

Yang Xiang  
Zheli Liu  
Jin Li (Eds.)

Communications in Computer and Information Science

1298

# Security and Privacy in Social Networks and Big Data

6th International Symposium, SocialSec 2020  
Tianjin, China, September 26–27, 2020  
Proceedings

 Springer



SocialSec 2020

# Communications in Computer and Information Science

1298

*Commenced Publication in 2007*

Founding and Former Series Editors:

Simone Diniz Junqueira Barbosa, Phoebe Chen, Alfredo Cuzzocrea,  
Xiaoyong Du, Orhun Kara, Ting Liu, Krishna M. Sivalingam,  
Dominik Ślęzak, Takashi Washio, Xiaokang Yang, and Junsong Yuan

## Editorial Board Members

Joaquim Filipe 


*Polytechnic Institute of Setúbal, Setúbal, Portugal*

Ashish Ghosh

*Indian Statistical Institute, Kolkata, India*

Igor Kotenko 

*St. Petersburg Institute for Informatics and Automation of the Russian  
Academy of Sciences, St. Petersburg, Russia*

Raquel Oliveira Prates 

*Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil*

Lizhu Zhou

*Tsinghua University, Beijing, China*

More information about this series at <http://www.springer.com/series/7899>

Yang Xiang · Zheli Liu ·  
Jin Li (Eds.)

# Security and Privacy in Social Networks and Big Data

6th International Symposium, SocialSec 2020  
Tianjin, China, September 26–27, 2020  
Proceedings

*Editors*

Yang Xiang  
Swinburne University of Technology  
Melbourne, VIC, Australia

Zheli Liu  
Nankai University  
Tianjin, China

Jin Li  
Guangzhou University  
Guangzhou, China

ISSN 1865-0929                      ISSN 1865-0937 (electronic)  
Communications in Computer and Information Science  
ISBN 978-981-15-9030-6              ISBN 978-981-15-9031-3 (eBook)  
<https://doi.org/10.1007/978-981-15-9031-3>

