding table. When a node receives a packet containing Δt, it means that the packet is forwarded using the percolation path. The node will first check whether it has accepted the packet. If it has received the packet, it drops the packet. If it has not received the packet yet, look at the Δt in the header. If $\Delta t = 0$, it will directly match the percolation path forwarding table. If $\Delta t = 1$, it will follow the normal matching process. However, if the next hop ID matches the node ID in the header, it is considered that the matching fails, and the path switching operation is performed. When the shortest path and backup path of the node fail again, the packet no longer matches the percolation path forwarding table. The node will decide whether to cache according to its own NL and the NL carried in the header. The priority of the paths is: shortest path > backup path > percolation path.

4 Percolation Path Algorithm

The principle of the percolation rerouting algorithm is to calculate the percolation paths by the NL of the nodes. The algorithm is a distributed algorithm, which is calculated and maintained by each node. The algorithm constructs a network hierarchy topology to generate nodes' NL. Each node sends its NL to its neighboring nodes to generate a percolation path routing table. It generates percolation paths based on the NLs of neighboring nodes. The process for calculating the percolation paths is as follows.

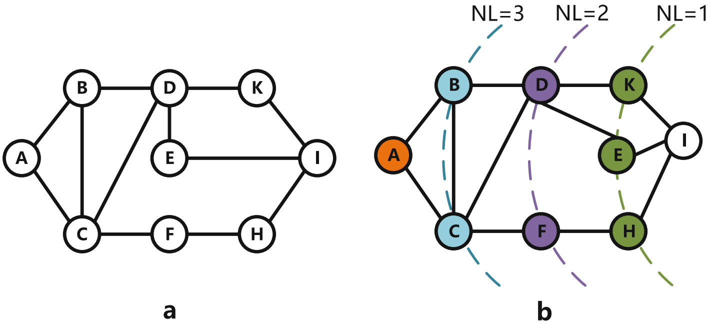


Fig. 2. Network hierarchy topology diagram

Step 1 Build Network Hierarchy Topology. First, we construct the network hierarchy topology. The network hierarchy is divided according to the distance of all nodes relative to the destination. According to the position of all nodes relative to the destination, the node with fewer hops represents the closer to the destination, and its NL is higher. Take the network topology shown in Fig. 2a as an example, its network hierarchy topology is shown in Fig. 2b. The relationships between nodes in Fig. 2b can be divided into three relationships: a) peer relationship, i.e., nodes have the same node level; b) superior relationship, such as K is a superior node of D; c) subordinate relationship, such as B is a subordinate node of D.

Algorithm 1 describes the process of calculating the network hierarchy. In constructing the network hierarchy, it is initiated by the destination. First, the destination sets its NL (D, D) to 0 and sends broadcast packets with P (D, D) = 1. When the node receives broadcast packets P (Ni, D) from its neighbor Ni, it will calculate its NL by Algorithm 1.

In Algorithm 1, G (N, E) represents the network topology. N represents the network node. E represents the link. S is the source. D is the destination.

Algorithm 1. Network node level calculation

```
input: $G(N,E),S,D,N_0$
output: $NL_0(N_0,D)$
$NL_0(N_0,D) \leftarrow 13$
if $N_0 = D$ then
    $NL_d(D,D) \leftarrow 0$
    send $P(D,D) = 1$
else if $N_0 = S$ then
    $NL_s(S,D) = 0$
else
    for receive $P(N_i,D)$ then
        if $P(N_i,D) < NL(N_0,D)$ then
            $NL(N_0,D) \leftarrow P(N_i,D)$
            for $N_j \leftarrow Adjacency(N_0)$ then
                if $N_j \neq K_i$ then
                    send $P(N_0,D) = NL_0(N_0,D) + 1$ to $N_k$;end
```

Step 2 Percolation Path Calculation. Each node sends its NL to neighboring nodes. The node generates a percolation routing table based on the NL of the neighboring nodes. The node finds the percolation paths based on the NL of its neighbors. The node finds nodes to construct percolation paths based on the percolation routing table. The node first finds a neighboring node with the NL greater than or equal to the node and sets it as a percolation path. When there are less than three eligible neighbor nodes, the node will take a backtrack operation. The node not only sends the packet to the eligible nodes but also sends the packet to neighboring nodes of lower NL. The algorithm sets the number of times a packet can be backtracked to one. Δt is used to determine whether the packet has been backtracked. When the node performs a backtracking operation, Δt is reduced by 1 to become 0.

In Algorithm 2. N_0 represents the failed-aware node. N_k represents the neighboring nodes of fault-aware nodes.

Algorithm 2. Percolation path generation algorithm

```
***input***: $NL_0(N_0, D), NL_k(N_k, D), \Delta t$
***output :*** *next hop nodes*
$C \leftarrow 0$
***for*** $N_j \leftarrow Adjacency(N_0)$ ***then***
    ***if*** $NL_j \geq NL_0$ ***then***
        $C = C + 1$
        $N_j$ ***is*** *next hop node*
    ***else***
        $Backtrack\{N_i\} \leftarrow N_j$
***if*** C<3 ***then***
    ***if*** $\Delta t = 0$ ***;end***
    ***else***
        ***for*** $N_m \leftarrow Backtrack\{N_i\}$ ***then***
            $\Delta t \leftarrow \Delta t - 1$
            $N_m$ ***is*** *next hop node*
```

5 Packet Caching

During link transmission, packets are dropped and retransmitted due to network failure problems. These phenomena greatly affect the quality of service. The unicast-based IP FRR mechanism generates packet loss when the pre-calculated backup path also fails; the multicast-based IP FRR mechanism can cope with multi-point network failures, but still takes packet loss when the network has no reachable path. To ensure that packets are not lost in network transmission, a caching mechanism is proposed for the rerouting mechanism. Packets are always present in the link during transmission by using the caching mechanism. When there is no reachable path in the network, the packets can be cached in the node closest to the destination. This mechanism achieves a packet loss rate of 0%. This mechanism reduces the long retransmission delay by sending the message to the destination first after the network failure is recovered.

After the packet is matched within the node, it is encapsulated and sent to the specified output port. To support the caching mechanism, the port is assigned the register to cache packets. When the port is not working properly, packets arriving at the port will be cached until the failure is recovered. When the port is restored to normal, messages are forwarded directly without the need for matching operations. The caching mechanism will not take effect during normal message transmission. The mechanism is triggered when the path fails. When a packet fails to match the shortest path forwarding table, the caching mechanism will copy the packet. The node will cache the copied packet in the register of the next-hop port. Similarly, when a packet fails to match the backup path, the caching mechanism also copies the packet and caches it in the corresponding output port register. The caching mechanism is different when using the percolation path. When the percolation path fails to match, the NL between the source and this node is compared to determine whether the packet needs to be cached. If the NL of this node is higher than that of the source node. This means that the node is closer to the destination than the sender, then the caching operation is performed. Otherwise, the packet is discarded. When the port resumes normal operation to send out cached packets, the existing cache is deleted.

6 Evaluation of P-LFA Mechanism

6.1 Experiment Environment

We develop a simulator to evaluate P-LFA, which runs on a physical device with a CPU of 3.20 GHz, a memory of 16G, and an operating system of Windows 10. The simulator can consider a comparison of the packet reachability when using different routing algorithms. Besides, it first calculates the paths based on the source and destination, and then randomly destroys links in the network. After links are destroyed, it can re-calculate paths and analyze the packet reachability.

Assume that the topology contains 56 nodes and 103 links, which are designed according to the real network topology. In our experiment, 100 sets of source and destination nodes are randomly selected in the network topology. In order to better highlight experimental results, each group of data was randomly executed 1000 times.

6.2 Path Robustness

When the network link occurs failure, P-LFA will permeate according to the node level of the neighbor node. After the neighbor node receives the packet, it will continue to transmit according to the routing table entry, without the need for pre-calculation.

The experimental results show that the path success rate changes significantly as the number of faulty links increases from one to six, as shown in Fig. 3. When any single link fails, TI-LFA and P-LFA can ensure that the packet is still 100% reachable, and SPF has lost packets. As the number of link failures increases, the success rate of P-LFA is significantly higher than that of TI-LFA. When six link failures occur in the network, the success rate of TI-LFA is reduced to 96%, while P-LFA can still be maintained above 99%. When a small number of link failures occur in the topology, P-LFA can improve the reachability of packets and achieve high reliability of packet transmission. Therefore, the path robustness of P-LFA is better than the existing SPF and TI-LFA.

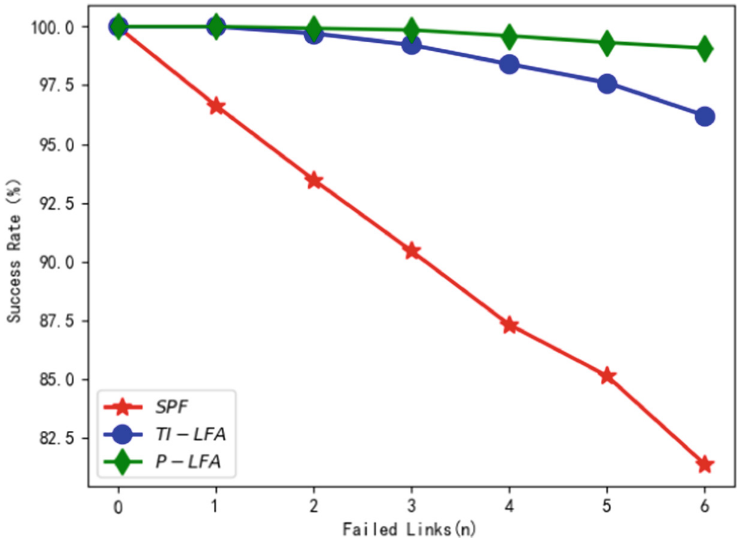


Fig. 3. Comparison of the success rates of packet transmission among the three mechanisms when a small number of links are faulty. The success rate of the P-LFA mechanism has been higher than 99%.

6.3 Loss Tolerance

Existing LFA mechanisms precompute the backup path based on assumed failure point. However, this mechanism only considers a single point of failure. When the network has multiple points of failure, its effectiveness may be greatly reduced.

P-LFA solves the weakness of LFA by introducing the percolation algorithm. The percolation algorithm is no longer limited to the faulty node, but also considers the neighboring nodes of the faulty node. When the network encounters one or more fault, the percolation algorithm takes the faulty node as the center of the circle and selectively spreads the packet to the neighboring node, which can provide more optional paths for packet transmission. Meanwhile, P-LFA achieves 100% topology-independent coverage by using the percolation algorithm between a faulty node and neighboring nodes. When the packet has no way to go, P-LFA can avoid packet loss by caching packets. When the fault is recovered, packets can be re-transmitted to the destination host for the first time. Thus, P-LFA effectively reduces the delay of retransmission and achieves zero-loss.

6.4 Computational Cost

To verify the computational cost of P-LFA, we simulate the time for generating three different paths. After randomly selecting a set of source and destination nodes, we record the time when the source node computes the path and generates the forwarding tables. The computational time represent the computational cost of three algorithms, as shown in Fig. 4. Due to the large difference in computation time of the three algorithms, we take e^{T} as the comparison value of T (T is used to indicate computational cost).

According to the experimental results, the computational time of the shortest path and the percolation path is nearly equal, which is about 0.1 ms. The computational time to calculate the backup path is about 294 ms, which is 3000 times as long as the computational time of percolation path. Therefore, the P-LFA mechanism adds a small computational cost to the nodes when calculating the percolation path, but the mechanism shows better protection than the TI-LFA mechanism when dealing with multi-point failures in the network.

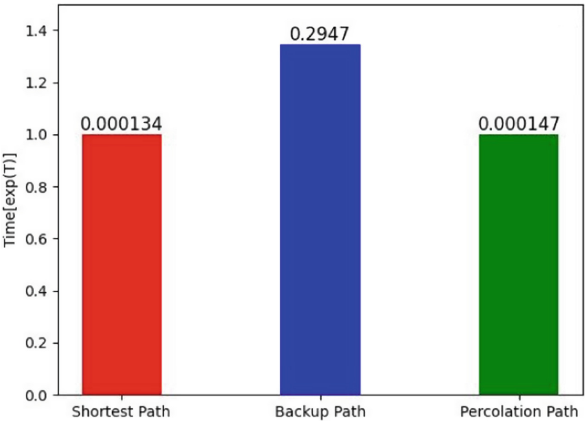


Fig. 4. Computational time of three paths. The computational cost of the algorithm is judged by Computational time. The computational cost of three algorithms is: shortest path < percolation path < backup path

7 Conclusions

We propose P-LFA, which utilizes the idea of percolation to provide a more robust fast re-routing mechanism for IP network. In P-LFA, the percolation algorithm is firstly proposed to find a reachable path by bypassing link faults without pre-computing paths. In addition, P-LFA uses cache to ensure that packets are not lost when the paths fail, avoiding the overhead of packet retransmission. These two mechanisms ensure that P-LFA can provide better network robustness and enhance link reliability without adding excessive computational complexity. The next works mainly analyze how to further learn from "gravitational potential energy" and to introduce it into the percolation algorithm. It may let packet forwarding no longer just based on given links.

References

1. Abaid, A., et al.: Convergence time analysis of border gateway protocol using GNS. In: 2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA, pp. 689–694 (2021)

2. Geng, H., Shi, X., Wang, Z., et al.: A hop-by-hop dynamic distributed multipath routing mechanism for link state network. Comput. Commun. **116**, 225–239 (2018)
3. Papan, J., et al.: Review of fast ReRoute solutions. In: 2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA), pp. 498–504 (2020)
4. Shand, M., Bryant, S.: IP fast reroute framework (2010)
5. Papan, J., et al.: Overview of IP fast reroute solutions. In: 2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA), pp. 417–424 (2018)
6. Qiu, K., Zhao, J., Wang, X., et al.: Efficient recovery path computation for fast reroute in large-scale software-defined networks. IEEE J. Sel. Areas Commun. **37**(8), 1755–1768 (2019)
7. Chiesa, M., Sedar, R., Antichi, G., et al.: Fast ReRoute on programmable switches. IEEE/ACM Trans. Netw. **29**(2), 637–650 (2021)
8. Bryant, S., Previdi, S., Shand, M.: A framework for IP and MPLS fast reroute using not-via addresses. Internet Requests for Comments. RFC Editor. 6981 (2013)
9. Papan, J., et al.: Existing mechanisms of IP fast reroute. In: 2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA), pp. 1–7 (2017)
10. Csikor, L., Rétvári, G.: On providing fast protection with remote loop-free alternates. Telecommun. Syst. **60**(4), 485–502 (2015). https://doi.org/10.1007/s11235-015-0006-9
11. Singh, J., Sachin, K., Shushrutha, K.: Implementation of topology independent loop free alternate with segment routing traffic. In: 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–5 (2021)
12. Papan, J., et al.: A new bit repair fast reroute mechanism for smart sensors IoT network infrastructure. Sensors. **20**(18), 5230 (2020)
13. Papan, J., Segec, P., Kvet, M.: Enhanced bit repair IP fast reroute mechanism for rapid network recovery. Appl. Sci. Basel **11**(7), 3133 (2021)
14. Papan, J., et al.: The new multicast repair (M-REP) IP fast reroute mechanism. Concurr. Comput.-Pract. Exp. **32**(13), 15 (2020)
15. Papan, J., et al.: Enhanced multicast repair fast reroute mechanism for smart sensors IoT and network infrastructure. Sensors **20**(12), 3428 (2020)

PU_Bpub: High-Dimensional Data Release Mechanism Based on Spectral Clustering with Local Differential Privacy

Aixin Lin and Xuebin Ma(✉)

Wireless Networking and Mobile Computing Laboratory,
Inner Mongolia University, Hohhot 010000, China
csmaxuebin@imu.edu.cn

Abstract. Although the release and analysis of high-dimensional data bring tremendous value to people, it causes great hidden danger to participants' privacy in the meantime. Various privacy protection methods based on differential privacy have been proposed at present. However, most of them cannot simultaneously solve the problems of high computational overhead and privacy threats from untrusted servers caused by the curse of high dimensionality. Therefore, we propose a safer and more effective high-dimensional data release algorithm based on local differential privacy, which is referred to as PU_Bpub. It effectively preserves the dimensional correlation of the original high-dimensional data and reduces the communication overhead of synthetic data. Extensive experiments on real-world datasets demonstrate that our solution substantially outperforms the state-of-the-art techniques in terms of computational overhead, and the synthetic dataset has high utility.

Keywords: High-dimensional data · Data synthesis and release · Local differential privacy · Spectral clustering

1 Introduction

Today, with the continuous progress of science and technology, high-dimensional datasets released have benefited people greatly. Nonetheless, unprecedented privacy threats to participants have emerged due to complex correlations among multiple attributes and the vulnerabilities of untrusted servers.

Currently, the privacy protection of high-dimensional data concerns mainly covers two-fold challenges: the curse of high dimensionality and non-local privacy protection [1]. Xiao et al. in [3] and Zhang et al. in [4] built the correlation models of high-dimensional attributes via dependency graph or threshold filtering to reduce dimensionality. Nevertheless, to high dimensionality, they do not account for the complex dependency between high-dimensional data, significantly reducing data availability. LDP (local differential privacy) can effectively

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 572–581, 2022.
https://doi.org/10.1007/978-3-031-19214-2_48

solve non-local privacy protection. However, the LDP also has some shortcomings. If the LDP wants to improve the data available in the training process, it needs much data, which will bring considerable communication overhead.

Therefore, releasing high-dimensional data with LDP still faces utility and computational efficiency challenges. We propose a new mechanism to address these issues. Our significant contributions are listed as follows.

- First, we propose PU_Bpub, a high-dimensional data release mechanism based on Spectral Clustering with LDP. This mechanism uses the dependency graph and weighted Bayesian network to ensure the complex dependency relationship between data. At the same time, it reduces the synthesis time of high-dimensional datasets by spectral clustering of the dependency graph.
- Second, we design a new weight selection method to establish a Bayesian network to better balance privacy and data utility.
- Third, we implement and evaluate our schemes on different real-world datasets. Experimental results show that our method dramatically reduces the running time and has high utility for synthetic data under LDP.

The remainder of the paper is organized as follows. Section 2 provides a literature review. In Sect. 3, we outline the preliminaries. Section 4 introduces our scheme. Section 5 presents the performance evaluation of our plans. Finally, we conclude the paper in Sect. 6.

2 Related Work

Currently, the release of high-dimensional datasets based on DP can be divided into four aspects: feature dimension reduction [6,7], random projection [8], and probability graph model [9–11]. A full survey of methods to realize differential privacy is beyond the scope of this work. Here, we identify the most related efforts and discuss why they cannot fully solve the problems above.

Feature Dimensionality Reduction: Chaudhuri et al. [6] proposed to combine PCA with CDP to solve the problem of privacy disclosure in high-dimensional data release. It can reasonably display the data correlation, but it increased the time complexity.

Random Projection: Sun et al. [8] proposed Multi-RPHM, which was a high-dimensional numerical data collection algorithm that met the LDP. However, the inherent structure of data is not considered in the generation of the transformation matrix, which reduces the data utility.

Probability Graphs mainly use Bayesian networks and joint trees: Zhang et al. [9] based CDP proposed PrivBayes. However, with the increasing number of attributes and the assumption of a trustworthy server, PrivBayes faces that the accuracy decreases significantly. To this end, Ren et al. in [10] proposed Lopub and in [11] proposed LoCop and DR_LoCop, which used joint trees to describe the data correlation and used LDP to protect the privacy of high-dimensional data. However, due to the estimation of multi-dimensional joint distribution, the communication overhead of the above methods increases significantly with the increase of data dimension.

3 Preliminaries

3.1 Local Differential Privacy

Generally speaking, a mechanism under differential privacy assumes that the data aggregation server is trustworthy. Nonetheless, the server may be dishonest and vulnerable to some inside attacks. To this end, local differential privacy (short for the LDP) was proposed to ensure the privacy of each individual's data on the user side. A formal definition of local differential privacy is given below.

Definition 1. *((ε)-LDP)) [12]: Given N users, each user has a record and is given a privacy algorithm M, if algorithm M obtains the same output result s^* on any two records s and $s^{'}$, the following relationship exists (1), algorithm M is said to meet ε-LDP.*

$$Pr(M(s) = s^*) \leq e^{\varepsilon} \times Pr(M(s) = s^{'}) \tag{1}$$

4 System Model

4.1 Basic Idea

We propose a high-dimensional data release algorithm based on LDP, which is referred to as PU_Bpub. LDP is used to disturb high-dimensional data at the client to prevent the privacy disclosure of sensitive data. The complex dependencies between high-dimensional data are processed through a dependency graph and weighted Bayesian network so that the synthetic dataset has high utility. Spectral clustering is used to reduce the problem of significant communication overhead when synthesizing new datasets.

4.2 Transformation with LDP

In this component, the local client performs data conversion based on RAPPOR [1]. First, encode the data, then disturb the encoded data, aggregate the disturbing string of each attribute to form a large string S' ,and then send it to the server.

4.3 Multi-dimensional Distribution Estimation

In this component, the multi-dimensional probability distribution can be efficiently estimated through the Lasso regression algorithm from the aggregated Bloom filter strings. It mainly includes the following steps.

- Forming candidate matrix. The Bloom filter is used as the characteristic variable. A cartesian production connects the binary vectors of different candidate values to create a matrix M', then transforms into a candidate matrix M.
- Counting and estimation. After the server receives the disturbing string S', for each bit $S_j^i[b]$ in each attribute A_j, the central server counts the number of "1". The sum formation vector $\widetilde{x}$ of 1 of all bits.

- Linear Regression. Given a vector $\widetilde{x}$ and a candidate matrix M, $\widetilde{x} = M\beta$ can be used to estimate the probability distribution.
- Lasso Regression. Lasso regression reduces the dimension of sparse data, and then the probability distribution is estimated at $\beta = Lasso(\widetilde{x}, M)$.

Algorithm 1. Weighted Bayesian Network

```
Input: Dataset $D$, Maximal Degree of Bayesian Network $k$, Attribute Set $A$
Output: Weighted Bayesian Network $N$
Initialization: $N = \emptyset$, adjacency matrix $G_{d\times d} = 0$, $\phi_{m,n} = \theta$
for each $i \in [1, d]$ do
    Compute Mutual Information of attributes $A_m$ and $A_n$:
    $I_{m,n} = \sum_{i\in\Omega_m} \sum_{j\in\Omega_n} p_{ij} \frac{p_{ij}}{p_{i.}p_{.j}}$
    if $I_{m,n} > \phi_{m,n}$ then
        $G_{m,n} = G_{n,m} = 1$, Get Adjacency Matrix $G_{d\times d}$
    end if
end for
Degree Matrix : $W_{d\times d} \leftarrow G_{d\times d}$
Laplacian Matrix : $L_{d\times d} = W_{d\times d} - G_{d\times d}$
Cut the Attribute Graph to get the Attribute Subgraph
Compute the weight value of Attribute Subgraph, select the one with the largest weight value as the initial node
return $N$
```

4.4 Computing Dependence Structure

In this component, the server calculates the mutual information according to the multi-dimensional joint probability distribution, obtains the adjacency matrix, establishes the dependency graph according to the adjacency matrix, then performs spectral clustering on the dependency graph to form the attribute subgraph. A dependency graph can represent complex dependencies between high-dimensional data attributes. However, the improper establishment of a dependency graph will reduce the utility of synthetic data and significantly reduce the effect in data analysis. Algorithm 1 illustrates the construction process of the weighted Bayesian network. In this paper, we use mutual information to compute correlations between high-dimensional data. The threshold $\phi_{m,n}$ is set and compared with mutual information $I_{m,n}$ to establish a good dependency graph. The threshold value $\phi_{m,n}$ is set as follows:

$$\phi_{m,n} = min(|\Omega_n| - 1, |\Omega_m| - 1) \cdot \frac{\gamma^2}{2} \tag{2}$$

where Ω_m , Ω_n are the domain sizes of attributes A_m, A_n, γ is the dependency degree that determines the correlation level of the attributes. In statistics, it is generally recognized that there have correlations between attributes when $0.3 \leq \gamma \leq 0.7$. Thus we set dependency threshold $\gamma = 0.3$.

4.5 Initial Node Selection

In this component, the attribute subgraph with practical significance should be chosen as far as possible to improve data utility. The specific operational steps are as follows.

- Firstly, we select the initial node according to the static weight. The degree matrix $W_{d\times d}$ has been calculated in the third step of the above algorithm. Therefore, the sum of the degree matrix in the subgraph can be used as the static weight value to select the initial node. If the sum of the degree matrix of the subgraph is larger, which is proved that the dependence between the subgraph nodes is strong and has more practical significance. Thus, the static weight value can be solved according to the following formula:

$$\text{weight } = \sum_{j=1}^{n} w_{ij} \tag{3}$$

  where w_{ij} is the diagonal value of the degree matrix.
- Secondly, we select the initial node according to the Information entropy. The static weight value may be the same when using the weight value of each subgraph to select the initial node. When the static weight value of the subgraph is the same, the weight value of each subgraph is calculated by information entropy. It can be seen from the formula that the greater the information entropy, the stronger the correlation between subgraphs. Therefore, the subgraph with the largest information entropy is selected first. The calculation formula of information entropy is as follows:

$$H(U) = -p\left(U_i\right)\log p\left(U_i\right) \tag{4}$$

  where U_i is the *ith* possible value of the subgraph, $p(U_i)$ is the edge probability of the subgraph. From Eqs. (3) and (4), it can be obtained that the time complexity of using the static weight value is $O(n)$. Using information entropy as the weight value, the time complexity is $O(n\log n))$, so the combination of the two can reduce the time complexity.

4.6 Synthesizing New Dataset

In this component, a new dataset is synthesized according to the conditional probability distribution of each attribute subgraph. Algorithm 2 illustrates the sampling and synthesizing process. Firstly, we predefined set $\mathrm{Y} = \emptyset$, which is used to store the selected attribute subgraph. According to the method of initial node selection, an attribute subgraph is chosen from the weighted Bayesian network, and the probability distribution $P(A_C)$ of the attribute subgraph is obtained by estimating the multidimensional joint distribution according to Lasso regression. We sample and synthesize the dataset until the attribute subgraph is empty. Repeat the above step until all attribute subgraphs are sampled.

Algorithm 2. Synthesizing New Dataset

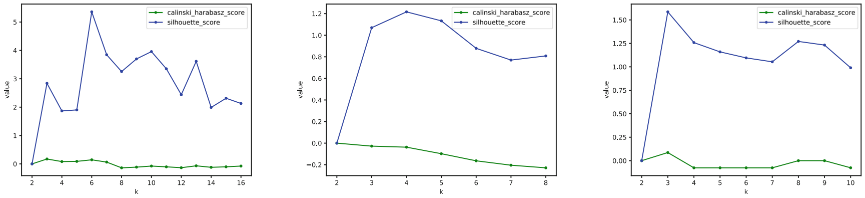
```
Input: Ç: a collection of attribute index clusters $C_1, \ldots C_k$, $A_j$ : $k$-dimensional attributes $(1 \leq j \leq k)$
Output: Synthetic Dataset $\widehat{U}$
Initialization: $\mathrm{Y} = \emptyset$
Repeat:
choose an attribute index cluster $C_j \in$ Ç:
estimate joint distribution $\mathrm{P}(A_C)$ by JD
sample $\widehat{U}_C$ according to $\mathrm{P}(A_C)$ until Attribute node sampling complete
Ç $=$ Ç $- C, Y = Y \cup C$
Select the attribute subgraph connected to the sampled node for sampling
Until Ç $= \emptyset$
Return: $\widehat{U}$
```

5 Performance Evaluation

This section demonstrates our scheme's performance and reports the experimental results. In the following, we first present the datasets and experimental setup. Then, we offer experimental results.

5.1 Datasets and Setup

We use three real-world datasets, including Adult, Retail and TPC-E [10,11]. In order to facilitate the comparative experiment, the dataset processing is consistent with the algorithms in [10,11]. All algorithms and experiments were implemented using Python 2.7, running on a Windows 10 PC with Intel Corelі5-8250 CPU 1.8 GHz and 8 GB RAM.



(a) TPC-E

(b) Adult

(c) Retail

Fig. 1. Number of clusters

5.2 Experimental Results

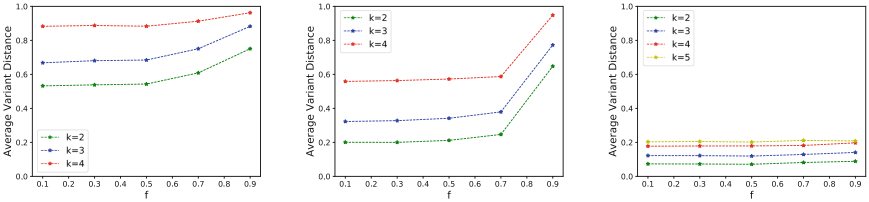
Number of Clusters. Because the change of the Calinski-Harabasz index in clustering is sometimes not significantly different, they are combined to select the best number of clusters. The larger they are, the better the clustering effect is. Thus, Fig. 1 describes TPC-E, Adult, and Retail optimal cluster sizes are 6, 4, and 3, respectively.

Accuracy. We use the Average Variation Distance (AVD) to measure the difference between the joint probability distribution of the synthetic datasets and the original datasets. The larger the AVD, the greater the error between the estimated joint distribution $P(w)$ and the original joint distribution $Q(w)$. The calculation formula of AVD is as follows:

$$AVD(P,Q) = \frac{1}{2} \sum_{w \in \Omega} |P(w) - Q(w)| \tag{5}$$

where Ω is the size of the dimensional attribute domain.

We test AVD using different k-way marginal (Binary dataset Retail with $k = 2, 3, 4, 5$, non-binary datasets Adult and TPC-E with $k = 2, 3, 4$). Figure 2 shows the AVD of TPC-E and Adult gradually increases with the privacy parameter f. The change of AVD of the Retail dataset is not apparent, which maintains within 1%. This is due to the estimated joint distribution of binary datasets being more accurate than non-binary datasets. The increase in f means that the better the privacy effect, the greater the disturbance of the data, which will bring more significant error in the joint distribution.

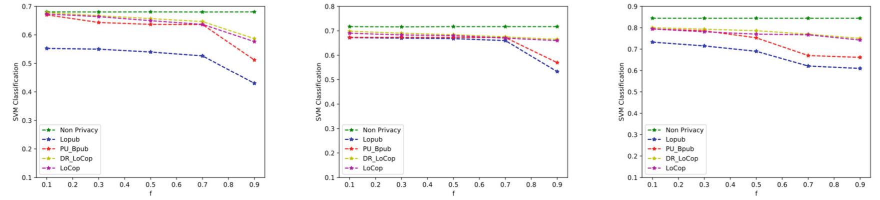


(a) TPC-E **(b)** Adult **(c)** Retail

Fig. 2. K-Way marginals query accuracy

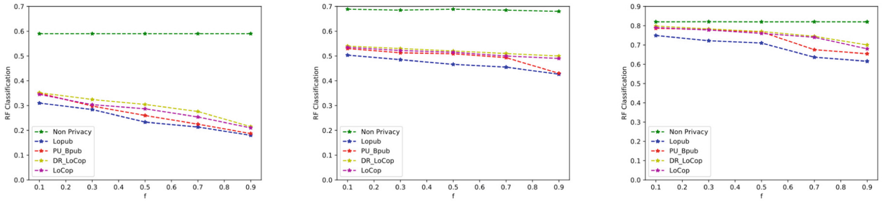
SVM and RF Classifications. The average SVM and RF classification accuracy of three datasets are shown in Fig. 3 and Fig. 4, which shows that the SVM/RF classification accuracy decreases with the flip probability f for all algorithms. This reflects the impact of privacy protection on the data utility. Our scheme consistently outperforms Lopub for classification algorithms. And it has little difference in the accuracy of SVM/RF classification of DR_LoCop and LoCop algorithm synthetic datasets. This is because we establish a dependency graph based on mutual information to maximize the retention of the attribute relationship between high-dimensional data after disturbance. As shown in Fig. 4, the RF classification rate of the Retail dataset is significantly higher than that of the TPC-E and Adult datasets when f is large, which is due to the large deviation in the joint distribution estimation of non-binary attributes. Overall, PU_Bpub retains good data utility to a certain extent.

Running Time. Figure 5 shows that PU_Bpub has significantly less running time for synthetic datasets than previous algorithms. This is because we use spectral clustering to cut the dependency graph and then build a weighted Bayesian network, which reduces time complexity. The time required for binary dataset Retail is significantly less than for non-binary datasets TPC-E and Adult. This is because the calculation of a binary dataset is simpler than that of a non-binary dataset for joint distribution estimation. To summarize, our scheme is highly efficient when processing high-dimensional data. While reducing the running time, the new synthetic dataset retains good data utility to a certain extent.



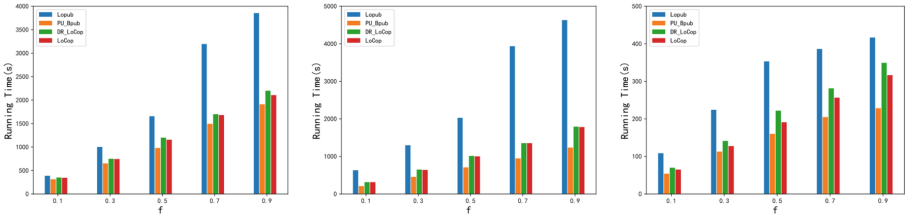
(a) TPC-E **(b)** Adult **(c)** Retail

Fig. 3. SVM classification



(a) TPC-E **(b)** Adult **(c)** Retail

Fig. 4. RF classification



(a) TPC-E **(b)** Adult **(c)** Retail

Fig. 5. Running time

6 Conclusion

In this paper, we propose a novel solution named PU_Bpub to achieve high-dimensional data release with local differential privacy. In order to effectively preserve the dimensional correlation of the original high-dimensional data, the server identifies the dimensional correlation based on the dependency graph after receiving the data protected by the user's local privacy. Spectral clustering divides the high-dimensional data attribute set into several relatively independent low-dimensional attribute sets to reduce the communication overhead of the synthetic data. A weighted Bayesian network is established to synthesize a new dataset by setting a reasonable weight to select the initial node to improve the utility of the synthetic dataset. The results show that the mechanism significantly reduces the communication overhead of synthesizing new datasets and maintains an excellent private utility trade-off.

Acknowledgement. This paper is supported by Inner Mongolia Natural Science Foundation (Grant No. 2018MS06026) and the Science and Technology Program of Inner Mongolia Autonomous Region (Grant No. 2019GG116).

References

1. Ye, Q.Q., Meng, X.F., Zhu, M.J., et al.: Survey on local differential privacy. J. Soft. **29**, 159–183 (2018)
2. Dwork, C.: Differential privacy in new settings. In: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 174–183. SIAM (2010)
3. Wang, N., Xiao, X.K., Yang, Y., et al.: PrivTrie: effective frequent term discovery under local differential privacy. In: Proceedings of IEEE ICDE, Piscataway, pp. 821–832. IEEE (2018)
4. Zhang, X., Chen, L., Jin, K., et al.: Private high-dimensional data publication with junction tree. J. Comput. Res. Dev. **55**(12), 2794–2809 (2018)
5. Hardt, M., Roth, A.: Beyond worst-case analysis in private singular vector computation. In: Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing, New York, USA, pp. 331–340. ACM (2013)
6. Chaudhuri, K., Sarwate, A.D., Sinha, K.: A nearoptimal algorithm for differentially-private principal components. J. Mach. Learn. Res. **14**(1), 2905–2943 (2013)
7. Peng, C., Zhao, Y., Fan, M.: A differential private data publishing algorithm via principal component analysis based on maximum information coefficient. Netinfo Secur. **20**(2), 37–48 (2020)
8. Sun, H., Yang, J., Cheng, X., et al.: A high-dimensional numeric data collection algorithm for local difference privacy based on random projection. Big Data Res. **6**(01), 3–11 (2020)
9. Zhang, J., Cormode, G., Procopiuc, C.M., et al.: PrivBayes: private data release via Bayesian networks. In: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, New York, USA, pp. 1423–1434 (2014)
10. Ren, X.B., Yu, C.-M., Yu, W.R., et al.: LoPub: high-dimensional crowdsourced data publication with local differential privacy. IEEE Trans. Inf. Forensic Secur. **13**, 2151–2166 (2018)

11. Wang, T., Yang, X., Ren, X., Yu, W., Yang, S.: Local private high-dimensional crowdsourced data release based on copula functions. IEEE Trans. Serv. Comput. **15**(2), 778–792 (2019)
12. Min, X., Bolin, D., Tianhao, W., et al.: Collecting and analyzing data jointly from multiple services under local differential privacy. Proc. VLDB Endowmen. **13**(11), 2760–2772 (2020)

R-TDBF: An Environmental Adaptive Method for RFID Redundant Data Filtering

Ziwen Cao[1,2], Degang Sun[2], Siye Wang[1,2,3](✉), Yanfang Zhang[1], Yue Feng[1,2],
and Shang Jiang[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{caoziwen,sundegang,wangsiye,zhangyanfang,fengyue,jiangshang}@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Sciences, Huairou, China

[3] School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China

Abstract. Radio Frequency Identification (RFID) technology plays an essential role in surveillance scenarios. However, redundant data hinders the efficient processing of data. The processing of RFID redundant data is of great importance to reduce the load of the RFID system and quickly detect the monitored tags. To address the issue, the research community introduced Bloom filtering technology into the RFID system. However, existing methods often use fixed thresholds and cannot adapt to complex environmental conditions. This work presents R-TDBF, a practical solution that enables data redundancy filtering in complex environments by rationally setting filtering thresholds. In addition, a signal strength threshold is also introduced in R-TDBF, which reduces the error caused by signal fluctuation. The experimental results show that the R-TDBF algorithm can filter redundant data well under different threshold conditions. Compared with the existing algorithms, our method has good practicality with an average reduction of 73.7% in the detection error rate.

Keywords: Bloom filter · Redundant filtering · Data cleaning · Radio frequency identification

1 Introduction

Radio Frequency Identification (RFID) is a non-contact automatic identification technology that mainly uses the backscattering characteristics of radio frequency signals to achieve automatic identification. It is widely used in surveillance scenarios due to its advantages of non-visible communication, low cost, and long-distance batch reading [1].

In surveillance scenarios, RFID devices are deployed in places that need to detect the transfer of equipment or documents, such as a warehouse or office entrance. RFID devices used for item monitoring typically capture data at high

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 582–594, 2022.
https://doi.org/10.1007/978-3-031-19214-2_49

frequencies to prevent loss. However, the frequent collection will lead to data redundancy to a certain extent. Due to the way RFID works, the same tag will be read multiple times if it stays within the read range of a fixed reader for a long time, resulting in a large number of duplicate and invalid data [2]. These redundant data occupy a large number of system storage resources and often lack practical value, which seriously reduces the operating efficiency of the system.

In order to effectively eliminate the redundant data in the RFID data stream, the traditional data redundancy processing technology is to store the collected data in the data warehouse or database and then return the query result [3]. However, with the expansion of the data scale, the query efficiency will decrease, and the real-time requirements will not be met. Due to the real-time and streaming characteristics of RFID data, redundant data filtering must be carried out in a limited space and time. Therefore, designing a lightweight and real-time redundancy data processing method will be the focus of research.

Bloom Filter has received extensive attention for its low memory occupation and efficient query [4]. For the uncertainty of RFID data, Liao et al. proposed an approximate probabilistic synthetic Bloom filter based on a block sliding window model [5]. Liu et al. combined the advantages of adaptive sliding window and Euclidean distance for filtering RFID stream data [6]. We have proposed a redundant data processing algorithm called TDBF based on Bloom Filter, which performs redundant filtering of data in time and distance dimensions [7]. The TDBF algorithm uses the received signal strength value (RSSI) to represent the distance from the tag to the reader roughly. However, in practical applications, due to the influence of environmental factors and multipath effects, the fluctuation of RSSI leads to a decrease in the accuracy of redundant processing. In addition, the threshold of redundant filtering in TDBF is often set empirically, which has limitations. Therefore, this research considers the complexity of the actual environment and illustrates how to set the filtering threshold adaptively.

We propose R-TDBF, a redundant data filtering method with a low error rate and high practicability. First, we construct a redundant filtering model for surveillance scenarios. Second, we illustrate how to set the threshold reasonably and adaptively adjust the threshold value according to the actual scenario. Finally, according to the redundancy definition, we perform real-time redundancy filtering on the RFID data stream. The results show that the influence of RSSI fluctuation is further eliminated in R-TDBF, resulting in a lower error rate and can be practically applied to different scenarios. The main advantages of R-TDBF compared to previous methods are as follows.

1. R-TDBF takes into account the influence of environmental factors. The practicability of the algorithm is effectively improved by setting the filtering threshold that adapts to the environment.
2. R-TDBF supports the processing of disturbing RSSI values to reduce the impact of RSSI fluctuations, thereby effectively reducing the detection error rate of redundant data.

3. Extensive experiments show that the proposed algorithm achieves real-time and efficient filtering of redundant data in the Spatio-temporal dimension and can be practically applied to surveillance scenarios.

2 Related Work

There is a broad interest in filtering the redundant data quickly and effectively. The literature [8] utilizes a finite state machine model for redundant data cleaning. The number of state machines can be limited and is not applicable in complex environments. In addition, with the popularity of machine learning, some data redundancy processing algorithms based on Bayesian networks and variational inference-based techniques [9,10] have emerged, which use prior knowledge to process data redundancy. However, these algorithms are more dependent on historical data and slower in redundancy decision making, so their applications are limited in places with strict real-time requirements.

Bloom Filter (BF) has received extensive attention in the field of RFID redundancy filtering due to its lightweight and real-time characteristics [11]. Bloom Filter is a probabilistic data structure consisting of a bit array of size m and k independent hash functions h_1,h_2,...,h_k, where the array is used to store data, and hash functions can be mapped to k different locations of the array. Bloom Filter can check whether the data is in the filter to achieve a redundant decision. Many improved algorithms based on Bloom Filter have been proposed [12]. Lee et al. proposed the Time Bloom Filter (TBF), which saves the timestamp of the latest data in an array and filters redundant data in the time dimension [13]. Wang et al. proposed the temporal-spatial bloom filter (TSBF) to handle redundant data from both temporal and spatial dimensions [14]. Zhu et al. proposed a redundant cleaning model R-TSBF, which introduced RSSI into the redundant judgment rules and reduced the error rate by retaining the maximum strength of the data [15]. Cao et al. proposed an algorithm called Time-Distance Bloom Filter (TDBF), which performs redundant filtering from the two dimensions to realize redundant processing of data within the surveillance range [7]. However, none of the above algorithms considers the threshold setting problem in redundant filtering, which is challenging to apply to different scenarios.

3 Scheme Overview

3.1 Related Definition

The relevant definitions of RFID redundant data in surveillance scenarios are given below.

Definition 1 (Surveillance data): Use S to denote a series of RFID data streams, $S = \{s_1, s_2, ..., s_n\}$, where s_i denotes an RFID triple $< TagID, Time, RSSI >$, where $TagID$ stands for the unique identifier of the tag, $Time$ is the timestamp of the data acquisition, and $RSSI$ is the signal strength value.

Definition 2 (Temporal redundancy): If there are two tag data $x \in S$, $y \in S$, if $x.TagID = y.TagID$, $x.Time > y.Time$ and $x.Time - y.Time \leq \tau$, where τ is a set time threshold. At this time, the data x is considered to be temporal redundant.

Definition 3 (Distance redundancy): If $x \in S$, $y \in S$ exists and the tag data x and y are, respectively, satisfied $x.TagID = y.TagID$, $x.Time - y.Time > 0$ and $x.RSSI < \epsilon$, where ϵ is the distance threshold. We use the RSSI value to represent the read distance roughly, and if it is smaller than this threshold, it indicates that the tag is far away. At this time, the data x is considered distance redundant.

Definition 4 (Strength redundancy): If $x \in S$, $y \in S$ exists and the tag data x and y are, respectively, satisfied $x.TagID = y.TagID$,$x.Time > y.Time$,$x.Time - y.Time \leq \tau$, when $x.RSSI - y.RSSI < \beta$, then the data x is strength redundancy, where β is the strength threshold, which is determined according to the specific application.

Definition 5 (Error rate): The formula of error rate (ER) is $N_f/(N_t + N_f)$, where N_t is the correctly judged RFID data, and N_f is the wrongly judged.

3.2 Redundancy Processing Framework

To further enhance the practicability of the algorithm, we propose an improved redundant data processing algorithm R-TDBF. The overall framework of R-TDBF mainly includes three stages(see Fig. 1).

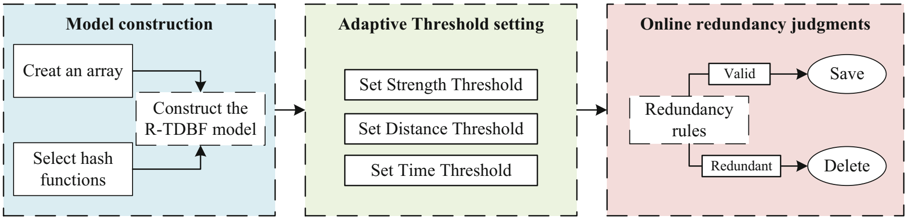


Fig. 1. R-TDBF redundant data processing model

Model Construction: The model construction phase is used to determine the redundant filtering structure of R-TDBF. R-TDBF adopts a two-dimensional integer array structure, which saves the tag reading time and the received signal strength value. The parameter $Time$ can filter the data with small time intervals, and the parameter $RSSI$ can filter the data beyond the detectable area.

Threshold Setting: The threshold setting stage is to determine the redundant filtering threshold according to the actual scene. First, we analyze the collected RSSI values to clarify the RSSI values' fluctuation range to determine the signal strength threshold. Second, we use the path loss model to get the relationship between distance and RSSI. The maximum likelihood estimation algorithm fits the model parameters adapted to the scene to determine the RSSI value corresponding to the distance threshold. Finally, we can determine the time threshold based on the speed of the monitored object.

Redundancy Judgment: This stage is the process of making redundant decisions for real-time data streams. R-TDBF introduces redundant judgment of signal strength to reduce the interference data caused by fluctuation of RSSI.

4 Detailed Scheme

4.1 Model Construction

The structure of R-TDBF is shown in Fig. 2. R-TDBF consists of a number of k hash functions and a two-dimensional array of size m, where the first dimension stores the timestamp $Time$ read by the tag, and the second dimension stores the tag received signal strength $RSSI$. The data of the i unit is represented as $M_i[Time][RSSI]$, where $M_i[Time] = Time_i$, $M_i[RSSI] = RSSI_i$.

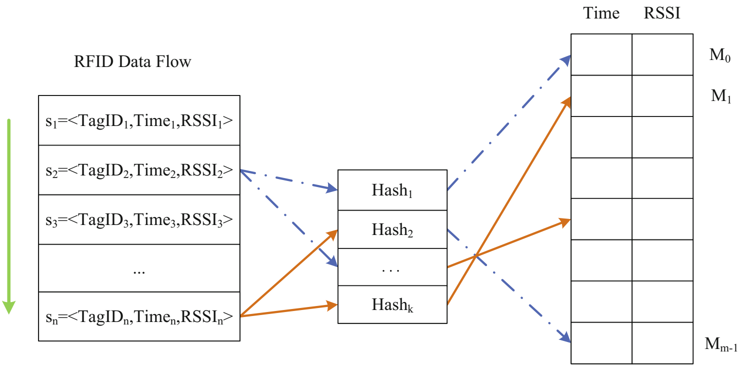

Time
RSSI
RFID Data Flow
M_0
M_1
s_1=<TagID$_1$,Time$_1$,RSSI$_1$>
s_2=<TagID$_2$,Time$_2$,RSSI$_2$>
s_3=<TagID$_3$,Time$_3$,RSSI$_3$>
...
s_n=<TagID$_n$,Time$_n$,RSSI$_n$>
Hash$_1$
Hash$_2$
...
Hash$_k$
M_{m-1}

Fig. 2. R-TDBF algorithm structure diagram.

4.2 Threshold Setting

Strength Threshold. Due to non-line-of-sight and multipath effects, the RSSI value in the actual scene has some volatility. We collected 300 sets of RSSI values at the same position (see Fig. 3). We found that the distribution of RSSI can be approximately considered to obey the Gaussian distribution through the experimental fitting, and its probability density function is as follows.

$$f(RSSI) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(RSSI-\mu)^2}{2\sigma^2}\right) \tag{1}$$

The mean μ and variance σ of the RSSI data distribution are as follows.

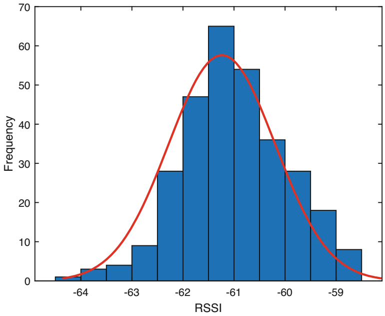


Fig. 3. Gaussian distribution of signal strength.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} RSSI(i)$$

,

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(RSSI(i)-\mu)^2}$$

We choose the region with a probability greater than 0.95 as the high probability occurrence interval (0.95 is the default confidence interval for normal distribution). From $0.95 \le P(RSSI(i)) \le 1$, the variation interval of RSSI can be obtained as follows.

$$-1.96\sigma + \mu \le RSSI(i) \le 1.96\sigma + \mu \tag{2}$$

We consider RSSI within the confidence interval to be reasonably volatile. Therefore, we get the strength threshold $\beta = 1.96\sigma$.

Distance Threshold. We achieve redundancy filtering in the spatial dimension by setting a distance threshold ϵ. In R-TDBF, we use the RSSI value to represent the tag-to-reader distance [16]. By setting a reasonable RSSI, the redundant filtering of the distance range is realized. We get the relationship between distance and RSSI according to the ranging model [17].

$$RSSI(d) = -(10nlgd + A) \tag{3}$$

where A is the reference RSSI value at 1 m, and n is the signal transmission constant, which is related to the signal propagation environment.

Table 1. Typical values of A, n in different environments [18]

Environment	A	n
Staircase balcony	33.4∼44.2	1.4∼2.4
Office	39.0∼50.5	1.4∼2.5
Hallway corridor	35.0∼38.2	1.9∼2.5
Lawn park	32.7∼36.0	3.0∼3.9

As shown in the Table 1, the values of A and n are different in different environments [18]. To make the model reflect the signal propagation characteristics as much as possible, optimising A and n to obtain the parameter values that best adapt to the environment is necessary.

Define $\rho_i = -10lgd_i, i = 1, 2, 3, ...N$. It can be seen from the Eq. 1 that the distribution of RSSI conforms to the normal distribution, so for each value of ρ, it satisfies $RSSI \sim N(-A + n\rho, \sigma^2)$. From the independence of RSSI, it can be known that the joint density of $RSSI_1, RSSI_2, ..., RSSI_N$ is as follows.

$$\begin{aligned} L &= \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma^2}\left(RSSI_i + A - n\rho_i\right)^2\right] \\ &= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(RSSI_i + A - n\rho_i\right)^2\right] \end{aligned} \tag{4}$$

Now use the maximum likelihood estimation method to estimate the unknown parameters A and n. Obviously, if L wants to take the maximum value, it only needs to satisfy the minimum function $Q(\rho, RSSI) = \sum_{i-1}^{N}(RSSI_i + A - n\rho_i)^2$. Take the partial derivatives of Q to A, n and make them equal to zero.

$$\begin{cases} \frac{\partial Q}{\partial A} = 2\sum_{i=1}^{N}(RSSI_i + A - n\rho_i) = 0 \\ \frac{\partial Q}{\partial n} = -2\sum_{i=1}^{N}(RSSI_i + A - n\rho_i)\rho_i = 0 \end{cases} \tag{5}$$

According to the formula 5, we can get the value of A, n. Therefore, we can calculate the best distance estimate according to different scenarios to achieve redundant filtering in different ranges.

Time Threshold. The time threshold τ is related to speed. Assuming that the movement speed of the tag is v, then as long as $\tau \leq \epsilon/v$ is satisfied, the valid tag data can be read at least once within the monitoring distance ϵ. In actual scenarios, if τ is too large, important monitoring data may be missed, and if τ is too small, it is difficult to achieve efficient redundant filtering. Therefore, setting a reasonable τ is crucial.

4.3 Online Redundancy Judgment

When new RFID data x arrives, R-TDBF judges whether the data is redundant in the temporal-spatial dimension. The redundant judgment process is shown in Algorithm 1.

1. Determine the redundant filtering threshold in the actual scene, including the time threshold τ and the distance threshold ϵ, and estimate the RSSI value corresponding to the distance threshold and the strength threshold β.
2. k independent hash functions are used for mapping its unique ID into the corresponding array unit.
3. If there is $i \in 1, 2, ..., k$, $M_i[Time] = 0$, $x.RSSI > \epsilon$, then we can think of this data x as the new arrival data within the detection range,. At this time, the time information and signal strength values in k different array units are updated, which is $M_i[Time] = x.Time$, $M_i[RSSI] = x.RSSI$.
4. If there is $i \in 1, 2, ..., k$, $x.Time - M_i[Time] > \tau$, $x.RSSI > \epsilon$, then the newly arrived data is outside the specified time interval and within the distance range, so data x is not redundant. At this time, the time information and signal strength value in k different array units are updated, which is $M_i[Time] = x.Time$, $M_i[RSSI] = x.RSSI$.
5. If there is $i \in 1, 2, ..., k$, $x.Time - M_i[Time] < \tau$, $x.RSSI > \epsilon$, $x.RSSI - M_i[RSSI] > \beta$, then the $RSSI$ of the newly arrived data is within the detection range and the $RSSI$increments exceed the fluctuation threshold β. It is reasonable to assume that the data x is anomalous and valid. At this time, the time information and signal strength value in k different array units are updated, twhich is $M_i[Time] = x.Time$, $M_i[RSSI] = x.RSSI$.
6. Otherwise, x is considered to be redundant and is deleted directly.

Algorithm 1 R-TDBF redundancy processing algorithm

```
Require: RFID data $x$:$x.TagID, x.Time, x.RSSI$
Ensure: Whether $x$ is redundant
Set the filtering threshold $\tau$,$\epsilon$,$\beta$
Seletc $k$ Hash function
for $i = 0$ to $k$ do
    $p[i] = Hash_i(x.TagID)$
    if $R\text{-}TDBF_{p[i]}[Time] = 0$ and $x.RSSI > \epsilon$ then
        Update $R\text{-}TDBF(x.Time, x.RSSI)$
    else if $x.Time - R\text{-}TDBF_{p[i]}[Time] > \tau$ and $x.RSSI > \epsilon$ then
        Update $R\text{-}TDBF(x.Time, x.RSSI)$
    else if $x.Time > R\text{-}TDBF_{p[i]}[Time]$ and $x.RSSI - R\text{-}TDBF_{p[i]}[RSSI] > \beta$ then
        Update $R\text{-}TDBF(x.Time, x.RSSI)$
    else
        Drop $x$
    end if
end for
```

5 Evaluation

This experiment uses an Impinj Revolution series passive reader and several supporting tags for data acquisition. We attach multiple RFID tags to the foam board to simulate the monitoring item with the attached tag. We use the slide rail to simulate the trajectory of the monitoring object entering and leaving(see Fig. 4). The car on the slide rail can freely set the speed.

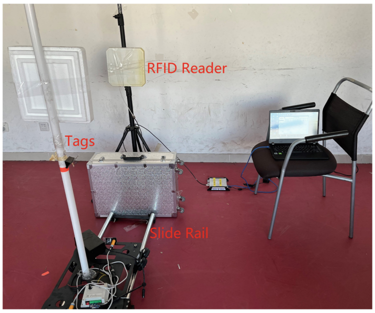


Fig. 4. Experimental scenario.

5.1 Filtration Performance Evaluation

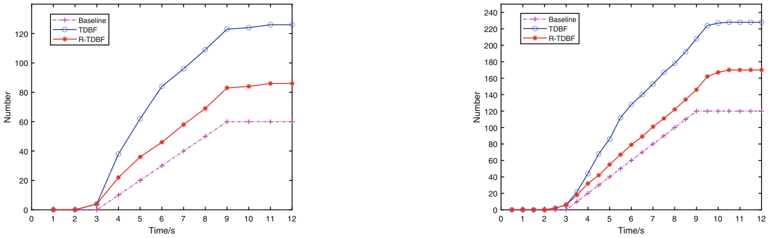
Analysis of Filtering Performance with Different Distance Thresholds. This experiment explores the redundant filtering performance of R-TDBF under different distance thresholds. The ten tags in the experiment reciprocate on a 3 m-long slide rail at a speed of 0.5 m/s. When τ=1 s, β=2 dB, we set the ϵ to 1 m, 2 m, and 3 m respectively (the estimated RSSI values are −39 dB, -48.9 dB, −54.7 dB respectively). Since the TDBF algorithm cannot estimate the distance adaptively, we adopt the same setting as R-TDBF.

Table 2. Comparison of filtering performance under different distance thresholds

	Raw data	Baseline	TDBF	R-TDBF
1 m	47162	2000	3932	2378
2 m	47162	4000	7273	4828
3 m	47162	6000	12267	7624

As shown in Table 2, baseline shows the results when RSSI does not fluctuate under ideal conditions, which is the upper limit of performance. The TDBF and R-TDBF algorithms show the results of data redundancy filtering in different detection ranges. The smaller the distance threshold, the better the redundancy filtering performance. The filtering performance of R-TDBF is better than that of the TDBF and is closer to the baseline under the same distance threshold. Therefore, the R-TDBF algorithm can estimate the detection range adaptively in the surveillance scene and perform well.

Analysis of Filtering Performance with Different Time Thresholds. This experiment verifies the effectiveness of R-TDBF by observing the redundant filtering effect under different time thresholds. We set ϵ=1.5 m (RSSI estimated value is −44.8 dB), β=2 dB, and the time threshold τ is set to ,s and 0.5 s, respectively. We let ten tags enter the detection range at a speed of 0.5 m/s along a 3 m long rail and then leave.



(a) (b)

Fig. 5. Comparison of redundant processing performance under different time thresholds. (a) The time threshold is 1 s (b) The time threshold is 0.5 s.

The curve of the total amount of data filtered by different time thresholds is depicted in Fig 5. The baseline represents the ideal filter curve. Under different time thresholds, both TDBF and R-TDBF can effectively filter out the data outside the detection range. However, as the tags enter the detection range, the TDBF filtering performance is affected by RSSI fluctuations. The R-TDBF algorithm introduces the strength threshold that can reduce the interference caused by the fluctuation of the RSSI value and achieve good filtering within the detection range. Experiments show that the R-TDBF algorithm can perform redundant filtering without losing data information, and its performance is better than the TDBF algorithm and closer to the ideal result.

5.2 Error Rate Analysis

The error rate (ER) of R-TDBF includes false negatives and false positives. False negative refers to judging RFID tag data belonging to redundant data as non-redundant and False positive refers to judging those RFID tags that do not belong to redundant data as redundant data. Hash collisions and RSSI fluctuations are the main contributors to the error rate.

The Eq. 6 shows the false positives of R-TDBF due to hash collision, where m represents the size of the R-TDBF array, k represents the number of hash functions, and n represents the added data total amount. This error rate is because R-TDBF may map $TagID$ to k array positions that have changed within the time threshold τ. Since the probability of this error is too small, its error rate is negligible in practical scenarios.

$$P(R\text{-}TDBF) = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^{k} \tag{6}$$

In addition, the fluctuation of RSSI may cause it to fail to reflect the distance accurately or the fluctuation range may exceed the set threshold, which is the main reason for the error rate in practical scenarios. For example, the data within the distance threshold is misjudged as outside. Therefore, we conduct experiments to explore the effect of RSSI fluctuations on the error rate.

Error Rate Analysis for Different Distance Thresholds. This experiment compares the error rate by setting different distance thresholds. We fixed τ=1 s, β=2 dB, and collected the data of the car moving freely for 10 min. When the distance threshold ϵ changes, its error rate changes(see Fig. 6).

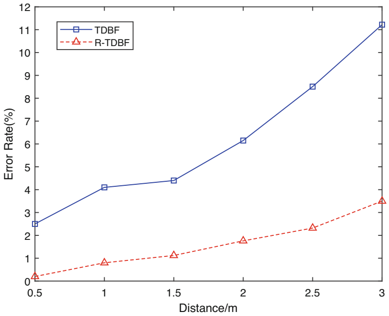


Fig. 6. Comparison of error rates for different distance thresholds.

Since TDBF does not consider RSSI fluctuations in redundant filtering, as the distance increases, the fluctuation range becomes larger and the error rate increases significantly. However, R-TDBF has a lower error rate and maintains a relatively stable trend under different distance thresholds. Experiments demonstrate that R-TDBF maintains a low error rate when the distance threshold changes.

Error Rate Analysis in Different Scenarios. This experiment compares the error rate by changing the environment. We test the error rates in three scenarios: laboratory, corridor, and outdoor. We fixed τ=1 s, ϵ=1.5 m (typical detection range), and collected the data of the car moving freely for 10 min.

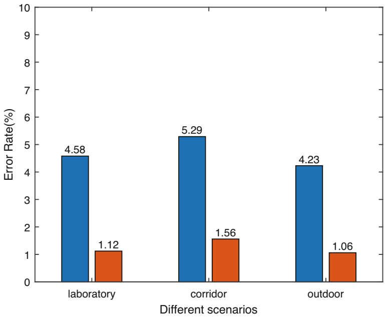


Fig. 7. Comparison of error rates in different scenarios.

The RSSI estimates in the scene are −44.8 dB, −42.6 dB, and −40.6 dB, respectively, and the strength threshold β is set to 2 dB, 1.8 dB, and 1.8 dB, respectively. The error rates in different scenarios are shown in Fig 7. By setting a reasonable threshold, the R-TDBF algorithm produces a lower error rate than the TDBF. The laboratory error rate drops by 75.6%, the building error rate drops by 70.5%, and the outdoor error rate drops by 74.9%. The error rate in different scenarios dropped by 73.7% on average. Experiments show that R-TDBF can greatly reduce the detection error rate compared with TDBF when the environment changes.

6 Conclusion

The generation of redundant data hinders efficient data processing in RFID-based surveillance scenarios. The R-TDBF algorithm proposed in this paper can filter the redundant data in the RFID data stream in real-time, reducing the pressure of data transmission and upper-layer application analysis. R-TDBF realizes data redundancy filtering by setting reasonable thresholds. In addition, a strength threshold is also introduced in R-TDBF, which reduces the error caused by signal fluctuation. The experimental results show that the R-TDBF algorithm can filter redundant data well under different distance and time thresholds. Compared with the TDBF algorithm, R-TDBF reduces the detection error rate by 73.7% on average, which has good practicability.

Acknowledgement. This work was supported by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDC02040300

References

1. Derakhshan, R., Orlowska, M.E., Li, X.: RFID data management: challenges and opportunities. In: IEEE International Conference on RFID, pp. 175–182 (2007)
2. Mahdin, H., Abawajy, J.: An approach for removing redundant data from RFID data streams. Sensors **11**(10), 9863–9877 (2011)

3. Gonzalez. H, Han, J, Li, X., et al.: Warehousing and analyzing massive RFID data sets. In: International Conference on Data Engineering IEEE(ICDE), pp. 83–90 (2006)
4. Bloom, B.: Space/Time tradeoffs in hash coding with Al-lowable errors. Ipsj Magazine **13**, 422–426 (1970)
5. Guoqiong, L., Jun, Z., Ni, H., Xiaomei, H., Zhiwei, H., Chang-xuan, W.: Approximately filtering redundant data for uncertain RFID data streams. In: IEEE International Conference on Mobile Data Management, pp. 56–61 (2017)
6. Liu, L.-L., Yuan, Z.-L., Liu, X.-W., Chen, C., Wang, K.-S.: RFID unreliable data filtering by integrating adaptive sliding window and Euclidean distance. Adv. Manuf. **2**(2), 121–129 (2014). https://doi.org/10.1007/s40436-014-0080-3
7. Wang, S., Cao, Z., Zhang, Y., Huang, W., Jiang, J.: A Temporal and Spatial Data Redundancy Processing Algorithm for RFID Surveillance Data, Wireless Communications and Mobile Computing (2020)
8. Luo, Y.J., et al.: Filtering and cleaning for RFID streaming data technology based on finite state machine. J. Softw. **25**(8), 1713–1728 (2014)
9. Wang, R.C., et al.: A method of cleaning RFID data streams based on Naive Bayes classifier. Int. J. Ad Hoc And Ubiquit. Comput.: IJAHUC **21**(4), 237–244 (2016)
10. Yousif, A., Kafafy, A., Abdlkader, H.M.: Reducing RFID data uncertainty using mean field variational inference. In: 2018 14th International Computer Engineering Conference (ICENCO), pp. 131–136 (2018)
11. Mahdin, H.: A review on bloom filter based approaches for RFID data cleaning. In: Herawan, T., Deris, M.M., Abawajy, J. (eds.) Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013). LNEE, vol. 285, pp. 79–86. Springer, Singapore (2014). https://doi.org/10.1007/978-981-4585-18-7_9
12. Mahdin, H, Abawajy, J.: An approach for removing redundant data from RFID data streams. Sensors **11**(10), 9863–9877 (2011)
13. Lee, C.H., Chung, C.W.: An approximate duplicate elimination in RFID data streams. Data Knowl. Eng. **285**(12), 1070–1087 (2011)
14. Wang, Y.L., Wang, C., Jiang, X.H.: RFID duplicate removing algorithm based on temporal-spatial Bloom filter. J. Nanjing Univ. Sci. Technol. **39**(3), 253–259 (2015)
15. Zhu, W.L.: Longitudinal Positioning of Warehouse Packages Based on Redundant Cleaning and Clustering of RFID Data, Chongqing University (2017)
16. Pinto, B., et al.: Robust RSSI-based indoor positioning system using k-means clustering and Bayesian estimation. IEEE Sensors J. **21**(21), 24462–24470 (2021)
17. Li, G., et al.: Indoor positioning algorithm based on the improved RSSI distance model. Sensors **18**(9) (2018)
18. Seidel, S.Y., Rappaport, T.S.: 914 MHz path loss prediction models for indoor wireless communications in multifloored buildings. IEEE Trans. Antennas Propag. **40**(2), 207–217 (2002)

Users' Departure Time Prediction Based on Light Gradient Boosting Decision Tree

Lingyu Zhang[1,2], Zhijie He[2], Xiao Wang[2], Ying Zhang[2], Jian Liang[2], Guobin Wu[2], Ziqiang Yu[3], Penghui Zhang[4], Minghao Ji[4], Pengfei Xu[4], and Yunhai Wang[1](✉)

[1] School of Computer Science and Technology, Shandong University, Qingdao, China
cloudseawang@gmail.com
[2] Didi Chuxing, Beijing, China
{hezhijie_i,yingzhangying,liangjian,wuguobin}@didiglobal.com,
wangxiao@didichuxing.com
[3] Yantai University, Yantai, China
zqyu@ytu.edu.cn
[4] School of Information Science and Technology, Northwest University,
Kirkland, USA
pfxu@nwu.edu.cn

Abstract. With the development of urban transportation networks, the flow of people in cities generally shows the characteristics of concentration, periodicity and irregularity, and a typical example is rush hour. For most existing taxi-hailing apps, users frequently queue up for a relatively long time during rush hour and may even fail to get orders taken due to various factors. To solve this problem, we propose a users' departure time prediction model based on Light Gradient Boosting Machine (TP-LightGBM), which will remind users to book taxis before their journeys. As we know, TP-LightGBM may be the first model for departure time prediction. We uncover that travel behavior patterns vary under different external conditions through statistics and analysis of users' historical orders from multiple perspectives. Furthermore, we extract multiple features from these orders and select the favorable features by calculating their information gain as the input of TP-LightGBM to predict users' departure time. Therefore, our model can provide users with the recommendations of the best departure time if they need them. The final experimental results on our datasets indicate that TP-LightGBM has more excellent performance with great stability in predicting user departure time than other baseline models.

Keywords: Departure time prediction · Light gradient boosting machine · Data analysis · Feature engineering · Loss assessment

1 Introduction

The accelerating development of Smart City has put forward new requirements for Intelligent Transportation System and Smart Travel, and the big data on travel provides strong support for related researches. As an essential part of intelligent travel, accurate travel time prediction is crucial. Specifically, user departure time prediction refers to

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 595–605, 2022.
https://doi.org/10.1007/978-3-031-19214-2_50

data mining from numerous users' historical travel records to predict where and when users may have travel plans, so as to timely remind them to arrange travel or book taxis in advance. As users cannot grasp the specific information of actual traffic flow, it is difficult to reasonably choose the departure time and travel modes, which is the crux of contradiction and is also one of the main problems to be solved in Smart Travel.

At present, the increasingly severe traffic jam causes great inconvenience to people's daily travel, especially at some specific time such as the rush hour. Due to the diversity of individual travel behavior and the complexity of traffic information, the model of departure time prediction based on historical travel reports cannot always be accurate. At present, there are few works about users' departure time prediction. While, the existing relevant models are passive statistical models, which passively predict the future through the statistics of actual historical data and analyzing their patterns. The insufficient information on future travel rein in the performance of models, which leads to that such models can not consistently maintain high prediction accuracy. However, the time prediction models based on deep learning require a large amount of data and complex calculations. Although they can achieve sound prediction effects, it is difficult for these models to guarantee real-time performance when many users are online.

Therefore, a users' departure time prediction model based on Light Gradient Boosting Machine (TP-LightGBM) is proposed in this paper. TP-LightGBM can be used to remind users to arrange travel and to book taxis in advance within a reasonable time. The prediction results can help users choose their optimal departure time and travel patterns more freely to reduce information delay, and also can avoid congestion and significantly improve the quality and efficiency of users' travel. Of course, whether to provide relevant services is determined according to users' requirements on the recommendations of the best departure time.

2 Related Works

Users' departure time prediction is a necessary function of intelligent transportation and is also an essential part of intelligent city construction. With the rapid development of Intelligent Traffic Systems (ITS), various machine learning algorithms have contributed to traffic data reconstruction, traffic flow prediction, urban traffic pattern mining, and so on.

There are many methods to predict travel time in previous works, most of which focus on the travel time prediction of vehicles on the road to assist traffic control, yet few works are about users' departure time prediction. For example, Chien et al. proposed a prediction model of bus arrival time based on an artificial neural network by using the data of trajectories and bus stops [6]. This model for arrival time and location prediction combines an artificial neural network, and Kalman filter [4]. It estimates the arrival time and updates the real-time locations of vehicles according to the data of automatic passenger counters. In addition, many other works focusing on the prediction of the travel time of vehicles on the road make it more convenient to analyze the traffic flow [1,18,20]. For example, combined with Decision Tree and Linear Regression, future highway travel time can be predicted based on flow and occupancy data [13]. Besides, to solve the problem that the travel time is just a simple addition of link time, the data for modeling is path-based rather than link-based [3]. Meanwhile, the Kalman filter is introduced, and through continuous updating state variables as the new observation variable to predict the traffic on the motorway driving time [7]. As to particular unpredictable events, a Bayesian dynamic linear learning model [10] was proposed, which could adjust the parameter settings and noise level adaptively.

The rising Machine Learning and Deep Learning methods in recent years shed light on a new way to predict travel time. Duan et al. established an LSTM network for each link [9], which verified the prospects of the deep learning model considering the time-series relationship in travel time prediction. In addition, Gradient Boosting Decision Tree (GBDT) is applied to analyzing and modeling the travel time of highway vehicles [21] and discussing the impact of different parameters on the model's performance. Gradient Boosting Tree(GBT) is a boosting method based on the weak learner of tree model [14] pertaining two typical usages as Gradient Boosting Decision Tree (GBDT) and Gradient Boosting Regression Tree (GBRT). GBDT can be applied to the prediction [5,8,19] and classification [17] problems, and it can effectively merge different types of variables and fit complex nonlinear relationships. However, rather low efficiency is always a demerit of GBDT, especially with large-scale features and big data. For this problem, a gradient-based unilateral sampling method [12] is offered using the information gain of samples with larger gradients to estimate the overall information gain to improve efficiency with little compromised accuracy. Meanwhile, a feature selection method based on artificial bee colonies and GBDT was presented in [16], which globally optimized the feature space to enhance the efficiency and quality. Besides, Light Gradient Boosting Machine (LightGBM) [12] downsizes the features by bundling mutual exclusive features and downs sample the data instances by keeping all instances with big gradients and randomly sampling instances with small gradients, which reduces the number without changing the distribution of original data by much.

In summary, to improve the performance of individual travel time prediction, we introduce the LightGBM to predict the user's departure time based on the historical taxi orders of Didi Chuxing's users. More specifically, we have trained a model with individual characteristics for each user based on his/her historical orders of Didi Chuxing, which will remind users to book a taxi in advance before needed. And experiments prove that our model is not sensitive to the independence between diverse features and can correctly fit complex feature relationships.

3 The Analysis of Users' Departure Time and Travel Behaviors

3.1 The Overall Information of Users' Historical Taxi Orders

In this section, we analyze the users' historical taxi orders from multiple perspectives and visualize the results of the data analysis. The dataset used in this paper is the historical taxi orders of users who used Didi Chuxing online taxi-hailing platform, and the information of each sample mainly includes the user ID, order ID, time, locations, date attribute, and so on. Where, time (Notation as T) is the specific time of a day, and has been processed into the form of periods with hourly granularity. The range of T is in [0, 23). Besides, we also transform the original time into a day of the week (Notation as W), so a type of time-series data can be used to mine the regularity of users' travel behaviors, and the range of W is in [0,6]. Each location name is mapped to a corresponding pair of longitude and latitude. The range of the longitude (Notation as Lng) is in [-180, +180], and the range of Latitude (Notation as Lat) is in [-90, 90]. The attribute values of the date (Notation as D) can be 0 or 1, and 1 stands for working days while 0 stands for holidays. In addition, all the orders have been anonymized and aggregated, and we correct the latitude and longitude of all locations and delete historical orders with abnormal order status.

3.2 The Analysis of Users' Travel Behaviors

Users' travel time shows noticeable regularity in a certain period. For example, users have a relatively regular commute time on weekdays and leisure time to go out on holidays, and even have regular travel times at several certain workplaces. Therefore, it is suitable for us to estimate the users' departure time using their historical orders. Users' travel behavior also shows strong regularities in the spatial domain. The following discussion is only a starting point, and we can draw similar conclusions in terms of destination. The spatial distribution of historical orders shows strong sparseness and concentration. Most of the starting points are concentrated in certain areas, while some others only appear once.

Moreover, we discover that users' travel time shows strong regularity in the spatial domain. The users' departure time in some places may concentrate on one specific time period. In addition, there are more cases of calling taxis in similar places in a similar time period, although there have been some effects of departure time on different starting points. For example, the users' taxi-hailing locations may be primarily residential areas in the morning, while workplaces, commercial areas, and entertainment venues at night.

The analysis result above is drawn from users' historical orders and reveals some commonalities in users' taxi booking behaviors: (1) The taxi-hailing time distribution of the same user tends to show a concentrated distribution in specific locations and times rather than a uniform distribution. (2) Most users have distinct travel patterns between workdays and holidays, and there are differences in users' departure times when the day type changes. (3) The same user tends to set the same destination in a certain period and rarely book a taxi in other periods. (4) Most users tend to go to a certain place within a fixed time.

3.3 Feature Selection

Feature selection plays a vital role in feature engineering. Due to the limited samples and the sparsity of distribution of users' historical orders, we use as few features as possible to predict users' departure time so as to avoid the high computational complexity and performance degradation of models caused by large-scale features. We list several candidate features which affect users' travel time, as shown in Table 1.

Table 1. Information gain (ratio) of each feature.

Feature	IG	IGR
Origin longitude	1.93	0.47
Origin latitude	1.92	0.47
Destination longitude	1.71	0.43
Destination latitude	1.81	0.33
Date attribute	0.11	0.14
Day of the week	0.55	0.42

Furthermore, we apply the feature selection method based on the Decision Tree. More specifically, we calculate the Information Gain (IG) and Information Gain Ratio (IGR) of each feature (see Table 1), which are respectively used in the module of ID3 [14], and C4.5 [15]. The methods to calculate IG and IGR are stated below:

Assume that the dataset is D, which has the size of $|D|$. The samples in D are divided into K categories $C_k (k = 1, 2, ..., K)$, and there are $|C_k|$ samples in class C_k. Then we assume that feature A can take n different values $a_1, a_2, ..., a_n$, which can divide D into n subsets $D_1, D_2, ..., D_n$, and $|D_i|$ $(i = 1, 2, ..., n)$ is the number of samples in D_i. In addition, let D_{ik} denotes the sample set of category k in subset D_i, and its size is denoted by $|D_{ik}|$. Thus, the information gained can be written as

$$g(D, A) = H(D) - H(D|A) \tag{1}$$

where, $H(D)$ is the empirical entropy of D, and $H(D|A)$ is the empirical conditional entropy of feature A to dataset D, $H(D)$ is calculated by

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} log_2 \frac{|C_k|}{|D|} \tag{2}$$

and $H(D|A)$ is

$$H(D|A) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} log_2 \frac{|D_{ik}|}{|D_i|} \tag{3}$$

The information gain ratio can be calculated by $g_R(D, A) = \frac{g(D,A)}{H_A(D)}$, where $H_A(D)$ is the entropy of the dataset D for feature A:

$$H_A(D) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} log_2 \frac{|D_i|}{|D|} \tag{4}$$

If the feature has a more significant Information Gain (Ratio), it will be more influential for classification and have a stronger ability to classify the samples. From the results in Table 1, IG and IGR of date attributes are both the smallest and should be discarded, while others should be retained in principle. However, the users' destinations are normally unknown in the actual scenario. If we first predict the destination and then the departure time according to the attribute of date, the cost will inevitably arise, and the accuracy cannot be guaranteed. Moreover, time features are weighty in the travel prediction. As we analyzed before, the feature day of the week contains information of regularity. Therefore, we decide to retain the attribute of date and day of the week and then elide the longitude and latitude of the destination. The experiments also demonstrate that the outcome using four features of origin longitude, latitude, day of the week, and date attribute is sounder than the origin longitude and latitude alone.

4 Time Prediction Model for Predicting Users' Departure Time

Through the in-depth analysis of orders of Didi Chuxing users, we convert users' departure time prediction into a multivariate classification problem. Users are classified according to their objective features using the category label of departure time. We use the hourly granularity as the classification standard and divide the users' historical orders into 24 categories.

4.1 Model Description

The probability of a user traveling at a fixed time period can be expressed in the form of conditional probability using Bayes' theorem:

$$P(T = t_i|X) = \frac{P(X|T = t_i)P(T = t_i)}{\sum_{i=1}^{24} P(X|T = t_i)P(T = t_i)} \tag{5}$$

where $X = Lng, Lat, D$. The process of solving the conditional probability $P(X|T = t_i)$ is extremely complicated, but the calculation difficulty will be greatly reduced if the method of conditional independent assumption of features in the naive Bayes algorithm is adopted, i.e. $P(X|T = t_i) = P(Lng|T = t_i)P(LatT = t_i|)P(D|T = t_i)P(W|T = t_i)$.

However, the features extracted from actual data are not as independent as the ideal assumption. Specifically, the latitude Lat and longitude Lng in the users' historical orders always emerge in pairs. For example, if location A often appears in one user's historical orders, then the latitude Lat_A and longitude Lng_A of location A have a highly correlated relationship, which does not meet the premise of conditional independence of each feature in the Naive Bayes algorithm. Moreover, taxi-hailing actions are purely personal behaviors, and regularity and irregularity coexist. The time distribution of taxi rides of a sample user may be evenly distributed throughout a day, which would confound the final prediction. Therefore, Gradient Boosting Decision Tree (GBDT) [11] is a suitable method for users' departure time prediction due to less demanding input features. However, the performance is unsatisfactory when the size of the data balloon. To balance this drawback, we introduce Light Gradient Boosting Machine (LightGBM) [12] which has an excellent performance to deal with a large number of data instances.

4.2 Departure Time Prediction Based on Light Gradient Boosting Machine (TP-LightGBM)

Decision Tree [14] is a primary classification and regression method, and its classification rules can be seen as a grouping of a series of if-then conditional statements or as a conditional probability model defined on features and class space. GBDT is a boosting algorithm based on the Classification and Regression Tree (CART) [2] and is one of the most widely used classification algorithms with high precision. Its main idea is to fit the residual of the previous base learner through the negative gradient of the loss function so that the residual estimation of each round declines. GBDT combines Gradient Boosting and Decision Tree to establish a new decision tree model (weak classifiers) in the gradient direction of the previous model residual reduction at each iteration. Finally, a well-trained GBDT classification model is a linear combination of these weak classifiers with different weights. The conventional implementation of GBDT is scanning all the instances for every feature to locate the optimal split points, which is time-consuming with big data. LightGBM based on GBDT proposes two techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to reduce computational complexities. GOSS downsizes the instances by keeping them with large gradients and randomly chooses instances with slight gradients. EFB decreases the features by bundling mutually exclusive features.

The training process of multi-class LightGBM can be viewed as an additive model, as shown in Algorithm 1. In practice, the multi-class LightGBM generates a tree for each category during the training process, i.e., a total of $K \times M$ sub-trees are generated in Algorithm 1, and Softmax obtains the final category result. Specifically, the loss function we choose is log-likelihood, which can be written as

$$L(y, f(x)) = -\sum_{k=1}^{K} y_k log(p_k(x)) \tag{6}$$

where y denotes the actual value of a sample, $f(x)$ is the predictive value, $p_k(x)$ represents the probability that the sample belongs to the category k.

5 The Experimental Results and Analysis

Due to the coexistence of regularity and uncertainty in user travel, we delete orders whose starting point appeared less than five times in a month. Since each user's travel pattern is unique and it is impossible to select every user who travels regularly, we set a threshold τ to filter the prediction results. The result will be output if its probability surpasses the threshold τ. Our purpose is to predict the period for taxi-hailing of the users of Didi Chuxing and does not involve a specific timestamp. Therefore, we take the prediction time as the midpoint to extend one hour as a period for the final result, i.e., so the final result will be expressed in $[t-1, t+1]$ if the output result is t, and the actual label of a test sample is regarded as a correct prediction if falls within the interval $[t-1, t+1]$.

Algorithm 1: The training process of GBDT classifier

```
Input: iterations (number of weak classifiers) M,number of samples N,
number of categorise K,loss function $L(y, f(x))$,training set
$T_{train} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$
Output: GBDT classifier $\hat{f}(x)$
Initialize:weak classifier $f_0(x) = arg\ \underset{\theta}{min} \sum_{i=1}^{N} L(y_i, \theta)$;
while $m = 1, 2, \ldots, M$ do
    for $i = 1, 2, \ldots, N$ do
        for $k = 1, 2, \ldots, K$ do
            // Calculate the probability of $x_i \subseteq class\ k$
            $p_k(x_i) = \left[\frac{exp(f_k(x_i))}{\sum_{k=1}^{K} exp(f_k(x_i))}\right]_{f_k(x)=f_{k,m-1}(x)}$;
            // Calculate negatice gradient error
            $r_{mik} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f_k(x)=f_{k,m-1}(x)} = y_{ik} - p_k(x_i)$;
        end
    end
    // Fit the decision tree
    use $r_{mik}$ to fit the decision tree, which leaf node area is $R_{mjk}$,
    $j = 1, 2, \ldots, J, k = 1, 2, \ldots, K$;
    // Estimate the gain of leaf nodes
    for $j = 1, 2, \ldots, J$ do
        for $k = 1, 2, \ldots, K$ do
            $\theta_{mjk} = \frac{K-1}{K} \frac{\sum\limits_{x_i \in R_{mjk}} r_{mjk}}{\sum\limits_{x_i \in R_{mjk}} |r_{mjk}| (1 - |r_{mjk}|)}$;
        end
    end
    // Update classification tree
    $f_{km}(x) = f_{k,m-1}(x) + \sum_{j=1}^{J} \theta_{mjk} I(x_i \in R_{mjk}), k = 1, 2, \ldots, K$;
end
// Output GBDT classifier
$\hat{f}_k(x) = f_{kM}(x) = \sum_{m=1}^{M} \sum_{j=1}^{J} \theta_{mjk} I(x_i \in R_{mjk}), k = 1, 2, \ldots, K$;
```

5.1 Experimental Setups

We randomly select 80% of the dataset as the training set and the other 20% as the test set, then multiple experiments are conducted using our model under different thresholds, and the results are shown in Fig. 1. We apply two metrics $PDP = N_{out}/N_{test}$ and $AUC = N_{auc}/N_{out}$ to measure the performance of the model. Where N_{test} denotes the number of samples in the test set, N_{out} is the number of samples with the output results, and N_{auc} denotes the number of samples accurately predicted. It can be seen that AUC and PDP are proportional and inversely proportional to the threshold τ respectively, and the growth rate of AUC slows down, but PDP still has a strong downward trend when $\tau > 0.7$. Therefore, we finally adopt the threshold $\tau = 0.7$ to test our model's all-around performance.

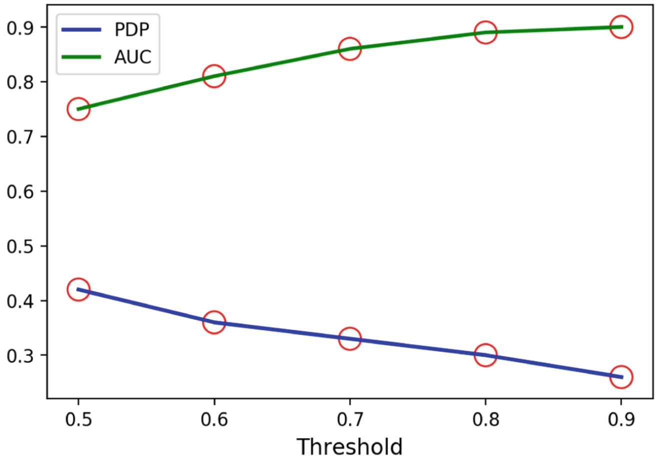


Fig. 1. An example of indirect blocking

5.2 Experimental Results and Comparative Analysis

In order to verify the superior performance of our model, GBDT, XGBoost, MultinomialNB, GaussianNB, BernoulliNB, and ModeMod are used as the comparison models. PDP, AUC, $Kappa$, $Hamming(H)$ and $Time(T)$ in Table 2 are used as the metrics of model performance evaluation. Here, $Hamming$ and $Time$ respectively represent Hamming distance and average time consumption of the models. $Kappa$ (Kappa coefficient) is often used to evaluate prediction accuracy and consistency, and it can be defined as $K_{appa} = (p_o - p_e)/(1 - p_e)$. Where p_o is the sum of the number of samples correctly classified in each category divided by the total number of samples, and p_e is the sum of the products of the actual and predicted sample numbers corresponding to all categories divided by the square of the total number of samples. Therefore, the prediction accuracy is positively correlated with the value of Kappa. Hamming distances measure the distance between the predicted label and the actual label. Thus, the prediction accuracy is negatively correlated with the value of Hamming distance. It needs to be stated that all metrics in Table 2 are the average values obtained from all test samples.

From Table 2, It can be seen that ModeMod has the most promising performance in terms of PDP and $Time$ metrics. ModeMod searches for orders that match the user's current status from the historical taxi-hailing orders and extract the departure times that meet the conditions. Then the departure time with the most occurrences is the predicted value. In this way, ModeMod shows the best stability and the lowest time complexity. In addition, the naive Bayes models (MultinomialNB, GaussianNB, and BernoulliNB) have a higher prediction accuracy than ModeMod. However, as we mentioned before, the conditional independence relationship between each feature is hard to achieve, and

Table 2. Experimental results on different models.

	PDP	AUC	$Kappa$	H	T(s)
TP-LightGBM	0.35	**0.90**	**0.64**	**0.14**	0.22
GBDT	0.33	0.87	0.58	0.16	0.74
XGBoost	0.33	0.88	0.62	0.17	0.51
MultinomialNB	0.84	0.54	0.16	0.44	0.107
GaussianNB	0.61	0.30	0.23	0.70	0.10
BernoulliNB	0.74	0.55	0.24	0.45	0.10
ModeMod	**1.00**	0.37	0.30	0.63	**0.09**

the distribution of features is difficult to determine. In contrast, the evaluation parameters of PDP, AUC, $Kappa$, and $Hamming$ are much better than the set of naive Bayes models, although the set of GBDT models has the highest time consumption, which TP-LightGBM can solve. In conclusion, TP-LightGBM only serves about 35% of orders, but the prediction accuracy has reached 92%. The main idea of TP-LightGBM is that the GBDT algorithm requires multiple iterations to fit data and train different weak classifiers, so it has a higher time consumption, but the average prediction time for each order can still be restrained within one second.

6 Conclusion and Future Work

Users' departure time prediction is an application of Machine Learning to Smart City construction. Accurately predicting users' departure times can remind them to call a taxi in advance and avoid queuing up during the rush hour. In this paper, we conduct a multi-perspective analysis and pattern discovery on the historical taxi orders of Didi Chuxing's users and verify the possibility of using these orders to predict departure time. Moreover, we propose such a model based on LightGBM using the users' current location, the order of the day of the week, and the date attribute. Finally, the experimental results indicate the superior performance of TP-LightGBM in predicting users' departure time. However, our model can only serve about 35% of orders. Thus, improving the prediction probability and expanding the service volume become the focus of our future research.

Acknowledgment. We are grateful to anonymous reviewers for their helpful comments. This work is partially supported by the National Key Research and Development Program of China under Grant No. 2018AAA0101100.

References

1. Van der Aalst, W.M., Schonenberg, M.H., Song, M.: Time prediction based on process mining. Inf. Syst. **36**(2), 450–475 (2011)
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Routledge, Milton Park (2017)
3. Chen, M., Chien, S.I.: Dynamic freeway travel-time prediction with probe vehicle data: link based versus path based. Transp. Res. Rec. **1768**(1), 157–161 (2001)
4. Chen, M., Liu, X., Xia, J., Chien, S.I.: A dynamic bus-arrival time prediction model based on APC data. Comput.-Aided Civil Infrastruct. Eng. **19**(5), 364–376 (2004)
5. Cheng, J., Li, G., Chen, X.: Research on travel time prediction model of freeway based on gradient boosting decision tree. IEEE Access **7**, 7466–7480 (2018)
6. Chien, S.I.J., Ding, Y., Wei, C.: Dynamic bus arrival time prediction with artificial neural networks. J. Transp. Eng. **128**(5), 429–438 (2002)
7. Chien, S.I.J., Kuchipudi, C.M.: Dynamic travel time prediction with real-time and historic data. J. Transp. Eng. **129**(6), 608–616 (2003)
8. Ding, C., Wang, D., Ma, X., Li, H.: Predicting short-term subway ridership and prioritizing its influential factors using gradient boosting decision trees. Sustainability **8**(11), 1100 (2016)
9. Duan, Y., Yisheng, L., Wang, F.Y.: Travel time prediction with LSTM neural network. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pp. 1053–1058. IEEE (2016)
10. Fei, X., Lu, C.C., Liu, K.: A Bayesian dynamic linear model approach for real-time short-term freeway travel time prediction. Transp. Res. Part C: Emerg. Technol. **19**(6), 1306–1318 (2011)
11. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. 1189–1232 (2001)
12. Ke, G., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
13. Kwon, J., Coifman, B., Bickel, P.: Day-to-day travel-time trends and travel-time prediction from loop-detector data. Transp. Res. Rec. **1717**(1), 120–129 (2000)
14. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
15. Quinlan, J.R.: C4. 5: programming for machine learning. Morgan Kauffmann **38**(48), 49 (1993)
16. Rao, H., et al.: Feature selection based on artificial bee colony and gradient boosting decision tree. Appl. Soft Comput. **74**, 634–642 (2019)
17. Sun, R., Wang, G., Zhang, W., Hsu, L.T., Ochieng, W.Y.: A gradient boosting decision tree based GPS signal reception classification algorithm. Appl. Soft Comput. **86**, 105942 (2020)
18. Xu, J., Rahmatizadeh, R., Bölöni, L., Turgut, D.: Real-time prediction of taxi demand using recurrent neural networks. IEEE Trans. Intell. Transp. Syst. **19**(8), 2572–2581 (2018)
19. Yang, L., Zhang, X., Liang, S., Yao, Y., Jia, K., Jia, A.: Estimating surface downward shortwave radiation over china based on the gradient boosting decision tree method. Remote Sens. **10**(2), 185 (2018)
20. Zhang, X., Rice, J.A.: Short-term travel time prediction. Trans. Res. Part C: Emerg. Technol. **11**(3–4), 187–210 (2003)
21. Zhang, Y., Haghani, A.: A gradient boosting method to improve travel time prediction. Transp. Res. Part C: Emerg. Technol. **58**, 308–324 (2015)

Radar and Sonar Networks

Accurate Contact-Free Material Recognition with Millimeter Wave and Machine Learning

Shuang He[1], Yuhang Qian[1], Huanle Zhang[1], Guoming Zhang[1], Minghui Xu[1], Lei Fu[2,3], Xiuzhen Cheng[1], Huan Wang[4], and Pengfei Hu[1](✉)

[1] Shandong University, Qingdao, Shandong, China
{heshuang,dtczhang,guomingzhang,mhxu,xzcheng,phu}@sdu.edu.cn
[2] Bank of Jiangsu, Nanjing, China
leileifu@163.sufe.edu.cn
[3] Fudan University, Shanghai, China
[4] Guangxi University of Science and Technology, Liuzhou, China
wanghuan@gxust.edu.cn

Abstract. Material recognition plays an essential role in areas including industry automation, medical applications, and smart homes. However, existing material recognition systems suffer from low accuracy, inconvenience (e.g., deliberate measuring procedures), or high cost (e.g., specialized instruments required). To tackle the above limitations, we propose a contact-free material recognition system using a millimetre wave (mmWave) radar. Our approach identifies materials such as metal, wood, and ceramic tile, according to their different electromagnetic and surface properties. Specifically, we leverage the following techniques to improve the system robustness and accuracy: (1) spatial information enhancement by exploiting multiple receiver antennas; (2) channel augmentation by applying Frequency Modulated Continuous Wave (FMCW) modulation; and (3) high classification accuracy enabled by Artificial Intelligence (AI) technology. We evaluate our system by applying it to classify five common materials. The experimental results are promising, with 98% classification accuracy, which shows the effectiveness of our mmWave-based material recognition system.

Keywords: Contact-free material recognition · Millimeter wave radar · Machine learning

1 Introduction

Recognizing materials have a wide range of applications, e.g., categorizing waste materials in industrial automation [3], detecting normal/cancerous cells in the

This work is supported by the National Key Research and Development Program of China (No. 2021YFB3100400), the Shandong Science Fund for Excellent Young Scholars (No. 2022HWYQ-038), the Guangxi Natural Science Foundation (No. 2020GXNSFBA159042).

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 609–620, 2022.
https://doi.org/10.1007/978-3-031-19214-2_51

medical field [8,29], and modeling environments in smart homes [15]. With the development of smart city, material recognition has become an imperative component for many intelligent devices. Compared to their contact-based counterparts, contact-free material recognition systems are gaining popularity because of their fewer physical constraints and better user experience.

There are several mainstream methods to build contact-free material recognition systems. (1) Near Infrared (NIR) spectroscopy. NIR spectroscopy is a method to detect the electromagnetic spectrum from 780 nm to 2500 nm wavelengths. It has been applied to recognize many organic materials [16]. However, NIR spectroscopy has many shortcomings, such as high cost and low accuracy. (2) Optical sensing technology (e.g., lidar) uses a light resistance with multi-spectral illumination to identify the surface materials [7], but its accuracy is severely affected by the visibility degree of objects. (3) Mechanical radars rely on signal factors such as distances and incident angles to classify materials [19]. However, such sensing technology is complex and expensive, and the hardware requirements are strict. None of those mentioned above methods provides affordable and accurate contact-free material recognition functionality.

In this paper, we propose a mmWave radar system to recognize materials. Our system has the same merits of robustness and versatility as the mechanical radars have, but ours does not have the problems of complex structures and challenging operational conditions faced by mechanical radars. Compared with other frequency bands, mmWave radar achieves a supreme performance regarding accuracy, cost, and size. Specifically, an mmWave radar system has the following strength:

1. *High Resolution.* mmWave radars have high resolutions because of their excellent signal beam-forming. For example, a 76–81 GHz radar's range resolution reaches the sub-millimetre level, and the angular resolution is as precise as 1° [29,30].
2. *Robustness.* When the visibility condition is poor, e.g., in rain and mist, the sensing performance of an mmWave radar is still robust. As a result, an mmWave system is capable of all-weather and all-time sensing.
3. *Lightweight.* Thanks to the development of microelectronic technology, mmWave radars are becoming miniature and low cost. Embedded devices and wearable devices are highly likely to incorporate mmWave radars to enable millimeter communications and sensing capability.

Although mmWave signals have incomparable advantages, realizing a practical and accurate mmWave radar system for material recognition entails careful considerations. This is because mmWave signals are susceptible to environments. Therefore, designing a robust and precise feature representation scheme for distinguishing materials is the core. To improve the material recognition accuracy, we leverage the following techniques:

1. Aggregation of multiple transmitter-to-receiver (Tx-to-Rx) paths. A typical mmWave hardware has multiple transmitter and receiver antennas. Each Tx-to-Rx pair captures different channel information. In light of it, we propose to incorporate multiple Tx-to-Rx pairs to exploit more spatial information.

2. Frequency-Modulated Continuous Wave (FMCW) modulation to measure the Received Signal Strength (RSS) of the signals reflected from a target material. Compared to a Continuous Wave (CW)-based radar, an FMCW radar enables more precise RSS profiles since it spans a frequency band.
3. A complete Machine Learning (ML) pipeline. Our system extracts useful features from the RSS profiles that are generated by multiple Tx-to-Rx pairs. Afterwards, it runs a powerful ML model to recognize the materials, which shows inspiring classification accuracy.

We evaluate our system by classifying five common building materials (copper, wood, acrylic, tile, and drywall), where an mmWave radar is placed at 40 cm from the materials. The evaluation results show the effectiveness of our system. In particular, our Convolutional Neural Network (CNN)-based pipeline achieves inspiring 98% classification accuracy.

This paper is organized in the following manner. First, we provide related works in Sect. 2. Then, we elaborate our system design in Sect. 3. Afterwards, we evaluate our system in Sect. 4. We discuss the limitations/opportunities in Sect. 5. Last, we conclude this paper in Sect. 6.

2 Related Works

This section provides related works in terms of contact-based material recognition and Radio Frequency (RF)-based contact-free material recognition.

2.1 Contact-Based Material Recognition

Several contact-based systems have been proposed to realize material recognition by utilizing physical-level features such as chemical properties [21], thermal properties [20], and optical properties [11]. Despite their industrial deployments, these solutions are task-oriented and have no mobility. Furthermore, they require to attach specified sensors to objects for recognition. In comparison, our system is a contact-free solution and thus is more flexible and user-friendly.

2.2 RF-Based Contact-Free Material Recognition

In addition to localization [1] and perception [5], signal reflection of RF waves can be used for material recognition. For example, RSA [31] determines curvature and surface material by measuring the reflected mmWave signals at multiple locations. RadarCat adopts a similar workflow but uses 60 GHz signals [28]. Yang et al. [26] investigate the feasibility of using 60 GHz millimeter-wave (mmWave) signal as a ubiquitous and non-invasive way to estimate the Soluble Sugar Content (SSC) in fruits. Beside, some other RF-based works consider more on magnetic properties (e.g., dielectric constant ϵ_r, losses $\tan_\delta$ [2,23]), which require expensive facility like vector network analyzer, with sophisticated calibrating procedures as beamforming with high-gain dielectric lenses or elliptical mirror.

In comparison, we only use one mmWave board with onboard transmitters and receivers. In addition, exiting millimeter works are vulnerable to the positions of transmitters and receivers, as they rely on sensitive phase information of signals. Compared to existing mmWave works, our system exploits ML learning that extracts features from RSS profiles, which is more robust against wavelength misalignment. Besides mmWave communications, other RF technologies such as Wi-Fi [6,13], UWB [4], and RFID [24] have been used for material classification. However, they have much lower classification accuracy in practice because of the long wavelengths and already congested frequency bands.

3 System Design

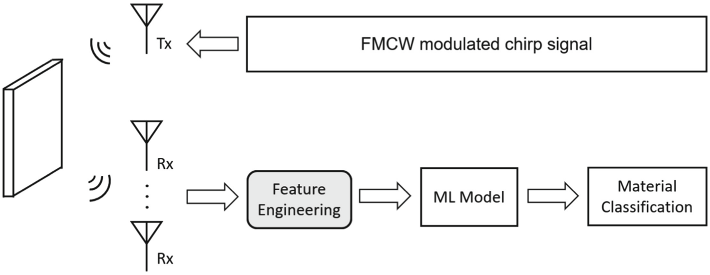


Fig. 1. The workflow of our system.

Our system recognizes materials based on the reflected mmWave signals. Figure 1 depicts the workflow of our system. An FMCW modulated chirp signal is emitted directly toward the target material for recognition, which signal is then reflected by the material and received by multiple receiver antennas. Afterward, we extract features from multiple receiver antennas (details in Sect. 3.3). Last, we adopt an ML model to classify the material.

3.1 Principle of mmWave Material Recognition

In a mono-static radar, the mmWave signal follows the propagation model [14]:

$$P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 d^4} \tag{1}$$

where P_r is the power of received signals, P_t is the transmit power, with G_t and G_r are the antenna gains for Tx and Rx respectively. λ is the wavelength transmitted in free space. Since mmWave has a short wavelength, it indicates that mmWave signals suffer severer attenuation than microwave signals. To compensate for signal attenuation, practical mmWave radars use Multiple Input Multiple Output (MIMO) antenna arrays to obtain high G_t and G_r gains.

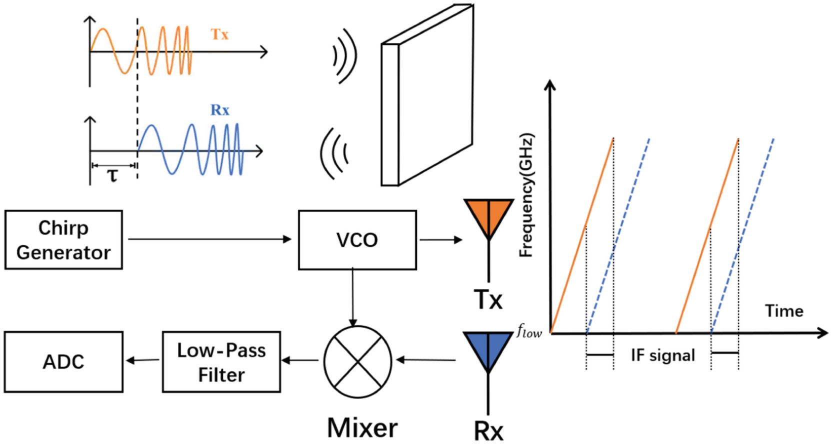


Fig. 2. Illustration of FMCW modulation.

σ is the Radar Cross Section (RCS), a metric to represent the size of an object that appears in the view of a radar. RCS can be regarded as the electromagnetic equivalent area of a target object, the area that intercepts the transmitter radar power and then scatters that power isotropically back to the radar receiver. The RCS area does not necessarily overlap with the physical area of an object. It is largely determined by the material reflectivity. For example, metal suffers a 0.6 dB RSS loss while that of wood is 12 dB. Therefore, when we measure objects in a homogeneous condition (e.g., shape, distance to the radar, radar configurations), we can leverage P_r (correspondingly σ) to classify their materials.

3.2 Channel Augmentation with FMCW Modulation

FMCW modulates signals in chirps—a sinusoidal wave signal in a linearly increasing frequency. FMCW is widely used for ranging. We adopt FMCW to augment channel information by changing the transmission frequency. Therefore, compared to a single-frequency Continuous Wave (CW) modulation, FMCW provides more detailed channel information and thus higher material recognition accuracy.

Figure 2 illustrates the FMCW modulation. For a monostatic radar, the Tx and Rx signal can be described with real numbers as

$$\begin{aligned} S_T(t) &= A_T \cdot \cos\left(2\pi \cdot f_T(t) \cdot t + \phi_T\right) \\ S_R(t) &= A_R \cdot \cos\left(2\pi \cdot f_R(t) \cdot t + \phi_R\right) \end{aligned} \tag{2}$$

where A_T and A_R are the amplitude of signal, $f_T(t)$ and $f_R(t)$ are the run-time frequency of signal at time t, ϕ_T and ϕ_R are the initial phrase of transmitted and received signal respectively. By multiplying $S_T(t)$ and $S_R(t)$, we obtain the Intermediate Frequency (IF) signal:

$$S_{IF}(t) = S_T(t) * S_R(t) \approx \frac{1}{2} A_T A_R * \cos \{[2\pi (f_T(t) - f_R(t))]\, t + (\phi_T - \phi_R)\} \quad (3)$$

where a low-pass filter is applied to remove the higher frequency.

3.3 Feature Engineering

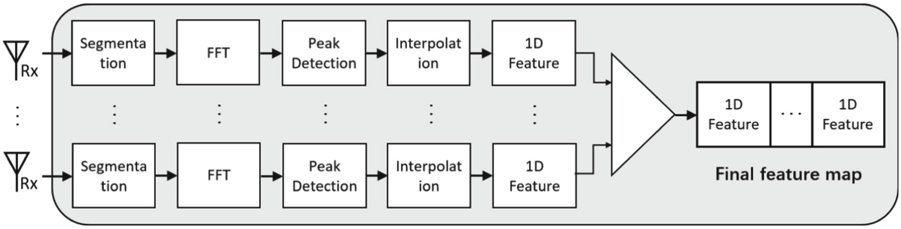


Fig. 3. The procedure of our feature engineering.

We extract features from $S_{IF}(t)$ and then apply an ML model to classify the materials based on the extracted features. Figure 3 illustrates our feature engineering procedure. Specifically, we design the following steps:

1. We segment the data stream chirp-wise, where each segment lasts 21 ms and has 64 Analogue to Digital Converter (ADC) samples. To avoid faraway RF clutters, the distance resolution and the detection coverage of our mmWave radar is set to 4 cm and 3 m, respectively.
2. We conduct a 64-point Fast Fourier transform (FFT) to calculate the frequency components of each segment. Before FFT calculation, we apply Hamming windows to mitigate the spectral leakage.
3. We identify the most informative region (7 data points) of the FFT spectrum by applying a Continuous Wavelet Transform (CWT)-based peak detection algorithm. We observe that these peak regions are representative of different materials. Therefore, instead of feeding the whole FFT spectrum to an ML model for classification, we only extract the region of the peak FFT spectrum, which is easier for the ML model to learn.
4. We extend the 7 data points from the peak spectrum to 13 points by three-point parabolic interpolation. This is because the frequency context from a single channel is coarse-grained, as the 4cm resolution is not precise enough. As a result, we extract a 1D feature of 13 numbers for each Tx-Rx pair.
5. We concatenate the 1D feature from each Tx-Rx pair into a longer 1D feature map [17]. For example, our mmWave radar has 4 receiver antennas and thus the final 1D feature has 52 (4 × 13) numbers. Since Rx antennas are separated about 2.5 mm apart, which is larger than the mmWave half wavelength (1.9 mm), the channel conditions captured by different Tx-Rx pair varies due to multi-path effect [27]. Therefore, by concatenating features from different Tx-Rx pairs, we obtain a more “panorama” view of the wireless channels and thus better classification accuracy.

3.4 Machine Learning Models

We apply two kinds of machine learning models to classify the materials based on the extracted feature maps. (1) Support Vector Machine (SVM), as it is one of the most well-defined supervised learning models [25]. (2) Convolutional Neural Network (CNN) [22]. We customize a 7-layer CNN as shown in Table 1. For the 1D convolutional layer, we set kernel size to 5 with stride to 1, as a smaller kernel is more perceived to edge information. The first three fully-connected layers have 64 neurons, and the forth fully-connected layer has 5 outputs (the number of materials in our experiments) [22]. Last, a softmax layer is appended and the material is classified to the output class with the highest probability.

Table 1. Our customized CNN model structure.

Layer	Type	Output shape
0	Input Layer	$(52, 1)$
1	Conv1D	$(48, 1)$
2	Conv1D	$(44, 1)$
3	Conv1D	$(40, 1)$
4	Fully-Connected Layer	(64)
5	Fully-Connected Layer	(64)
6	Fully-Connected Layer	(64)
7	Fully-Connected Layer	(5)

4 Evaluation

In this section, we first introduce the hardware setup of our system. Then, we explain our collected data set, which is used for the system evaluation. Last, we present our evaluation results.

4.1 Implementation

Figure 4 shows the experiment setup in a corridor. The material plate is mounted on a tripod. The mmWave radar is placed at 40 cm away from the target material plate, with signals transmitted directly to the material plate. In addition, we attach a vibrator motor to the bottom of the mmWave board tripod to mimic the hand-held case. The mmWave radar could be mounted on a mobile robot to search for an optimal distance between the material plate and the mmWave board, which we leave as future work.

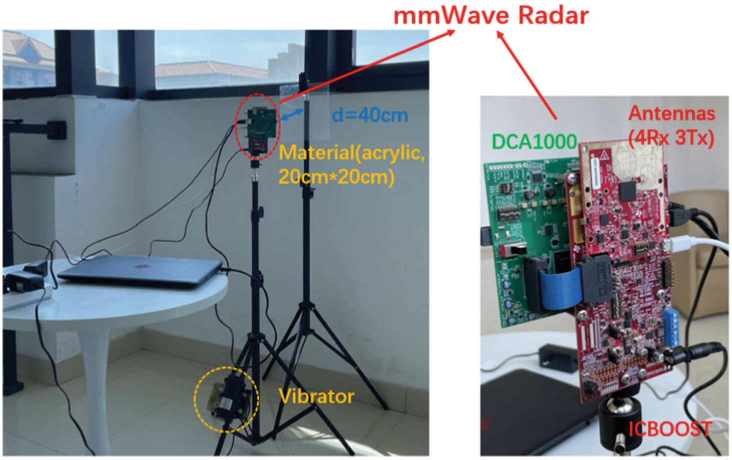


Fig. 4. Experiment setup

(a) Copper (b) Wood (c) Acrylic (d) Tile (e) Drywall

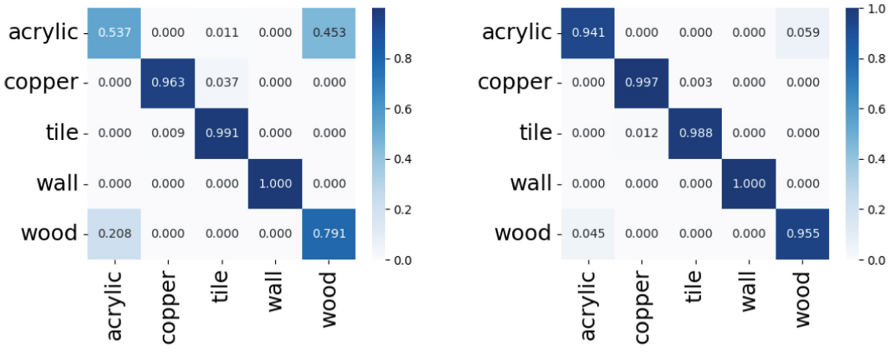
Fig. 5. Materials used for evaluation.

We use a TI IWR1642 Booster Pack that includes an evaluation board (IWR1642BOOST) and a real-time data-capture adapter (DCA1000EVM). The evaluation board has two Tx and four Rx antennas in the 76–81 GHz working frequency range. We use one Tx antenna to transmit the FMCW signal and all four Rx antennas to receive the reflected signal. The antenna chip is directly connected to a laptop (an Intel Core i7-10750H CPU and a 16 GB memory) through two Micro USB cables, and the DCA1000 data capturer is connected to it via an Ethernet RJ45 interface. ICBOOST is supported by a 5V/3A AC power supply adapter, and a 12V/2A adapter powers the vibrator. We use mmWave studio and Matlab for system configuration and data processing.

We select five most common building materials in our experiments. Figure 5 illustrates our chosen material plates, i.e., copper, wood, arylic, tile, and drywall. Each plate is square in shape, with 20 cm in length/width and 1mm thickness.

4.2 Data Collection and ML Training Configuration

We enable the vibrator to enforce a slight vibration during data collection. In our experiments, we collect raw ADC data stream by the data-capture adapter,



(a) SVM (b) CNN

Fig. 6. Confusion matrices of material classification using (a) SVM model and (b) our customized CNN model.

whose sampling rate is set to 3048 KHz. A range resolution up to 4 cm is obtained. In total, we collect 200K data samples from different locations, where each type of material has 40K data samples. Each data sample is an array of 13 floating numbers, and the peak value is located in the center. The total file size of our collected dataset is more than 2 GB.

For the SVM model, we use the default configurations in Matlab. Regarding the CNN model, we adopt the Adam optimizer [12], with a learning rate of 0.001. We apply 10-fold cross-validation and report the average results.

4.3 Evaluation Results

Figure 6 depict the confusion matrices of our mmWave-based material classification system using two different types of ML models. Overall, both models achieve good classification accuracy, as the diagonal cells are much darker than the non-diagonal cells. In particular, our CNN model obtains excellent performance in classifying these materials. In comparison, the SVM model is moderately confused about acrylic and wood. The results indicate that the CNN model structure is powerful for the mmWave-based material classification.

Table 2. SVM (left) and CNN (right) evaluation metrics.

	Accuracy	Precision	Recall	F_1
Acrylic	0.86	0.72	0.54	0.62
Copper		0.99	0.96	0.97
Tile		0.95	0.99	0.97
Wall		0.99	0.99	0.99
Wood		0.64	0.79	0.71

	Accuracy	Precision	Recall	F_1
Acrylic	0.98	0.94	0.95	0.94
Copper		0.99	0.98	0.98
Tile		0.98	0.99	0.98
Wall		1	1	1
Wood		0.95	0.94	0.94

Table 2 shows the details of the classification performance when the SVM model and the CNN model are applied, respectively. In addition to the average accuracy, we also report the precision, recall, and F_1 score for each type of material. On average, our CNN-based system achieves an inspiring 98% accuracy in classifying these five materials. In addition, our CNN-based system has almost perfect precision, recall, and F_1, with scores all higher than 0.94.

5 Discussion

Although our system achieves approximately perfect accuracy in classifying materials in our experiments, it has several limitations/opportunities worth further investigation.

Recognizing more Types and Forms of Materials. We select the five most common solid materials in buildings. The chosen materials are diverse, and thus our results are representative. Nonetheless, we plan to evaluate our system with more types of materials such as organic material, and even more forms of materials, including liquid and gas [4,10,18].

Support of Dynamic Number of Receiver Antennas. In our current implementation, we fix the number of receiver antennas. As a result, the size of the extracted features and the corresponding ML model are kept the same, which may not be appropriate for other mmWave hardware equipped with a different number of receiver antennas. Therefore, we will design an ML component that supports a dynamic number of antennas. Besides, we will study the classification accuracy versus the number of antennas in our future work.

Developing Finer Feature Engineering. We propose a feature engineering component, which extracts the region around the spectrum peak [9]. In our future work, we want to explore other feature extraction procedures. For example, instead of the 1D spectrum, we can also extract the 2D spectrogram features, which could contain more related information for material recognition.

Support of Less Controlled Experiment Settings. We want to evaluate our system in more dynamic and practical settings. For example, we currently fix the distance between the mmWave board and the material, showing excellent classification accuracy. In our future work, we want to relax this physical constraint so that users can place our mmWave board at various distances from the target material.

6 Conclusion

This paper presents an accurate contact-free material recognition system by leveraging millimeter wave communication and machine learning technology, which is verified efficient in the static indoor material experiment. mmWave has a better sensing capability than other RF technologies because the wavelength of mmWave is much shorter. To extract informative features from mmWave signals, we propose a unique feature engineering procedure that incorporates frequency domain operations. The extracted features along with our customized CNN model achieves 98% accuracy in classifying five building materials. Our material recognition system is promising considering its high accuracy, low cost, and small size, thanks to the mass-production of mmWave modules. We leave it as future work to design a mobile version of it.

References

1. Adib, F.M., Kabelac, Z., Katabi, D.: Multi-person localization via RF body reflections. In: Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation (NSDI), pp. 279–292 (2015)
2. Barowski, J., Zimmermanns, M., Rolfes, I.: Millimeter-wave characterization of dielectric materials using calibrated FMCW transceivers. IEEE Trans. Microw. Theory Tech. **66**(8), 3683–3689 (2018)
3. Cesetti, M., Nicolosi, P.: Waste processing: new near infrared technologies for material identification and selection. J. Instrum. **11**(9), C09002–C09002 (2016)
4. Dhekne, A., Gowda, M.K., Zhao, Y., Hassanieh, H., Choudhury, R.R.: LiquID: a wireless liquid identifier. In: Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys), pp. 442–454 (2018)
5. Ding, H., et al.: A platform for free-weight exercise monitoring with passive tags. IEEE Trans. Mob. Comput. **16**(12), 3279–3293 (2017)
6. Feng, C., et al.: WiMi: target material identification with commodity Wi-Fi devices. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 700–710 (2019)
7. Harrison, C., Hudson, S.E.: Lightweight material detection for placement-aware mobile computing. In: Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST), pp. 279–282 (2008)
8. Heunisch, S., Ohlsson, L., Wernersson, L.E.: Reflection of coherent millimeter-wave wavelets on dispersive materials: a study on porcine skin. IEEE Trans. Microw. Theory Tech. **66**(4), 2047–2054 (2018)
9. Holloway, C.L., Perini, P.L., Delyser, R.R., Allen, K.C.: Analysis of composite walls and their effects on short-path propagation modeling. IEEE Trans. Veh. Technol. **46**(3), 730–738 (2002)
10. Huang, Y., Chen, K., Wang, L., Dong, Y., Huang, Q., Wu, K.: Lili: liquor quality monitoring based on light signals. In: Proceedings of ACM MobiCom, pp. 256–268 (2021)
11. Iqbal, F., et al.: Alcohol sensing and classification using PCF-based sensor. Sens. Bio-sens. Res. **30**, 100384 (2020)
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv Learning (2014)
13. Koike-Akino, T., Wang, P., Pajovic, M., Sun, H., Orlik, P.V.: Fingerprinting-based indoor localization with commercial MMWave WiFi: a deep learning approach. IEEE Access **8**, 84879–84892 (2020)
14. Lin, S.K.: Microwave and millimeter-wave remote sensing for security applications. By Jeffrey A. Nanzer, Artech House, 2012; 372 pages. Remote. Sens. **5**, 367–373 (2013)
15. Lu, C.X., et al.: See through smoke: robust indoor mapping with low-cost MMWave radar. In: Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services (MobiSys), pp. 14–27 (2020)

16. Manley, M.: Near-infrared spectroscopy and hyperspectral imaging: non-destructive analysis of biological materials. Chem. Soc. Rev. **43**(24), 8200–8214 (2014)
17. das Neves Franco, L.A.P., Sinatora, A.: 3D surface parameters (ISO 25178-2): actual meaning of Spk and its relationship to Vmp. Precis. Eng. **40**, 106–111 (2015)
18. Rahman, T., Adams, A.T., Schein, P., Jain, A., Erickson, D., Choudhury, T.: Nutrilyzer: a mobile system for characterizing liquid food with photoacoustic effect. In: Proceedings of ACM SenSys, pp. 123–136 (2016)
19. Sagala, T.B.V., Suryana, J.: Implementation of mechanical scanning and signal processing for FMCW radar. In: 2016 International Symposium on Electronics and Smart Devices (ISESD), pp. 46–50 (2016)
20. Saponaro, P., Sorensen, S., Kolagunda, A., Kambhamettu, C.: Material classification with thermal imagery. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4649–4656 (2015)
21. Smidt, E., Schwanninger, M., Tintner, J., Böhm, K.: Ageing and deterioration of materials in the environment - application of multivariate data analysis. In: Multivariate Analysis in Management, Engineering and the Sciences (2013)
22. Stathakis, D.: How many hidden layers and nodes? Int. J. Remote Sens. **30**, 2133–2147 (2009)
23. Vakili, I., Ohlsson, L., Wernersson, L.E., Gustafsson, M.G.: Time-domain system for millimeter-wave material characterization. IEEE Trans. Microw. Theory Tech. **63**(9), 2915–2922 (2015)
24. Wang, J., Xiong, J., Chen, X., Jiang, H., Balan, R.K., Fang, D.: TagScan: simultaneous target imaging and material identification with commodity RFID devices. In: Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom), pp. 288–300 (2017)
25. Wu, T., Lin, C.J., Weng, R.C.H.: Probability estimates for multi-class classification by pairwise coupling. J. Mach. Learn. Res. **5**, 975–1005 (2003)
26. Yang, Z., et al.: On the feasibility of estimating soluble sugar content using millimeter-wave. In: Proceedings of ACM/IEEE IoTDI, pp. 13–24 (2019)
27. Yanik, M.E., Torlak, M.: Near-field MIMO-SAR millimeter-wave imaging with sparsely sampled aperture data. IEEE Access **7**, 31801–31819 (2019)
28. Yeo, H.S., Flamich, G., Schrempf, P., Harris-Birtill, D., Quigley, A.J.: Radarcat: radar categorization for input interaction. In: Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST), pp. 833–841 (2016)
29. Zhang, R., Cao, S.: Extending reliability of mmWave radar tracking and detection via fusion with camera. IEEE Access **7**, 137065–137079 (2019)
30. Zhao, M., et al.: RF-based 3D skeletons. In: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM), pp. 267–281 (2018)
31. Zhu, Y., Zhu, Y., Zhao, B.Y., Zheng, H.: Reusing 60ghz radios for mobile radar imaging. In: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom), pp. 103–116 (2015)

Subcarrier Index Modulation Aided Non-Coherent Chaotic Communication System for Underwater Acoustic Communications

Deqing Wang[1,2], Minghang You[1,2], Weikai Xu[1,2](✉), and Lin Wang[2]

[1] Key Laboratory of Underwater Acoustic Communication and Marine Information Technology (Xiamen University), Ministry of Education,
Xiamen, People's Republic of China

[2] Department of Information and Communication Engineering, Xiamen University,
Xiamen, People's Republic of China

xweikai@xmu.edu.cn

Abstract. In this paper, a subcarrier index modulation aided code-shifted differential chaos shift keying (SIM-CS-DCSK) system based on orthogonal frequency division multiplexing (OFDM) is proposed. In the proposed system, the transmitted bits are divided into two parts, where one part is used for conventional CS-DCSK modulation and the other part, served as subcarrier mapping bits, is used for the subcarrier index modulation. Benefiting from index modulation, SIM-CS-DCSK can achieve higher data rate compared to conventional multicarrier CS-DCSK system. Numerical simulations indicate that SIM-CS-DCSK has good BER performance over the time and frequency selective fading channels. Specifically, the proposed system outperforms the multicarrier spread-spectrum system (MCSS) by 0.5 dB to 2 dB. Real-field experiments in water pool and lake also confirm the superiority of the proposed system.

Keywords: Code-shifted differential chaos shift keying (CS-DCSK) · Orthogonal frequency division multiplexing (OFDM) · Subcarrier index modulation (SIM) · Underwater acoustic (UWA) communication

1 Introduction

Underwater acoustic (UWA) communications has been continuously studied over the past two decades. It has been widely applied in military and civil affairs, such as submarine communication, oil exploration, disaster warning, and so forth [1]. Specially, benefiting from the low probability of detection (LPD), underwater covert communications has attracted growing attention in scientific and engineering communities [2]. The reasons are twofold: (i) the transmitted signal with low transmission power is hidden behind background noise, and therefore the eavesdropper cannot detect the transmitted signal; (ii) the low-power signal can also reduce the interference for the marine organisms. Therefore, it is important to study and develop the underwater covert communications.

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 621–634, 2022.
https://doi.org/10.1007/978-3-031-19214-2_52

However, the UWA channel is considered to be the most complex channel, where the signal distortion is not only relative to the transmission distance but also the signal frequency [3]. It is noted that the bandwidth for the UWA communications is quite small. For example, when the transmission distance is more than 20 km, the available bandwidth is less than 10 kHz [4]. The UWA channel is a typical doubly selective channel. On the one hand, the speed of underwater sound is 1500 m/s and the maximum multipath delay is at least 10 ms in magnitude [5], thereby resulting in the frequency selective fading. On the other hand, the coherent time of the UWA channel is several seconds [6], and therefore the transmitted signal suffers from the time selective fading. Consequently, it is a challenging work to develop reliable, high-data-rate and low-complexity underwater covert communications.

The direct sequence spread-spectrum (DSSS) technology is widely used in the underwater covert communications. DSSS uses the pseudo-random sequence, such as m-sequence, to spread the information bits, therefore reducing the power spectrum density of the transmitted signal. The demodulation methods for DSSS can be categorized into coherent and non-coherent detection. The coherent detection needs the channel estimation and its complexity is high [7]. For example, a compressive sensing aided channel estimation was proposed in [8] to demodulate the information bits. Moreover, the authors of [9] proposed a joint channel estimation and interference cancellation technology to retrieve the transmitted bits. The non-coherent detection has lower complexity than the coherent one. For example, the double differential demodulation is used in [10], where the experimental results show that such a method can achieve higher data rate. Furthermore, a low-complexity match filter aided demodulation was proposed in [11], where multiple spreading sequences are transmitted in a parallel manner. In [12], the authors designed a bio-friendly covert communication system, and the filed experiment performed in the North Sea shows the transmission distance can reach to 10 km.

Different from the m-sequence, chaotic sequence is aperiodic and sensitive to the initial value. Exploiting the chaotic sequence to spread information bits, chaotic communications show good resistance against the multipath fading. The digital chaos shift keying (CSK) proposed in [13] needs the chaos synchronization, which increases the system complexity. Different from CSK, different CSK (DCSK) uses the differential demodulation, where the chaos synchronization and channel estimation are avoided [14]. The experiment performed in [15] shows that the chaotic-sequence spread-spectrum system can achieve good performance over the UWA channel. A low-complexity differential demodulation method was proposed in [16]. Motivated by [16], the authors of [17] proposed a time reversal mirror demodulation method. The properties of chaotic signals and the feasibility of chaotic signal over the UWA channel were studied in [18].

Although DCSK is capable of achieving good BER performance over the multipath fading channel, the energy efficiency and data rate of DCSK are low. This is because half of the DCSK symbol is used to transmit the non-information-bearing reference signal. Moreover, the requirement for the delay lines in DCSK imposes a burden on system complexity in practical applications. To avoid the use of delay lines, a code-shifted DCSK (CS-DCSK) system was proposed in [19], where the reference and information-bearing signals are superimposed in the same time slot. To increase the date rate, the authors of [20] integrated the orthogonal frequency division multiplexing (OFDM) into CS-DCSK, and the resultant system is referred to as multicarrier CS-DCSK (MC-CS-DCSK). It is shown in [20] that MC-CS-DCSK has good BER performance over the time-frequency doubly selective UWA channel.

Index modulation (IM) can provide new dimensions for the transmission of addition information bits. Particularly, additional information bits are transmitted by the ON-OFF state of different entities (such as antennas, sucbarriers, and so on) without requiring extra energy, and therefore IM-based systems can achieve a better energy efficiency [22]. In [23], an IM-based OFDM system was proposed for UWA communications. This system can not only achive higher energy efficiency, but also better peak to average power ratio (PAPR) performance. In order to improve the date rate of DSSS system, a parallel combination assisted UWA communication system was proposed in [21].

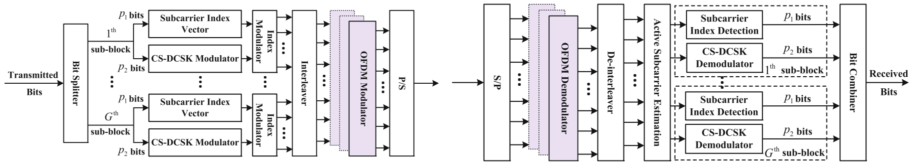
In this paper, a subcarrier index modulation aided CS-DCSK (SIM-CS-DCSK) system is proposed for the UWA communications, where additional information bits are transmitted by the indices of activated subcarries, improving the data rate. Different from MC-CS-DCSK, only part of subcarriers is activated in the proposed SIM-CS-DCSK system, which reduces the inter-carrier interference (ICI). Different from the conventional multicarrier spread-spectrum (MCSS) [24], SIM-CS-DCSK can achieve higher data rate and energy efficiency at the cost of system complexity. In order to verify the superiority of SIM-CS-DCSK, the BER performance of SIM-CS-DCSK is compared to its competitors over the different UWA channels. Simulation results show SIM-CS-DCSK not only has the ability to resist Doppler spread, but also harvests the benefits in time diversity. It is also shown that the proposed SIM-CS-DCSK system has a good trade-off between the data rate and BER performance.

2 The SIM-CS-DCSK System

2.1 The Transmitter

The block diagram of the SIM-CS-DCSK transmitter is shown in Fig. 1(a). There are m information bits transmitted in an SIM-CS-DCSK frame, where m bits are split into G sub-blocks with p bits per sub-block, i.e., $m = pG$. In each block, the number of transmitted bits is $p = p_1 + p_2$, where the first p_1 bits (referred to as subcarrier index bits) are utilized to determine the positions of the activated subcarriers for an OFDM symbol, while the remaining p_2 bits (referred to as modulated bits) are loaded into the CS-DCSK modulators to get the subcarrier modulated symbol.

In an OFDM symbol, there are N subcarriers divided into G sub-blocks with n subcarriers per sub-block. According to the combinational method [23], in each subcarrier sub-block, n_a out of n available subcarriers are activated to transmit



(a) Transmiter

(b) Receiver

Fig. 1. The block diagram of the SIM-CS-DCSK system.

the CS-DCSK signals, and the remaining $n - n_a$ subcarriers are idle. Therefore, the number of subcarrier index bits is calculated as $p_1 = \lfloor \log_2 C(n, n_a) \rfloor$, where $C(\cdot, \cdot)$ denotes the binomial coefficient and $\lfloor \cdot \rfloor$ is the floor function.

According to [19], the reference signal and information-bearing signal are superimposed to form a CS-DCSK signal, formulated as

$$\mathbf{d}_g^j = \mathbf{w}_R \otimes \mathbf{c} + a_g[j]\mathbf{w}_I \otimes \mathbf{c}, \tag{1}$$

where $\mathbf{w}_R$ and $\mathbf{w}_I$ denote the length-P reference and information-bearing Walsh codes, respectively. In addition, $\mathbf{c} = [c_1, c_2, \cdots, c_\beta]^T$ is the length-β chaotic signal, $a_g[j] \in \{-1, +1\}, j \in 1, 2, \ldots, n_a, g \in 1, 2, \ldots, G$ is the binary phase shift keying modulated symbol, and $\otimes$ is the Kronecker operator. In (1), $\mathbf{d}_g^j$ is a length-$P\beta$ signal transmitted by the j^{th} activated subcarrier of the g^{th} sub-block. The spreading factor of SIM-CS-DCSK is defined as $SF = P\beta$.

Generally, the subcarrier index vector of the g^{th} sub-block is defined as $\mathbf{I}_g = \{i_1, \cdots, i_\theta, \cdots, i_{n_a}\}$, where $g \in [1, 2, \cdots, G]$ and $i_\theta \in [1, 2, \cdots, n]$. Therefore, the g^{th} SIM-CS-DCSK symbol in the sub-block can be obtained as

$$\mathbf{S}_g = [\tilde{\mathbf{S}}(1), \tilde{\mathbf{S}}(2), \cdots, \tilde{\mathbf{S}}(n)]^T, g \in [1, 2, \cdots, G] \tag{2}$$

where $[\cdot]^T$ is the transpose operation and $\tilde{\mathbf{S}}(j)$ is given by

$$\tilde{\mathbf{S}}(j) = \begin{cases} 0, & j \notin \mathbf{I}_g \\ \mathbf{d}_g^j, & j \in \mathbf{I}_g \end{cases}, j \in [1, 2, \cdots, n]. \tag{3}$$

When the G sub-blocks of the SIM-CS-DCSK symbol are connected together to form

$$\mathbf{S} = [\mathbf{S}_1^T, \cdots, \mathbf{S}_G^T]^T. \tag{4}$$

To enhance the ability of anting frequency selective fading, a chip-based interleaver is used for the SIM-CS-DCSK system. Specifically, a circular shift interleaving operation is applied for each column of $\mathbf{S}$, expressed as

$$\mathbf{S}'(j) = \mathbf{Q}_{[j]}\mathbf{S}(j), j = 1, \cdots, P\beta, \tag{5}$$

where $\mathbf{Q}_{[j]}$ is a circular shift matrix used in the j^{th} OFDM symbol.

The duration of an OFDM symbol is T and the length of cyclic prefix (CP) is T_{cp}, respectively. Thus, the duration of an OFDM symbol is $T + T_{cp}$ and the subcarrier spacing is $\Delta f = 1/T$. Furthermore, the frequency of the k^{th} subcarrier is

$$f_k = f_c + k\Delta f, \qquad k = -N/2, \cdots, N/2 - -1, \tag{6}$$

where f_c is the carrier frequency, N is the number of subcarriers, and the bandwidth is $B = N\Delta f$. In (5), matrix $\mathbf{S}'$ has N rows and $P\beta$ columns. It is assumed that the j^{th} column of matrix $\mathbf{S}'$ is denoted by $\mathbf{x}_j$ which corresponds to the j^{th} OFDM symbol in a frame. For simplification, the subscript of $\mathbf{x}_j$ is omitted in the sequel. Therefore, the transmitted pass-band signal is given by

$$x(t) = \mathrm{Re}\left\{ \left[\sum_{k=1}^{N} x[k] e^{j2\pi k\Delta f t} g(t) \right] e^{j2\pi f_c t} \right\}, \tag{7}$$

where $x[k]$ is the k^{th} element of $\mathbf{x}_j$ and $g(t) = 1, t \in [-T_{cp}, T]$. After performing the parallel-to-serial (P/S) and digital-to-analog conversions, the transmitted signal $x(t)$ is sent through a UWA channel.

2.2 The Receiver

The block diagram of the SIM-CS-DCSK receiver is shown in Fig. 1(b). The received signal in passband is

$$y(t) = \mathrm{Re}\left\{ \sum_{l=1}^{L} A_l \left[\sum_{k=1}^{N} x[k] e^{j2\pi f_k (t+at-\tau_l)} \right] e^{j2\pi f_c (t+at-\tau_l)} \right\} + n(t), \tag{8}$$

where $n(t)$ is additive noise. After Doppler mitigating, downshifting, and low pass filtering, the baseband signal is approximated by

$$\hat{y}(t) \approx e^{j2\pi\epsilon t} \sum_{k=1}^{N} \left\{ x[k] e^{j2\pi f_k t} \left[\sum_{l=1}^{L} A_l e^{-j2\pi f_k \tau_l} g(t-\tau_l) \right] \right\} + \hat{n}(t), \tag{9}$$

where $e^{j2\pi\epsilon t}$ can be regarded as the carrier frequency offset (CFO) resulted by residual Doppler effect. Remove CP from $\hat{y}(t)$, the received signal becomes

$$z(t) = e^{j2\pi\epsilon t} \sum_{k=1}^{N} \left\{ x[k] e^{j2\pi f_k t} \left[\sum_{l=1}^{L} A_l e^{-j2\pi f_k \tau_l} \right] \right\} + \hat{n}(t). \tag{10}$$

Considering that the SIM-CS-DCSK system is an non-coherent system, the CFO is not solved. Thus, the output at the k^{th} subcarrier after demodulation is expressed by

$$z[k] = \frac{1}{T} \int_0^T z(t) e^{-j2\pi f_k t} \mathrm{d}t \approx \sum_{m=-N/2}^{N/2--1} H[m,k] x[k] + v[k], \tag{11}$$

where $v[k]$ is the circularly symmetric independent and identically distributed (i.i.d.) complex additional white Gaussian noise (AWGN) with zero mean and

σ^2 variance. Put (11) into a vector form, it yields the output of the m^{th} OFDM symbol

$$\mathbf{z}^m = \mathbf{H}^m \mathbf{x}^m + \mathbf{v}^m, m \in [1, 2, \cdots, \beta P], \tag{12}$$

where $\mathbf{z}^m = [z^m[-N/2], \cdots, z^m[N/2-1]]^T$, and $\mathbf{v}^m = [w^m[-N/2], \cdots, w^m[N/2-1]]^T$. The channel matrix is obtained by [25]

$$H^m(p;q) = \frac{1}{N} \sum_{n=1}^{N} \sum_{l=1}^{L} h(n;l) \exp\left\{ j2\pi \frac{n(q-p)-ql}{N} \right\} \tag{13}$$

where $h(n;l)$ is the discrete form of the time varying channel impulse response.

After performing the de-interleaving operation, the signal carried by an OFDM frame is expressed as

$$\mathbf{Z} = [\mathbf{Q}_{[1]}^T \mathbf{z}^1, \cdots, \mathbf{Q}_{[P\beta]}^T \mathbf{z}^{P\beta}]^T, \tag{14}$$

where $\mathbf{Q}_{[m]}^T$ is a circular shift matrix for de-interleaving of the m^{th} OFDM symbol.

The receiver needs to estimate the indices of the activated subcarriers and retrieve the subcarrier index bits. Then, according to the estimated subcarrier indices, the modulated bits carried by the activated subcarriers are recovered. In this paper, a low-complexity greedy detector is presented to detect the active subcarriers. The g^{th} sub-block of an OFDM frame is given by

$$\mathbf{R}_g = [\mathbf{Z}_{(g-1)n+1}^T, \mathbf{Z}_{(g-1)n+2}^T, \cdots, \mathbf{Z}_{gn}^T]^T. \tag{15}$$

Here, $\mathbf{Z}_m$ presents m^{th} row of matrix $\mathbf{Z}$. Thus, $\mathbf{R}_g$ is a matrix with n rows and $P\beta$ columns. To detect indices of the active subcarriers, the n received subcarriers of each sub-block are processed by the CS-DCSK demodulator. According to the demodulation principle of CS-DCSK, the decision variable of the CS-DCSK demodulator for the g^{th} sub-block can be calculated as

$$\mathbf{D}_g = \mathrm{diag}([\mathbf{R}_{\mathrm{g}} \odot (\mathbf{w}_{\mathrm{R}} \otimes \mathbf{1}_{\mathrm{n}\times\beta})] \, [\mathbf{R}_{\mathrm{g}} \odot (\mathbf{w}_{\mathrm{I}} \otimes \mathbf{1}_{\beta\times 1})]^{\mathrm{H}}), \tag{16}$$

where $\odot$ denotes the Hadamard product operation, $[\cdot]^H$ is the Hermitian transpose, and $\mathrm{diag}(\mathbf{A})$ is the function that takes the diagonal elements of matrix $\mathbf{A}$.

According to the greedy detection algorithm, the subcarriers corresponding to the n_a largest absolute values of decision variables are estimated as the indices of the activated subcarriers. Let $|\mathbf{D}_g| = \{|D_g[1]|, |D_g[2]|, \cdots, |D_g[n]|\}$. The subcarrier index vector can be estimated by **Algorithm 1**. Therefore, the subcarrier index bits are retrieved by the elements of subcarrier index vector, i.e., $\hat{\mathbf{I}}_g(l) = [\hat{I}_g[n_a], \cdots, \hat{I}_g[1]]$. Finally, the modulated bits carried by the activated subcarriers are obtained as

$$\hat{b}_g[j] = \mathrm{sign}\left[D_g\left[j\right]\right], j \in \hat{\mathbf{I}}_g. \tag{17}$$

Algorithm 1. Estimate Subcarrier Indices $\{\hat{I}_g[q]\}_{q=1}^{n_a}$

```
**Initialize:** Let $i = 1, \Xi = |\mathbf{D}_g|$.

**repeat**
    $\nu_i = \arg \max_{i=1,\cdots,n} (\Xi)$

    Set $\Xi_{\nu_i} = 0$

    $i = i + 1$

**until** $i == n_a$.
Output: $[\hat{I}_g[n_a], \cdots, \hat{I}_g[1]] = \mathrm{sort}([\nu_1, \cdots, \nu_{n_a}])$.
```

3 Numerical Simulations

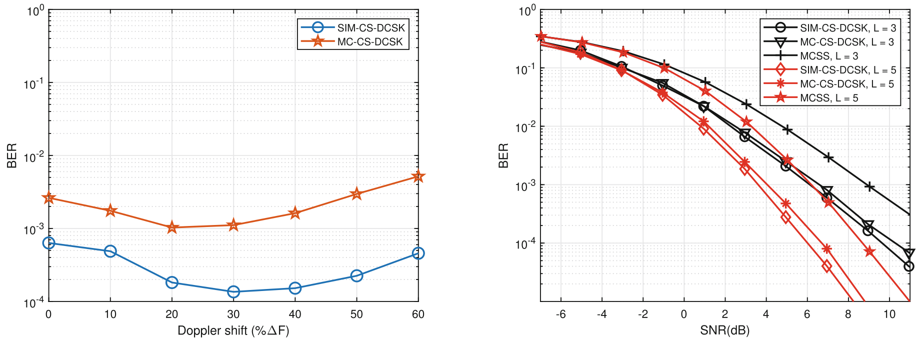
In this section, Monte Carlo simulations are performed to evaluate the BER performance of the proposed SIM-CS-DCSK system. Moreover, the BER performance of SIM-CS-DCSK is compared to other spread-spectrum systems, such as MCSS, SIM-MCSS[1] and MC-CS-DCSK. It is noted that the MCSS and SIM-MCSS systems are based on coherent demodulation, where the channel impulse response is estimated by the least square (LS) algorithm. Unless otherwise stated, the length of the Walsh code is set to $P = 2$ in SIM-CS-DCSK and MC-CS-DCSK. To demonstrate the system reliability under the low signal-to-noise ratio (SNR) region, the BER performance of all systems is evaluated based on the chip SNR, where the chip SNR E_{chip}/N_0 is given by

$$(E_{chip}/N_0)_{\mathrm{dB}} = (E_b/N_0)_{\mathrm{dB}} - 10 \log_{10}(SF)_{\mathrm{dB}}, \tag{18}$$

where E_{chip} and E_b are the chip energy of the chaotic signal and the energy of transmitting a information bit, respectively. In addition, N_0 is the power of a complex AWGN signal.

Figure 2(a) shows the BER performance of SIM-CS-DCSK over a multipath fading channel with different Doppler shift f_d. The channel is a five path fading channel with equal path gain. As observed from Fig. 2(a), the BERs of SIM-CS-DCSK and MC-CS-DCSK don't deteriorate as increasing of Doppler shift. For MC-CS-DCSK, when Doppler shift (f_d) is half of sub-carrier bandwidth ($\varDelta F$), it still achieves almost the same BER with zero Doppler shift. Moreover, SIM-CS-DCSK reach lower BER level at $f_d = \%50\varDelta F$ than that of $f_d = 0$. The performance of conventional OFDM system dose not deteriorate if an ICI equalizer was applied (see Fig. 2 in [26]). In contrast, there is no channel and ICI equalizers in the proposed system. Within a certain Doppler shift range, our proposed chaotic schemes achieve lower BER performance. It shows that MC-CS-DCSK and SIM-CS-DCSK system harvest the benefits of Doppler diversity.

[1] When the subcarrier index modulation is applied in MCSS [24], the resultant system is referred to as SIM-MCSS.



(a) BER versus f_d

(b) BER versus L

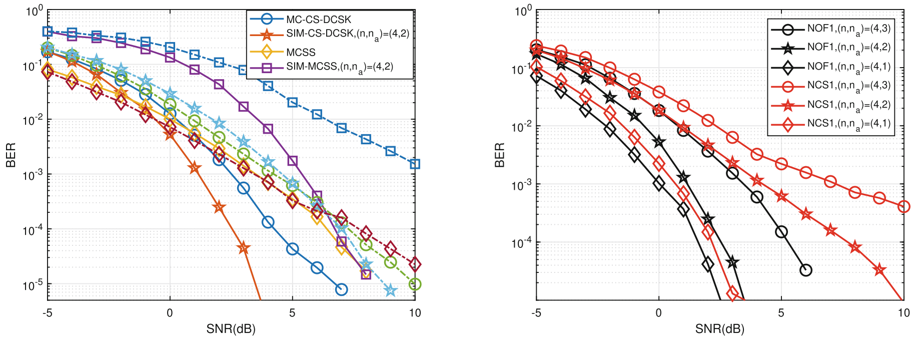
Fig. 2. BER performances of SIM-CS-DCSK and other competitors over a multipath fading channel. (a) BER versus f_d, $N = 512, E_b/N_0 = 17dB$, $(n, n_a) = (4, 2)$. (b) BER versus L, $N = 512, (n, n_a) = (4, 2)$.

In addition, from $f_d = 20\% \Delta F$ to $f_d = 50\% \Delta F$, SIM-CS-DCSK obtains big performance gains over $f_d = 0$ than that of MC-CS-DCSK. It shows that the SIM-CS-DCSK has better ani-ICI performance since SIM-CS-DCSK has inactive subcarriers.

In Fig. 2(b), the BER performance of the proposed SIM-CS-DCSK system is compared to its competitors over the three-path and five-path Rayleigh fading channels. When the number of paths is increased, both SIM-CS-DCSK and its competitors can offer better BER performances. It shows that the proposed SIM-CS-DCSK also achieve diversity gain. In addition, for $L = 5$ and BER $= 10^{-4}$, the proposed SIM-CS-DCSK system outperforms MC-CS-DCSK and MCSS by about 1 dB and 3 dB gains, respectively.

We use the Watermark simulator to evaluate the BER performance of SIM-CS-DCSK. Watermark is a recently proposed benchmark for comparing the performance of physical layer algorithms for underwater acoustic communications [27]. This channel simulator is driven by channel measurements, thereby producing different channel models. It includes a library of channels measured in Norway, France, and Hawaii, offering three frequency bands (i.e., $4 - 8$ kHz, $10 - 18$ kHz, and $32.5 - 37.5$ kHz), single-hydrophone and array receivers, and play time varying from 33 s to 1980 s [27].

We show the performance on two channel environments, NOF1 and NCS1, corresponding to a low Doppler spread and high Doppler spread channel in the Norwegian shallow water and continental shelf, respectively. The Doppler spectrum of NOF1 has a sharp peak around zero frequency while the Doppler spectrum of NCS1 is significantly spread out within [-15, 15] Hz. It is clear that the largest energy of the NOF1 channel is concentrated in a narrow delay-Doppler window. In contrast, NCS1 has wider Doppler spectrum. Thus, NOF1 is considered to be a benign channel, the NCS1 channel is more challenging due to its smaller coherence time.



(a) BER performance

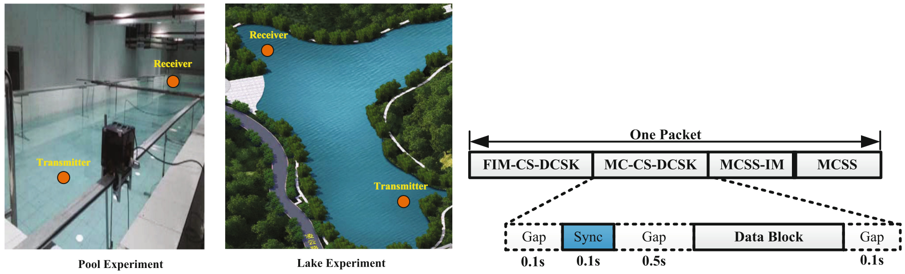
(b) BER versus n_a

Fig. 3. (a) BER performance of SIM-CS-DCSK over the watermark channel NOF1 and NCS1. (b) BER versus n_a, $N = 512$.

In Fig. 3(a), the BER performance of SIM-CS-DCSK is compared to that of MC-CS-DCSK, MCSS and SIM-MCSS over the watermark channel NOF1 and NCS1. Real lines represent BER over NOF1 channel, dot lines represent BER over NCS1 channel. In the simulations, the carrier frequency is set to $f_c =$ 14 kHz while the bandwidth is $B = 8$ kHz. The system parameters are $N = 512$, $(n, n_a) = (4, 2)$ and $SF = 32$. Therefore, the number of transmitted bits per SIM-CS-DCSK frame is 512. In addition, the numbers of transmitted bits per MCSS and SIM-MCSS frame are 409 and 408, respectively. It can be observed from Fig. 3(a) that SIM-CS-DCSK achieves best performance among MC-CS-DCSK, MCSS and SIM-MCSS over the NOF1 channel. Considering a more challenging NCS1 channel, SIM-CS-DCSK, MC-CS-DCSK offer better BER performance than MCSS and SIM-MCSS at high SNR range. Meanwhile, SIM-CS-DCSK achieves higher data rate.

Figure 3(b) evaluates the effect of the number of activated subcarriers in each sub-block on BER performance in SIM-CS-DCSK system. From the figure, one see that BER performance is decreased when the number of activated subcarriers n_a is decreased from 4 to 1. This is because the bit energy decreases as the number of activated sub-carriers decreases. However, the data rate of the SIM-CS-DCSK system is decreased with decreasing of n_a. Therefore, there is a good tradeoff between the BER performance and data rate in SIM-CS-DCSK.

4 Field Experiments



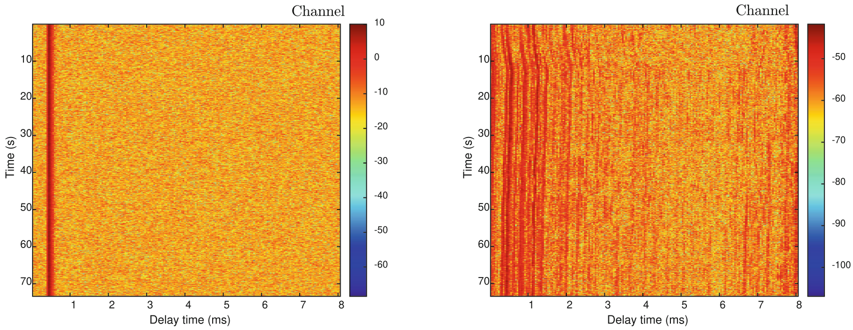
(a) Experiment settings (b) Frame structure

Fig. 4. The snapshots of experiment settings and frame structure of the transmitted signal for the field experiments.

In this section, the field experiments conducted in the water pool and lake are used to demonstrate the superiority of SIM-CS-DCSK. The snapshots of pool and lake experiments are shown in Fig. 4(a). The pool experiment is carried out in a $22.89 \times 5.18 \times 1.6$ m^3 non-anechoic water pool, where the transmitter and receiver are diagonally placed and their distance is 22 m. The lake experiment is conducted in Siyuan Lake of Xiamen University. The depth of the lake is about 3 m, and the transmission range is about 150 m.

As shown in Fig. 4(b), to ensure that the transmitted signals of different systems have the same channel impulse response, the transmitted signals of SIM-CS-DCSK, MC-CS-DCSK, SIM-MCSS and MCSS are concatenated into a packet. In addition, the guard interval, synchronization signal and data block are involved in each transmitted signal, where an linear frequency modulated (LFM) signal with the duration of 0.1 s is regarded as the synchronization signal to synchronize different signals.

The parameters of SIM-CS-DCSK and other comparison systems used in this study are as follows. Center frequency and bandwidth are 25 kHz and 6 kHz, respectively. The number of subcarrier and subcarrier spacing are 512 and $\Delta F =$ 11.72 Hz, respectively. Cyclic prefix (CP) is 19.94 ms. For fairness, spreading factor of all systems is set to 32. For MCSS and SIM-MCSS systems, a pilot signal inserts every 4 sub-carriers in the form of comb type pilot to estimate the channel. Considering the trade-off between data rate and BER performance, the SIM parameters of SIM-CS-DCSK and SIM-MCSS are set to $(n, n_a) = (4, 2)$.



(a) Channel impulse responses of the water pool channel.

(b) Channel impulse responses of the lake channel.

Fig. 5. Channel characteristics of the water pool and lake channels.

4.1 The Pool Experiment

Due to its calm characteristics, the pool channel can be regarded as a time-invariant multipath UWA channel. Channel impulse responses of water pool channel are illustrated in Fig. 5(a). From this figure, the energy mainly focuses on a delay path whose delay is less than 1 ms. It is a channel with slight frequency selective fading. In addition, the Doppler spectrum of the channel has a sharp peak around zero frequency. Hence, this water pool channel is a benign channel. Table 1 shows the BERs with different estimated input-SNRs. Specifically, the signal of data 1 has highest input-SNR, leading to lowest BERs for all schemes. More importantly, SIM-CS-DCSK achieves the lowest BER among the four schemes.

Table 1. BER performance comparisons of four systems in the water pool experiment.

Data	Estimated SNR(dB)	SIM-CS-DCSK	MC-CS-DCSK	SIM-MCSS	MCSS
1	6.57	0.0000	0.0206	0.0569	0.0199
2	6.50	0.0017	0.0349	0.0851	0.0313
3	5.59	0.0016	0.0426	0.0944	0.0353
4	3.79	0.0100	0.0638	0.1715	0.0572
Data rate		$179bps$	$149bps$	$179bps$	$149bps$

4.2 The Lake Experiment

For lake experiment, we also collected four experiment data. Channel impulse responses of lake channel are depicted in Fig. 5(b). From the figure, the lake channel shows lager delay spread than that of water pool channels. Besides, we can see that the Doppler effect of lake channel is significant. The channel shows relatively obvious time-varying property due to the fluctuation of the lake water. Table 2 shows the BER performances for the lake experiment. In the table, estimated input-SNRs are also listed. The lake data show lower input-SNRs than pool data due to long communication distance for lake experiment. From the BER results, one can see that the proposed SIM-CS-DCSK and MC-CS-DCSK display an obviously advantage versus MCSS and SIM-MCSS. For example, the BER of SIM-CS-DCSK reaches to 3% level when the estimated SNRs are in range $[-4, -1.6]$ dB.

Table 2. BER performance comparisons of four systems in the lake experiment.

Data	Estimated SNR(dB)	SIM-CS-DCSK	MC-CS-DCSK	SIM-MCSS	MCSS
1	-1.63	0.0000	0.0000	0.0087	0.1664
2	-1.71	0.0472	0.0699	0.3215	0.1764
3	-4.05	0.0346	0.0545	0.3484	0.1112
4	-7.18	0.2777	0.2480	0.4481	0.1842
Data rate		$179bps$	$149bps$	$179bps$	$149bps$

5 Conclusion

In this paper, an OFDM-based multicarrier differential chaos shift keying with subcarrier index modulation has been proposed for the UWA communications. The subcarrier index modulation is used in each sub-block to carry additional information bits, and therefore the proposed scheme can achieve a higher data rate than that of MC-CS-DCSK when appropriate parameters are used. The BER performances of the proposed system and benchmark systems are evaluated by Watermark simulator, water pool and lake field experiments. For the Watermark simulations over the NOF1 and NCS1 channel, SIM-CS-DCSK achieves better BER performance than its competitors at the high SNR range with appropriate SIM parameters. Considering the advantages of SIM-CS-DCSK in terms of the tradeoff between the data rate and BER performance, the good choice for number of activated subcarriers is half of the number of subcarriers in each sub-block. The water pool and lake field experiment results further verify that the superiority of SIM-CS-DCSK system in terms of BER performance and data rate.

References

1. Jahanbakht, M., Xiang, W., Hanzo, L., Rahimi, A.M.: Internet of underwater things and big marine data analytics: a comprehensive survey. IEEE Commun. Surv. Tut. **23**(2), 904–956 (2021)
2. Diamant, R., Lampe, L.: Low probability of detection for underwater acoustic communication: a review. IEEE Access **6**, 19099–19112 (2018)
3. Qarabaqi, P., Stojanovic, M.: Statistical characterization and computationally efficient modeling of a class of underwater acoustic communication channels. IEEE J. Ocean. Eng. **38**(4), 701–717 (2013)

4. Huang, J., Wang, H., He, C., Zhang, Q., Jing, L.: Underwater acoustic communication and the general performance evaluation criteria. Front. Inf. Technol. Electron. Eng. **19**(8), 951–971 (2018). https://doi.org/10.1631/FITEE.1700775
5. Hu, X., Wang, D., Lin, Y., Su, W., Xie, Y., Liu, L.: Multi-channel time frequency shift keying in underwater acoustic communication. Appl. Acoust. Part A **103**, 54–63 (2016)
6. Yang, T.C.: Properties of underwater acoustic communication channels in shallow water. J. Acoust. Soc. Am. **131**(1), 129–145 (2012)
7. Ling, J., He, H., Li, J., Roberts, W., Stoica, P.: Covert underwater acoustic communications. J. Acoust. Soc. Am. **128**(5), 2898–2909 (2010)
8. Xu, X., Zhou, S.: Per-survivor processing for underwater acoustic communications with direct-sequence spread spectrum. J. Acoust. Soc. Am. **133**(5), 2746–2754 (2013)
9. Kuai, X., Zhou, S., Wang, Z., Cheng, E.: Receiver design for spread-spectrum communications with a small spread in underwater clustered multipath channels. J. Acoust. Soc. Am. **141**(3), 1627–1642 (2017)
10. Liu, Z., Yoo, K., Yang, T.C., Cho, S.E., Song, H.C., Ensberg, D.E.: Long-range double-differentially coded spread spectrum acoustic communications with a towed array. IEEE J. Ocean. Eng. **39**(3), 482–490 (2014)
11. Qu, F., Qin, X., Yang, L., Yang, T.C.: Spread-spectrum method Using multiple sequences for underwater acoustic communications. IEEE J. Ocean. Eng. **43**(4), 1215–1226 (2018)
12. Sherlock, B., Neasham, J.A., Tsimenidis, C.C.: Spread-spectrum techniques for bio-friendly underwater acoustic communications. IEEE Access **6**, 4506–4520 (2018)
13. Dedieu, H., Kennedy, M.P., Hasler, M.: Chaos shift keying: modulation and demodulation of a chaotic carrier using self-synchronizing Chua's circuits. IEEE Trans. Circuits Syst. Analog. II Digit. Signal Process **40**(10), 634–642 (1993)
14. Fang, Y., Han, G., Chen, P., Lau, F.C.M., Chen, G., Wang, L.: A survey on DCSK-based communication systems and their application to UWB scenarios. IEEE Commun. Surv. Tut. **18**(3), 1804–1837 (2016)
15. Shu, X., Wang, J., Wang, H., Yang, X.: Chaotic direct sequence spread spectrum for secure underwater acoustic communication. Appl. Acoust. **104**, 57–66 (2016)
16. Bai, C., Ren, H.P., Li, J.: A differential chaos-shift keying scheme based on hybrid system for underwater acoustic communication. In: Proceedings IEEE/OES China Ocean Acoustics (COA), pp. 1–5. IEEE, Harbin, China (2016)
17. Bai, C., Ren, H.P., Grebogi, C., Baptista, M.S.: Chaos-based underwater communication with arbitrary transducers and bandwidth. Appl. Sci. **8**(2), 1–11 (2018)
18. Bai, C., Ren, H.P., Baptista, M.S., Grebogi, C.: Digital underwater communication with chaos. Commun. Nonlinear Sci. **73**, 14–24 (2019)
19. Xu, W., Wang, L., Kolumban, G.: A novel differential chaos shift keying modulation scheme. Int. J. Bifur. Chaos **21**(3), 799–814 (2011)
20. Chen, M., Xu, W., Wang, D., Wang, L.: A multi-carrier chaotic communication scheme for underwater acoustic communications. IET Commun. **13**(14), 2097–2105 (2019)
21. Xu, F., Zhan, C., Xie, Y., Wang, D.: Performance of CZT-assisted parallel combinatory multicarrier Frequency-Hopping Spread Spectrum over shallow underwater acoustic channels. Ocean Eng. **110**, 116–125 (2015)
22. Basar, E.: Index modulation techniques for 5G wireless networks. IEEE Commun. Mag. **54**(7), 168–175 (2016)

23. Wen, M., Cheng, X., Yang, L., Li, Y., Cheng, X., Ji, F.: Index modulated OFDM for underwater acoustic communications. IEEE Commun. Mag. **54**(5), 132–137 (2015)
24. van Walree, P.A.: Comparison between direct-sequence and multiuser spread-spectrum acoustic communications in time varying channels. J. Acoust. Soc. Am. **128**(6), 3525–3534 (2010)
25. Tu, K., Fertonani, D., Duman, T.M., Stojanovic, M., Proakis, J.G., Hursky, P.: Mitigation of intercarrier interference for OFDM over time-varying underwater acoustic channels. IEEE J. Ocean. Eng. **36**(2), 156–171 (2011)
26. Aval, Y.M., Stojanovic, M.: Differentially coherent multichannel detection of acoustic OFDM signals. IEEE J. Ocean. Eng. **40**(2), 251–268 (2015)
27. van Walree, P.A., Socheleau, F., Otnes, R., Jenserud, T.: The watermark benchmark for underwater acoustic modulation schemes. IEEE J. Ocean. Eng. **42**(4), 1007–1018 (2017)

Constrained Graph Convolution Networks Based on Graph Enhancement for Collaborative Filtering

Jingjing Zhang, Zhaogong Zhang(✉), and Xin Xu

School of Computer Science and Technology, Heilongjiang University, Harbin, China
2013010@hlju.edu.cn

Abstract. Graph Convolutional Networks (GCNs) have gained much attention and have achieved excellent performance in many graph-based collaborative filtering (CF) tasks in recent years. Its success relies on a fundamental assumption that the original graph structure is reliable and consistent with the properties of GNNs. However, most original graphs can seriously impair model performance due to noise and data sparsity problems. In addition, for large user-item graphs, the explicit message passing in traditional GCNs slows down the convergence speed during training and weakens the training efficiency of the model. Based on this, we propose Constrained Graph Convolution Networks Based on Graph Enhancement for Collaborative Filtering (EL-GCCF). The graph initialization learning module integrates the structural and feature information in the graph by generating two graph structures. It enhances the original graph and effectively mitigates the noise problem. Second, the multi-task constrained graph convolution skips explicit message passing. It effectively mitigates the over-smoothing problem in training and improves the training efficiency of the model by using an auxiliary sampling strategy. Experimental results on two real datasets show that the EL-GCCF model outperforms many mainstream models and has higher training efficiency.

Keywords: Recommendation system · Collaborative filtering · Graph enhancement · Graph convolutional network

1 Introduction

Graph neural networks (GNNs) have improved recommendation tasks in recent years. Some recent studies [1,2] chose graph convolution network (GCN) [3] to learn the embedding of nodes. Currently, the success of existing GCN-based CF models (NGCF [1], LightGCN [4], and LR-GCCF [5]) relies on an underlying assumption that the original user-item graph is reliable and consistent with the properties of GNNs. However, graphs are usually built on complex interactive systems. In reality, this assumption does not entirely hold. The original graph inevitably contains noisy information, such as missing, meaningless, or spurious edges, the propagation of noisy information in the graph deteriorates the quality of many other representations. Second, data sparsity is also a big problem. Due to

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 635–643, 2022.
https://doi.org/10.1007/978-3-031-19214-2_53

limited a priori knowledge, the predefined raw graphs carry only partial information, limiting the prediction task's accuracy. Most of the original graphs may not be optimal for the downstream tasks of the recommendation system. Unreliable graph structure may severely limit the representational power of GNNs [6].

In downstream tasks, to capture higher-order collaboration signals and better model the user-item interaction. Current GCN-based CF models tend to find increasingly complex network encoders [7]. However, we note that these methods are difficult to train on large user-item graphs. For this reason, several studies [4,5] have been carried out to simplify GCN-based CF models. For example, LR-GCCF borrows the graph linkage theory from SGCN [8], which simplifies the traditional GCN to a large extent. However, it does not get much improvement in training speed. Through empirical analysis, it is possible that message passing of stacking multiple layers causes the model to converge slowly, i.e., message passing still dominates the model's training. Therefore, avoiding noisy information, alleviating data sparsity, and constructing reliable graph structures while improving the efficiency of GCN models is still a problem to be solved.

We propose Constrained Graph Convolution Networks Based on Graph Enhancement for Collaborative Filtering (EL-GCCF) based on the above problems. The model consists of two modules: (1) Graph initialization learning module is used to generate enhanced graphs suitable for downstream tasks. (2) Constrained graph convolution module with an auxiliary sampling strategy. Our main contributions are as follows:

(1) We study the poor performance of existing GCN-based CF models. GNNs rely on a reliable graph structure. The original graph can seriously impair the model performance due to noise and data sparsity problems; the explicit message passing in traditional GCNs largely slows down the convergence speed of GCNs during training, which weakens the training efficiency of the model.
(2) We propose a novel EL-GCCF model. It enhances the original graph by the graph initialization learning method, which effectively alleviates the noise problem in the graph. In the enhanced graph, it skips explicit message passing via using constrained graph convolution. Also, it obtains efficient performance by using multi-task learning and auxiliary sampling strategies in the loss.
(3) Extensive experimental results on two real datasets show that EL-GCCF outperforms many mainstream GCN models and achieves more than a 4-fold speedup over LR-GCCF, verifying the model's effectiveness.

2 Related Work

2.1 Graph Neural Networks

In recent years, graph neural networks have received much attention and have achieved great success in solving the field of graph-based collaborative filtering

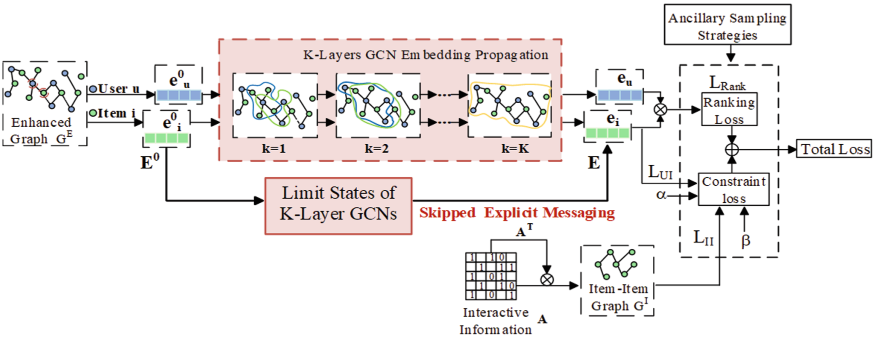


Fig. 1. Architecture of the proposed model

[1,4,5]. GNNs are used to learn the topology of the graph and the feature information of the nodes, and one of the most representative methods is the graph convolutional network (GCN) [3], which uses explicit message passing. It is an effective method to extract information from graph structure. The messaging process is defined as follows:

$$E^{(k+1)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}E^{(k)}W^{(k)}) \tag{1}$$

where $E^{(k)}$ and $W^{(k)}$ denote the embedding and weight matrix of the kth layer, A and D are the adjacency matrix and the diagonal node degree matrix, respectively.

3 Methodology

3.1 Problem Definition

Let $G = (V, E, F)$ be a graph, where $V = \{v_1, v_2, ..., v_N\}$ is the set of N nodes, E is the set of edges, $F = [f_1, f_2, ..., f_N]$ is the feature matrix of the nodes. f_i is the feature vector of node v_i. $F^U \in R^{|U| \times d^U}$ and $F^I \in R^{|I| \times d^I}$ are user and item feature matrices. $|d^U|$ and $|d^I|$ are the number of node attributes. $A \in R^{|U| \times |I|}$ is the user-item interaction matrix, $A_{ij} = <v_i, v_j>$ is used to describe the relationship between nodes. The framework diagram is shown in Fig. 1.

3.2 Graph Initialization Learning Module

Feature Interaction. The feature interaction graph S^{UI} determines the likelihood of an edge between nodes based on the node features. We adopt a mapping layer to project the feature f to the d^C-dimensional common feature $f^{'}$:

$$f_u' = \sigma(f_u \cdot W_u + b_u), f_i' = \sigma(f_i \cdot W_i + b_i) \tag{2}$$

where $\sigma(.)$ is the nonlinear activation function, W_u and W_i are the weight matrices, b_u and b_i are the bias terms, $f'_u \in R^{1\times d^C}$, $f'_i \in R^{1\times d^C}$.

According to the network homogeneity assumption, edges tend to connect similar nodes [9], and the graph structure can be optimized by promoting strong relational connections. We use cosine similarity for metric learning [10], threshold ε^{UI} for constraint, and thus capture the interactions in the feature matrix $F^{'}$.

$$\begin{aligned} &\Gamma^{UI}(f_x^{'}, f_y^{'}) = \cos(w^{UI} \odot f_x^{'}, w^{UI} \odot f_y^{'}) \\ &S^{UI}[x,y] = \begin{cases} \Gamma^{UI}(f_x^{'}, f_y^{'}) & \Gamma^{UI}(f_x^{'}, f_y^{'}) \geq \varepsilon^{UI} \\ 0 & otherwise \end{cases} \end{aligned} \tag{3}$$

where $\Gamma^{UI}(.)$ is the similarity metric function. By performing metric learning as in Eq. (3), we learn the feature interaction graph S^{UI}.

Feature Propagation. We calculate the similarity between user nodes in F^U to find similar users. We use the threshold ε^{FU} for similarity determination to filter the edges with small feature similarities. We obtain the user feature similarity graph $S^{FU} \in R^{|U|\times|U|}$ by metric learning:

$$\begin{aligned} &\Gamma^{FU}(f_x, f_y) = \cos(w^{FU} \odot f_x, w^{FU} \odot f_y) \\ &S^{FU}[x,y] = \begin{cases} \Gamma^{FU}(f_x, f_y) & \Gamma^{FU}(f_x, f_y) \geq \varepsilon^{FU} \\ 0 & otherwise \end{cases} \end{aligned} \tag{4}$$

Feature similarity propagation generates new interaction edges to obtain the feature propagation graph S^{FPU} and S^{FPI}. We use topology structure A for feature similarity propagation:

$$\begin{aligned} S^{FPU} &= S^{FU} \cdot A \\ S^{FPI} &= A \cdot S^{FI} \end{aligned} \tag{5}$$

Feature Graph Generation. We obtain the feature generation graph by combining two types of graph structures through the graph channel attention layer:

$$S^{Feat} = h([S^{UI}, S^{FPU}, S^{FPI}]) \tag{6}$$

where $[S^{UI}, S^{FPU}, S^{FPI}] \in R^{|U|\times|I|\times 3}$ is the stacked matrix of three candidate graphs and $S^{Feat} \in R^{|U|\times|I|}$ is the feature generation graph. $h(.)$ denotes a graph channel attention layer whose weight matrix W^{Feat} represents the importance of different candidate graphs.

3.3 Constrained Graph Convolution Module with an Auxiliary Sampling Strategy

User-Item Graph Learning. EL-GCCF model uses the limit method to approximate the final representation of the aggregation process. The final convergence condition of the model can be defined according to the theorem [11] as:

$$e_u = \lim_{k\to\infty} e_u^{(k+1)} = \lim_{k\to\infty} e_u^{(k)} \tag{7}$$

The model reaches convergence when the embedding representations of the last two layers remain unchanged. For [5], when the model reaches convergence, its linear embedding process can be rewritten as:

$$e_u = \frac{1}{d_u} e_u + \sum_{i \in N_u} \frac{1}{d_i \times d_u} e_i \tag{8}$$

where e_u and e_i are the embedding representations of the nodes in the final convergence state. If each node satisfies the Eq. (9) during the training process, the model is considered to reach the convergence state of message passing. The following convergence state can be derived after multi-step simplification:

$$e_u = \sum_{i \in N_u} w_{ui} e_i, w_{ui} = \frac{1}{d_i(d_u - 1)} \tag{9}$$

We hope the model will no longer use the explicit message passing of stacking multiple layers but directly approximate the convergence state. The most straightforward way is to minimize the difference between the two sides of the Eq. (9), i.e., maximize cosine similarity:

$$\max \sum_{i \in N_u} w_{ui} e_u^T e_i, \forall u \in U \tag{10}$$

We introduce the sigmoid activation function and the negative log-likelihood function to facilitate optimisation. The loss function is as follows:

$$L_{UI} = -\sum_{u \in U} \sum_{i \in N_u} w_{ui} \log(\sigma(e_u^T e_i) \tag{11}$$

where the limit conditions will structurally constrain the loss, L_{UI} is the constraint function, and w_{ui} is the constraint coefficient.

However, the current loss will still be affected by over-smoothing. User and item nodes will be easily aggregated into the same embedding. We sample difficult samples via using an auxiliary sampling strategy during training. We define a hard positive example as an item that is further from the user than at least one negative item. Formally, a positive pair $\{u, i\}$ is selected if:

$$E_{ui} \geq \min_{j \in N_u} E_{uj} - \varepsilon \tag{12}$$

where E is the Euclidean distance, N_u is the set of negative items for u, and ε is a margin parameter that controls the degree of separation.With the inclusion of hard positive and negative samples, the loss function can be rewritten as:

$$L_{UI} = -\sum_{u \in U} \sum_{\substack{(u,i) \in N^+ \\ (u,j) \in N^-}} w_{ui} \log(\sigma(e_u^T e_i) + w_{uj} \log(\sigma(-e_u^T e_j)) \tag{13}$$

where N^{+} and N^{-} are positive and negative samples respectively. L_{UI} is the constraint loss function. $\hat{y}_{ui} = e_u^T e_i$ is used as the recommended ranking score. The ranking loss L_{Rank} is shown below:

$$L_{Rank} = \sum_{u \in U} \sum_{\substack{(u,i) \in N^{+} \\ (u,j) \in N^{-}}} - \log(\sigma(\hat{y}_{ui} - \hat{y}_{uj})) + \lambda ||E^{(0)}||^{2} \tag{14}$$

Item-Item Graph Learning. In addition to user-item relationships, item-item relationships are equally important. We construct the item-item graph G^{I} based on the co-occurrence of items as follows:

$$G^{I} = A^{T} A \tag{15}$$

where each entry denotes the co-occurrences of two items, A is the adjacency matrix of the user-item graph. We follow Eq. (9) to approximate infinite-layer graph convolution on G^{I} and derive the new constraint coefficient w_{ij}:

$$w_{ij} = \frac{G_{i,j}}{g_i - G_{i,i}} (\frac{g_i}{g_j})^{c}, g_i = \sum_{k} G_{i,k} \tag{16}$$

where g_i and g_j denote the degrees (sum by column) of item i and item j in G^{I}, respectively. c is usually taken as 0.5 from experience.

Compared with the direct construction of item-item graph, we construct the co-occurrence relation graph by the adjacency matrix of user-item graph, which can reduce the training difficulty of the whole multi-task model. The loss function is as follows:

$$L_{II} = - \sum_{(u,i) \in N^{+}} \sum_{j \in S(i)} w_{ij} \log(\sigma(e_u^T e_j) \tag{17}$$

where N^{+} is positive samples. Thus, with this constraint loss, we extend the model to learn item-item relationships better and ultimately derive the total training objective for the model:

$$L = L_{Rank} + \alpha L_{UI} + \beta L_{II} \tag{18}$$

where α and β are hyper-parameters used to adjust the relative importance of the user-user and item-item relationships, respectively.

4 Experiments

4.1 Experiment Setup

Datasets and Evaluation Metrics. We have conducted extensive experiments on two available real datasets: MovieLens-1M and Amazon-Books. Table 1 shows the statistical information of the used dataset. Recall@K and NDCG@K (K = 10, 20) are used as evaluation metrics in this paper.

Table 1. Statistics of the two datasets

Dataset	Users	Items	Interactions	Density
MovieLens-1M	6022	3043	995154	5.431%
Amazon-Books	52643	91599	2984108	0.062%

Table 2. Performance comparison on two datasets

Model	Amazon-Books				MovieLens-1M			
	Recall@10	*NDCG*@10	*Recall*@20	*NDCG*@20	*Recall*@10	*NDCG*@10	*Recall*@20	*NDCG*@20
MF-BPR	0.0607	0.043	0.0956	0.0536	0.1704	0.2044	0.2153	0.2175
NeuMF	0.0507	0.0351	0.0823	0.0447	0.1657	0.1953	0.2106	0.2067
DeepWalk	0.0286	0.02511	0.0346	0.0264	0.1248	0.1025	0.1348	0.1057
Node2Vec	0.0301	0.2936	0.0402	0.0309	0.1347	0.1095	0.1475	0.1186
NGCF	0.0617	0.0427	0.0978	0.0547	0.1846	0.2328	0.2513	0.2511
LightGCN	0.0797	0.0565	0.1206	0.0689	0.1876	0.2314	0.2576	0.2427
LR-GCCF	0.0591	0.0504	0.1135	0.0558	0.1785	0.2051	0.2231	0.2124
EL-GCCF	**0.0973**	**0.0643**	**0.1363**	**0.0768**	**0.1925**	**0.2636**	**0.2657**	**0.2882**
Improv	64.64%	27.58%	20.01%	37.63%	7.84%	28.52%	19.09%	35.69%

4.2 Performance Comparison

Table 2 reports the results of the overall performance comparison. We have the following observations:

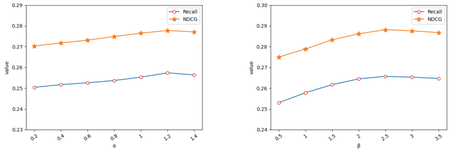
Compared to MF-based models(MF-BPR [12], NeuMF [13]), all GCN-based models perform better. [1,4,5] can use powerful graph convolution to learn higher-order relationships between users and items to capture deeper collaborative information. GCN-based models outperformed network embedding models (DeepWalk [14], Node2Vec [15]). It can make full use of the structural information of the graph, and graph convolution is more effective than traditional random walk to capture collaborative information. Compared with GCN-based models, EL-GCCF performs better. Instead of using explicit message passing, EL-GCCF learns the nature of infinite-layer graph convolution through the constraint loss with multi-tasking. Compared with other baseline models, EL-GCCF can perform enhanced learning on the original graph using the graph initialization learning method. It effectively alleviates the problem of noise and data sparsity in the original graph. In general, EL-GCCF achieves consistent and better performance on all datasets.

4.3 Ablation Study

Effectiveness of Constrained Graph Convolution with an Auxiliary Sampling Strategy. In order to compare the effects of different components on the model, we compared the EL-GCCF model and its variants. In Table 3 are the results of the comparison.

Table 3. Performance of variant models on two datasets

Model	Amazon-Books		MovieLens-1M	
	Recall@20	*NDCG*@20	*Recall*@20	*NDCG*@20
EL-GCCF(null)	0.1135	0.0558	0.2231	0.2124
EL-GCCF(α)	0.1162	0.0583	0.239	0.2325
EL-GCCF(β)	0.0942	0.0373	0.2187	0.2037
EL-GCCF(n_s)	0.1278	0.0688	0.2633	0.2742
EL-GCCF	**0.1363**	**0.0768**	**0.2657**	**0.2882**



(a) Impact of α (b) Impact of β

Fig. 2. Effect of different hyperparameters on model performance

(1) Importance of multi-task learning. EL-GCCF(α) uses constrained graph convolution with an auxiliary sampling only on user-item graphs, EL-GCCF(β) learns only on item-item graphs. EL-GCCF performs much better than both EL-GCCF(α) and EL-GCCF(β), indicating that EL-GCCF can comprehensively learn different types of relationships in the graph via using multi-task learning.

(2) Importance of constrained graph convolution. EL-GCCF(null) uses traditional explicit message passing for embedding learning. EL-GCCF, EL-GCCF(α), and EL-GCCF(β) outperform EL-GCCF(null) in general, indicating that the explicit message passing of stacking multiple layers does limit the model performance improvement. Constrained graph convolution can effectively learn the information in the graph to improve the model's performance.

(3) Importance of ancillary sampling strategies. EL-GCCF(n_s) uses only a simple sampling strategy. EL-GCCF outperforms EL-GCCF(n_s), indicating that the model uses an auxiliary sampling strategy to consider the training problem of difficult samples, which can effectively alleviate the problem of over-smoothing and help the model converge quickly during training.

Effect of Hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. We first identify the value of hyperparameter α from [0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4] when the model performs optimally. Then we test with different β in [0.5, 1, 1.5, 2, 2.5, 3, 3.5] based on the optimal α. We conducted experiments on the MovieLens-1M dataset, and the results are shown in Fig. 2. We find that smaller hyperparameters limit the play of the constraint loss, where β is more significant. EL-GCCF performs best when $\alpha = 1.2$ and $\beta = 2.5$. Appropriate parameter settings allow EL-GCCF to learn different types of relations effectively.

5 Conclusion

In this paper, We study the poor performance of existing GCN-based CF models and propose constrained graph convolution networks based on graph enhancement for collaborative filtering (EL-GCCF). Finally, The experimental results on two datasets show the effectiveness of EL-GCCF.

References

1. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering, pp. 165–174 (2019)
2. Sun, J., et al.: Neighbor interaction aware graph convolution networks for recommendation, pp. 1289–1298 (2020)
3. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2016)
4. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: simplifying and powering graph convolution network for recommendation, pp. 639–648 (2020)
5. Chen, L., Wu, L., Hong, R., Zhang, K., Wang, M.: Revisiting graph based collaborative filtering: a linear residual graph convolutional network approach. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 27–34 (2020)
6. Zhang, Y., Pal, S., Coates, M., Ustebay, D.: Bayesian graph convolutional neural networks for semi-supervised classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5829–5836 (2019)
7. Yu, W., Qin, Z.: Graph convolutional network for recommendation with low-pass collaborative filters, pp. 10936–10945 (2020)
8. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: International Conference on Machine Learning, pp. 6861–6871. PMLR (2019)
9. Newman, M.: Networks. Oxford University Press, Oxford (2018)
10. Luo, D., et al.: Learning to drop: robust graph neural network via topological denoising, pp. 779–787 (2021)
11. Chen, M., Wei, Z., Huang, Z., Ding, B., Li, Y.: Simple and deep graph convolutional networks. In: International Conference on Machine Learning, pp. 1725–1735. PMLR (2020)
12. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback (2012)
13. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
14. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
15. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)

Network Intrusion Detection Based on Hybrid Neural Network

Guofeng He(✉), Qing Lu, Guangqiang Yin, and Hu Xiong

University of Electronic Science and Technology of China, Chengdu 611731, China
{201812090812,qlu}@std.uestc.edu.cn, yingq@uestc.edu.cn

Abstract. The rapid development of the Internet has brought great changes and convenience to the society and people. With the development of the Internet, its security has been paid more and more attention. Intrusion detection can detect network attacks in real time and respond to them in time, which has become an essential and important security line. With the novel of network attack and the diversification of network traffic, traditional intrusion detection based on attack load matching and the intrusion detection based on machine learning has problems of inaccurate feature extraction and insufficient detection effect. To solve the above problems, this paper designs a hybrid neural network DCT-IDS model, using dense convolution neural network to achieve traffic feature fusion, reducing the number of parameters, using Transformer to extract time sequence features, and experimental tests were carried out on the latest dataset CIC-IDS2018. The experimental results show that the accuracy of the proposed DCT-IDS model reaches 98%, and all the indexes are better than the existing excellent models.

Keywords: Intrusion detection · Deep learning · Convolutional neural network · Transformer

1 Introduction

In recent years, with the emergence of new concepts such as the Internet of everything and cloud computing, the Internet has entered a new era. The development of the Internet not only brings breakthrough innovation and historic transformation to the society, but also provides the possibility for some illegal elements to carry out malicious activities on the Internet. Internet security is one of the most concerned issues today. Hackers can steal personal privacy, refuse to provide services, interrupt business, commit financial fraud, demand ransom, destroy physical equipment and other criminal acts [1] through the network, which will bring huge losses to individuals, enterprises and governments. Network threats are increasing day by day.

Intrusion detection can protect the system in real time by monitoring the behavior of network and host and judging whether it conforms to the preset policy. According to the detection principle, intrusion detection is mainly divided into anomaly detection and signature matching detection [2]. Signature matching intrusion detection collects the features of various attacks and constructs the corresponding feature database. If

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 644–655, 2022.
https://doi.org/10.1007/978-3-031-19214-2_54

the database finds relevant matching features after analysis, the intrusion is judged to exist. This method has the advantages of high detection rate, few false positives and no need to go through complex model calculation. However, its disadvantages are that it is limited to the detection of known attacks and cannot identify attacks that have not oc-curred before. Moreover, it is difficult to define the feature database uniformly, and its iterative update also requires a certain cost. Anomaly detection firstly extracts the behavior characteristics of normal operations of users, and all operations that do not conform to normal operations are judged as intrusion behaviors. Because of the novelty of network attack and the rapid development of Internet, anomaly detection has become a research hotspot in the academic world and a mainstream intrusion detection method in industry.

This paper designs a hybrid DCT-IDS model based on convolutional neural network and Transformer, which optimizes the network structure, reduces the number of parameters, speeds up the training efficiency and improves the detection performance. Firstly, the structure of traffic data is fully considered, and the dense structure based CNN network is used to extract the underlying features of data packets, reducing the number of parameters, and realizing feature fusion by residual structure. Transformer is used to extract the timing features of data packets, and combined with self-attention and multi-attention mechanism, on the one hand, it solves the problem of forgetting the sample information of some attacks lasting for a long time, and on the other hand, it realizes the self-selection of important features.

2 Related Work

The application of machine learning in the field of intrusion detection has been studied for 20 years [3], and the literature proves that methods based on machine learning have better performance than traditional methods. Li et al. [4] designed an IDS based on PCA-SVM. PCA is first used to extract important features, and then SVM is used for traffic prediction. PCA uses statistical information to reduce the dimension of feature vector, which effectively improves the training efficiency of model. Li et al. [4] designed a new IDS using support vector machine algorithm. The author uses a strategy called feature removal method for feature selection, which selects the most important features by evaluating the impact on a specific classifier. Although the algorithm achieves high accuracy, its tedious feature selection process has a high computational cost, which cannot meet the needs of practical scenes. Wang et al. [5] used artificial neural network (ANN) and fuzzy clustering (FC) methods to realize an FC-ANN based IDS, which improved the detection accuracy and reduced the FAR index.

Because the performance of deep learning in many fields has far exceeded the traditional machine learning methods, and deep learning does not need Feature Engineering, a large number of security researchers have carried out research on deep learning. The research on CNN has a long history. Li et al. Finally used the simple Lenet5 for traffic classification. Literature [6–11] successively used various variant structures based on CNN, such as densenet, RESNET and googlenet. After investigation, it is found that the experimental results of the above model are not much higher than CNN. Cyclic neural network and its variants have always been a research hotspot, because it can extract the time-series characteristics of traffic. Yakubu et al. [12] used the improved bilstm to improve the accuracy compared with the traditional LSTM, but the training time of the model also increases. Singh et al. [13] combined Gru and transfer learning. This method improves the generalization ability of the model, but the improvement of detection rate is not obvious. Therefore, another research [14–20] starts with the traffic characteristics, selects the features that are more beneficial to the detection results through various feature selection methods, and then classifies them in combination with the deep learning model. The experiments show that this method does effectively improve the detection accuracy, and the training time is shortened due to the reduction of features. After that, some researches began to use different networks for different feature extraction and fusion. S. et al. [21] and others used CNN and LSTM for spatio-temporal feature extraction without considering the optimization of network structure. Kanna [22] et al. Connected CNN and multi-layer LSTM in series and optimized the super parameters of CNN. The experiment shows that the accuracy can reach more than 90% on unswnb15 data set, but the training time is not mentioned in the paper. Yao et al. [23] also use CNN and LSTM. The difference is that the outputs of the two networks will aggregate together for feature fusion. Experiments show that the performance is improved compared with series combination. However, the data set used in this paper is nslkdd, which has no practical value.

Yang [24] used bidirectional LSTM RNN for anomaly detection and multi-classification attack identification. The results show that although the average accuracy of BiLSTM can reach 93.00%, which is better than that of ordinary LSTM, the problem of class imbalance exists, which makes the generalization ability of the system not high. Hussain [25] and Wu et al. [26] used the CNN network based on ResNet and GoogleNet structure for traffic classification, but the number of original network parameters was large and there was redundancy. On the basis of previous studies, Shang et al. [27] proposed to use the combination of DSCNN and BI-LSTM to build a model to obtain spatio-temporal features, but there are too many model parameters and it is easy to fall into local optimization.

3 Model Construction

This paper designs an intrusion detection model DCT-IDS based on multi-level feature ex-traction. This model has the following advantages: firstly, the structure of traffic data is fully considered, and the underlying features of data packets are extracted by CNN network [28] based on dense structure [29], and the timing features of data packets are extracted by Transformer, and feature fusion is realized by residual structure. Secondly, the dense link mechanism is introduced to reduce the number of parameters and strengthen feature reuse. Finally, the attention mechanism based on Transformer [30] structure is introduced. Combining self-attention and multi-attention mechanism, on the one hand, the problem of forgetting the sample information of some attacks lasting for a long time is solved, and on the other hand, the self-selection of important features is realized.

The overall framework of the intrusion detection model DCT-IDS is shown in Fig. 1. The framework consists of four modules: data preprocessing module, Dense-CNN module, Transformer module and classified output module.

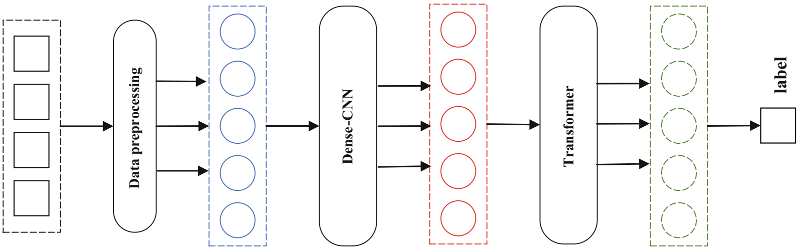


Fig. 1. DCT-IDS model framework

3.1 Dense-CNN Module

The whole structure of dense-CNN module is shown in Fig. 2. The input first goes through a 3*3 convolution layer and carries out convolution operation with 64 convolution kernels to obtain feature vectors of 64 channels. Then it is sent to DenseBlock to extract features. DenseBlock contains three layers of DenseLayer. The convolution kernel size of each DenseLayer layer is set to 3*3, and the channel is set to 64. Each layer DenseLayer the output of the input contains not only the adjacent layer, also include the front all of the output layer, they together, by means of concat fusion with the underlying characteristics of spatial information, the last layer, DenseLayer, connects all the previous outputs and inputs together to output the fused eigenvectors.

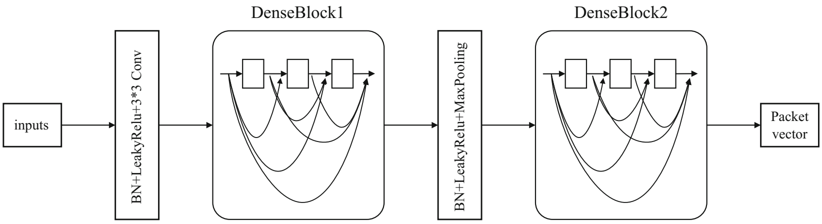


Fig. 2. Dense-CNN module

Each DenseLayer layer in the DenseBlock contains 128 1*1 convolution kernels and 64 3*3 convolution kernels. The 3*3 convolution kernel can be used to extract features, while the 1*1 convolution kernel is needed to reduce computation. In addition, the third layer DenseLayer adopts the empty convolution with Dilation Rate of 2. By introducing the parameter of Dilation Rate, the convolution kernel can obtain a larger receptive field and carry more information. Feature vectors extracted by DenseBlock will be sent to the maximum pooling layer, which replaces the original Transition layer and is used to connect two DenseBlock blocks and compress feature maps to reduce computation.

3.2 Transformer Module

As shown in Fig. 3, the structure of timing feature extraction module based on Transformer mainly consists of timing encoding, encoder and decoder. The feature vector

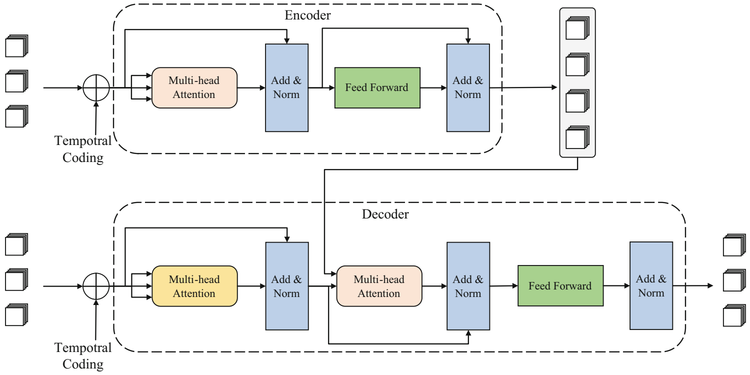


Fig. 3. Sequence feature extraction module structure

extracted from the underlying feature module is a two dimensional X $\in R^{S \times d_{model}}$, S is the length of feature vector, d_{model} is the dimension of feature after Embedding, this tensor is input into encoder and decoder respectively after sequence encoding. As mentioned earlier, the Transformer input is parallel and lacks the sequential flow sequence information compared to the serial input, so a sequential encoding module is required to compute a position vector that can record the sequential flow sequence information and embed it into the features in an additive manner.

After calculating the eigenvectors and position vectors, we can get $\hat{\mathrm{X}} \in \mathrm{R}^{\mathrm{S} \times \mathrm{d}_{\mathrm{model}}}$, then send them respectively to the encoder and decoder as the input, and the input is the same, the encoding module maps features into feature vector sets containing key vector K and value vector V, and then sends them to the decoder. The decoder adjusts the information it focuses on according to the attention vector output by the encoder, and finally outputs the decoded feature set.

3.3 Classification Output

The module is mainly composed of a global maximum pooling layer, a Dense layer with 32 neural units, and an output layer. The output layer uses the sigmoid activation function, and the rest of the layer uses the Leaky Relu function. After extracting the underlying and temporal features, the feature vectors were transformed into one dimensional vectors through global maximum pooling, and then reduced the feature dimension through the dense layer, and classified by softmax. In addition, the full connection layer is followed by alpha dropout, which is set to 0.2 and sets 20% of the neural units to 0 to reduce computational parameters and avoid overfitting. The loss function adopts cross entropy function.

4 Data Analysis

This paper adopts CIC-IDS2018 [31] dataset, which is a modern network traffic dataset. The definition of traffic characteristics and attack behavior is more consistent with contemporary network attacks, and the distribution of training set and test set conforms to

Table 1. Final experimental data distribution

Category	Number of training sets	Number of test sets	Category	Number of training sets	Number of test sets
Benign	50811	12698	Bot	11452	2863
Brute Force -Web	10669	2667	Brute Force -XSS	7412	1853
SQL injection	84	21	DDOS attack-HOIC	27840	6960
DDOS attack-LOIC-UDP	1465	366	DDoS attacks-LOIC-HTTP	23128	5782
DoS attacks-GoldenEye	16048	4012	DoS attacks-Hulk	18516	4629
DoS attacks-SlowHTTPTest	55593	13898	DoS attacks-Slowloris	6396	1599
FTP-BruteForce	488	122	SSH-Bruteforce	177	44
Infilteration	6096	1524			

statistical correlation. The dataset contains 3227424 traffic, including six types of attack methods: brute force cracking, botnet, Dos, DDos, Web attack, and network penetration. Among them, there are 2678039 normal traffic and 549385 attack traffic. The number difference is huge, and the normal traffic has great redundancy.

The normal traffic of THE CIC-IDS2018 dataset is simulated by scripts for several communication protocols. Although the number is large, it is not real. In order to make the dataset more meaningful, this paper installed Winpcap library and Wireshark tool on three Windows hosts. The data packet was collected by Wireshark for 3 days. During the period of collection, common online behaviors such as chatting, downloading papers, browsing the web, sending emails, listening to music and playing online games were carried out. The collected DATA packets in pcap format are converted into csv files in network flow format by CICFlowMeter. In order to improve the training efficiency, this paper filters and combines the attack traffic in IDS2018 dataset and the normal traffic collected, and the distribution of the finally obtained data is shown in Table 1.

5 Experimental Results and Analysis

5.1 Network Structure Analysis

In order to optimize the model, this paper evaluated the performance of the model under different dense blocks, and conducted a binary classification experiment. The accuracy results are shown in Fig. 4. It can be seen that with the increase of DenseBlock, the accuracy can reach more than 98%, and the change is not great, but the parameters, complexity and training time of the model will greatly increase, so this paper chooses one DenseBlock to build our model.

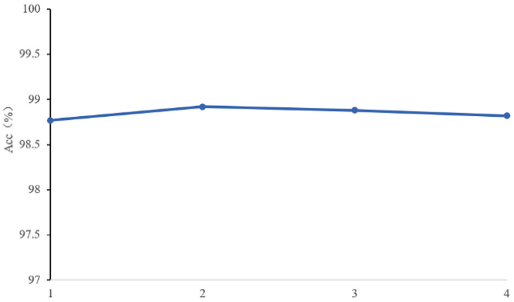


Fig. 4. Model accuracy under different Dense Blocks

5.2 Classification Experiment

The binary classification of network traffic is called exception detection, just need to distinguish normal traffic from abnormal traffic. Exception detection is a necessary function of IDS. Anomaly detection values the ability to identify abnormal traffic, so accuracy and F1-score are enough to judge the performance of the model. The experimental results of binary classification are shown in Table 2. It can be seen that in the abnormal detection scenario, the accuracy of the model for normal traffic and abnormal traffic is up to 98.87%, and the model can accurately detect the most traffic.

Table 2. Dichotomize the experimental results of each category

Category	Accuracy	Precision	Recall	F1-score
Attack	0.9887	0.9971	0.9766	0.9867
Benign	0.9887	0.9871	0.9994	0.9932

In order to verify the detection ability of the model proposed in this paper on attack categories, multi-classification experiments are carried out in this paper. The experimental results of multi-classification are shown in Table 3. As can be seen from Table 3, for categories with sufficient training samples, all indicators of the model performed well, with an accuracy of 98% and an average F1-score of 94.28%. However, for categories with a small number of training samples, its detection ability was far below our requirements, such as SQL Injection. Due to the rarity of training samples, the model could not learn such features at all, and thus could not detect them correctly, leading to 0 for all four indicators. The same problem also occurred in SSH-BruteForce, FTP-BruteForce and DDOS attack-LOIC-UDP categories. This shows that unbalanced datasets have a great impact on the detection effect of the model, and data balance processing is particularly necessary. In addition, although the accuracy rate of the model for the Infilteration category is as high as 97.01%, the recall rate and accuracy rate are not high, indicating that the model is not strong in identifying the attack samples of the Infilteration category, and the high accuracy is because it is judged as other attack categories. The training samples in the Infilteration category are sufficient, and there is no impact of category imbalance. This may be because the Infilteration attack only uses NMAP to scan some IP addresses, ports and services, and does not cause actual harm. However, in the real network, such scans are ubiquitous, so the model does not recognize them well. When defending against such attacks, we can implement import and export policies with the firewall and filter out such scans by setting the blacklist of firewall ports and IP addresses, so that such attacks cannot enter the detection range of the intrusion detection system.

Table 3. Experimental results of multiple classification of DCT-IDS model

Category	Accuracy	Precision	Recall	F1-score
Benign	0.9792	0.9775	0.9796	0.9785
Bot	0.9896	0.9727	0.9862	0.9794
Brute Force -Web	0.9899	0.9838	0.9750	0.9793
Brute Force -XSS	0.9889	0.9835	0.9749	0.9767
SQL Injection	0	0	0	0
DDOS attack-HOIC	0.9897	0.9828	0.9575	0.9698
DDOS attack-LOIC-UDP	0.9854	0.8477	0.6456	0.7329
DDoS attacks-LOIC-HTTP	0.9841	0.9866	0.9869	0.9867
DoS attacks-GoldenEye	0.9899	0.9933	0.9723	0.9827
DoS attacks-Hulk	0.9896	0.9737	0.9895	0.9815
DoS attacks-SlowHTTPTest	0.9842	0.9814	0.9873	0.9843
DoS attacks-Slowloris	0.9899	0.9827	0.9883	0.9855
FTP-BruteForce	0.9842	0.7022	0.3341	0.4527
SSH-Bruteforce	0.9792	0.8822	0.1251	0.2191
Infilteration	0.9701	0.5596	0.4120	0.4746

5.3 Comparative Experiment

In order to illustrate the advantages of this model in detection performance, SVM [32], Naive Bayes [33], random forest [34] and KNN [35] models in machine learning algorithm are selected in this paper to conduct transverse comparison experiments with the model proposed in this paper, and the experimental results are shown in Fig. 5. It can be obviously found that among machine learning algorithms, random forest has the best detection effect, but compared with the model proposed in this paper, the model proposed in this paper is much better than random forest in accuracy and F1 value.

In order to highlight the advantages of this model, it is necessary to compare the excellent deep learning models in the field of intrusion detection. In this paper, the models CNN, LSTM and CNN-LSTM in literature are selected as comparison, and dichotomous and multi-classification experiments are carried out.

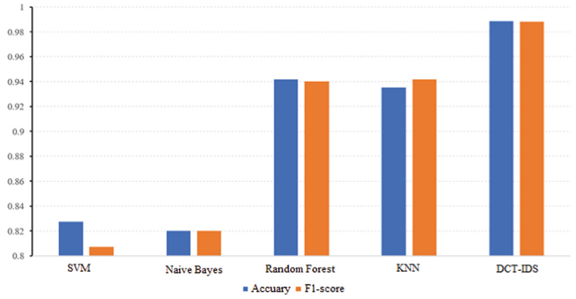


Fig. 5. Compared with machine learning algorithms of binary classification results

Transverse comparison of experimental results is shown in Fig. 6. It can be seen that the performance of LSTM is better than CNN, indicating that LSTM is more dominant in the process of traffic data related to timing. The serial model of CNN and LSTM takes into account the extraction of spatio-temporal features, and its performance is slightly worse than that of the model in this paper. However, its structure is single and network optimization is not taken into consideration deeply. The DCT-IDS model proposed in this paper is optimized and improved for feature fusion and network structure, and the experimental results show that the comprehensive performance of this model is better than other models.

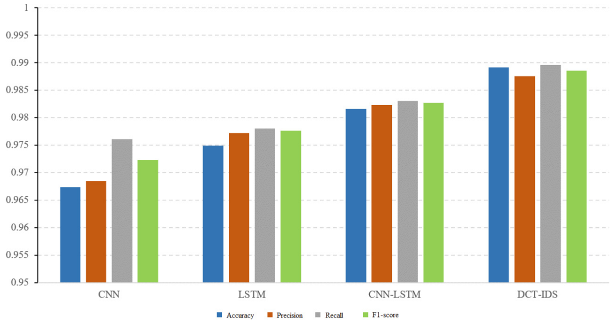


Fig. 6. Dichotomous experimental results of different deep learning models

F1-score of different deep learning models is shown in Table 4. It can be found that for categories with sufficient training samples, CNN and LSTM have different F1-score in each category. Due to LSTM's timing learning ability, its detection ability is stronger than CNN for time-related attack categories. For example, in the DoS Attacks-SlowHTTPTest and DoS attacks-Slowloris categories, LSTM outperformed CNN by 2 percentage points. The F1-score of CNN-LSTM series model is better than CNN and LSTM alone for each category, and the F1-score of the model in this paper is higher than other models. For a few sample categories, no matter the comparison model or the model in this paper, due to insufficient training, the detection ability of a few sample categories is low and not ideal, and F1-score varies greatly. This also indicates that data imbalance will greatly affect the detection results. For the Infilteration category, all models have low detection ability, but this model has the best detection effect. In addition, the model in this paper can also detect part of SSH-BruteForce data, while F1-score of which is 0 for the comparison model. Based on the above experiments, the model proposed in this paper has excellent performance in all indicators, and its detection performance is better than that of the comparative machine learning and deep learning models.

Table 4. F1-score of Multi-classification experiments of different deep learning models

Category	CNN	LSTM	CNN-LSTM	DCT-IDS
Benign	0.9624	0.9708	0.9768	0.9785
Bot	0.9662	0.958	0.9675	0.9794
Brute Force -Web	0.9715	0.9636	0.9731	0.9793
Brute Force -XSS	0.9714	0.9675	0.9745	0.9767
SQL Injection	0	0	0	0
DDOS attack-HOIC	0.9514	0.9629	0.9654	0.9698
DDOS attack-LOIC-UDP	0.6441	0.6321	0.70	0.7329
DDoS attacks-LOIC-HTTP	0.9773	0.9718	0.9851	0.9867
DoS attacks-GoldenEye	0.9715	0.9727	0.9745	0.9827
DoS attacks-Hulk	0.9664	0.9752	0.9805	0.9815
DoS attacks-SlowHTTPTest	0.9525	0.9718	0.9772	0.9843
DoS attacks-Slowloris	0.9257	0.9757	0.9788	0.9863
FTP-BruteForce	0	0	0.1538	0.4527
SSH-BruteForce	0	0	0	0.2191
Infilteration	0.4001	0.4026	0.4210	0.4746

6 Conclusion

This paper designs an intrusion detection model based on multi-level feature extraction. In this model, dense connection mechanism was introduced to design the Dense-CNN network module, which was used to extract the underlying features of data packets. Compared with feature extraction based on traditional convolutional neural network, the calculation of parameters was reduced. Then, a sequence feature extraction module is designed based on Transformer, which uses multi-head self-attention mechanism and powerful feature extraction ability to extract the sequence feature of flow, and its parallel computing feature reduces the training time. The feature vectors after the two modules are fully fused, and finally the detection results are obtained by the classification output module. In order to verify the superiority of DCT-IDS model, we combined the latest dataset CIC-IDS 2018 with the actual collected normal traffic, and designed a binary and multi-classification comparison experiment with machine learning and deep learning model. The experimental results show that the detection performance of the proposed model is better than other models.

References

1. Fernandez, G.C.: Deep learning approaches for network intrusion detection. The University of Texas at San Antonio (2019)
2. Mishra, P., Varadharajan, V., Tupakula, U., et al.: A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Commun. Surv. Tutor. **21**(1), 686–728 (2018)
3. Mukkamala, S., Janoski, G., Sung. A.: Intrusion detection using neural networks and support vector machines. In: Proceedings of the 2002 International Joint Conference on Neural Networks, pp. 1702–1707. IEEE, Honolulu (2002)
4. Li, Y., Xia, J., Zhang, S., et al.: An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Syst. Appl. **39**(1), 424–430 (2012)
5. Wang, G., Hao, J., Ma, J., et al.: A new approach to intrusion detection using artificial neural networks and fuzzy clustering. Expert Syst. Appl. **37**(9), 6225–6232 (2010)
6. Li, Z., Qin, Z., Huang, K., Yang, X., Ye, S.: Intrusion Detection using convolutional neural networks for representation learning. In: Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.M. (eds.) ICONIP 2017. LNCS, vol. 10638, pp. 858–866. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70139-4_87
7. Li, X., Sheng, Z., Huang, Y., Zhang, S.: A new network intrusion detection method based on deep neural network. In: Sun, X., Wang, J., Bertino, E. (eds.) ICAIS 2020. CCIS, vol. 1254, pp. 46–57. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-8101-4_6
8. Shaikh, A., Gupta, P.: Real-time intrusion detection based on residual learning through ResNet algorithm. Int. J. Syst. Assur. Eng. Manag. 1–15 (2022). https://doi.org/10.1007/s13198-021-01558-1
9. Sun, M., Hao, X., Li, W.: Research on intrusion detection method based on PGoogLeNet-IDS model. In: Liang, Q., Wang, W., Mu, J., Liu, X., Na, Z., Cai, X. (eds) Artificial Intelligence in China. Lecture Notes in Electrical Engineering, vol. 653, pp. 315–323. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-8599-9_37
10. Zhipeng, C., Zaobo, H., Xin, G.: Collective data-sanitization for preventing sensitive information inference attacks in social networks. IEEE Trans. Depend. Secur. Comput. **15**(4), 577–590 (2018)
11. Zhipeng, C., Xu, Z.: A private and efficient mechanism for data uploading in smart cyber-physical systems. IEEE Trans. Netw. Sci. Eng. **7**(2): 766–775 (2020)
12. Imrana. Y., Xiang. Y., Ali. L.: A bidirectional LSTM deep learning approach for intrusion detection. Expert Syst. Appl. **185**(8), 115524 (2021)
13. Singh, N.B., Singh, M.M., Sarkar, A.: A novel wide & deep transfer learning stacked GRU framework for network intrusion detection. J. Inf. Secur. Appl. **2021**(61), 102899 (2021)
14. Ayo, F.E., Folorunso, S.O., Abayom, A.: Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection. Inf. Secur. J Glob. Perspect. **29**(6), 267–283 (2020)
15. Thakkar, A., Lohiya, R.: A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. Artif. Intell. Rev. **2021**(1), 1–111 (2021)
16. Riyaz, B., Ganapathy, S.: A deep learning approach for effective intrusion detection in wireless networks using CNN. Soft Comput. **24**(22), 17265–17278 (2020)

17. Alazzam, H., Sharieh, A., Sabri, K.E.: A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Expert Syst. Appl. **2020**(148), 113249 (2020)
18. Zhipeng, C., Xu, Z., Jinbao, W., Zaobo, H.: Private data trading towards range counting queries in Internet of Things. IEEE Trans. Mob. Comput. Early Access 2022
19. Zhipeng, C., Zaobo, H.: Trading private range counting over Big IoT data. In: The 39th IEEE International Conference on Distributed Computing Systems, pp. 144–153 (2019)
20. Zheng, X., Cai, Z.: Privacy-preserved data sharing towards multiple parties in industrial IoTs. IEEE J. Sel. Areas Commun. **38**(5), 968–979 (2020)
21. Emadi, S., Mohannadi, A., Senaid, F.: Using deep learning techniques for network intrusion detection. In: 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), pp. 171–176. IEEE, Doha (2020)
22. Kanna, P.R., Santhi, P.: Unified deep learning approach for efficient intrusion detection system using integrated spatial–temporal features. Knowl. Based Syst. **2021**(226), 107132 (2021)
23. Yao, R., Wang, N., Liu, Z.: Intrusion detection system in the advanced metering infrastructure: a cross-layer feature-fusion CNN-LSTM-based approach. Sensors **21**(2), 626 (2021)
24. Yang, S.U.: Research on network behavior anomaly analysis based on bidirectional LSTM. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 798–802. IEEE, Chengdu (2019)
25. Hussain, F., et al.: IoT DoS and DDoS attack detection using ResNet. In: 2020 IEEE 23rd International Multitopic Conference (INMIC), pp. 1–6. IEEE, Pakistan (2020)
26. Wu, K., Chen, Z., Li, W.: A novel intrusion detection model for a massive network using convolutional neural networks. IEEE Access **2018**(6), 50850–50859 (2018)
27. Sang, B., Han, Z., Lin, S.: Intrusion detection method based on DSCNN BiLSTM. Sci. Technol. Eng. **21**(8), 9 (2021)
28. Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET), pp. 1–6. IEEE, Antalya (2017)
29. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSP, pp. 108–116 (2018)
30. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. Adv. Neural. Inf. Process. Syst. **2017**(30), 5998–6008 (2017)
31. Ashish, V., Noam, S., Niki, P.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), pp. 6000–6010. ACM, NY (2017)
32. Noble, W.S.: What is a support vector machine?. Nat. Biotechnol. **24**(12), 1565–1567 (2006)
33. Murphy, K.P.: Naive Bayes classifiers. Univ. Br. Columbia **18**(60), 1–8 (2006)
34. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
35. Zhang, M.L., Zhou, Z.H.: ML-KNN: a lazy learning approach to multi-label learning. Pattern Recogn. **40**(7), 2038–2048 (2007)

Author Index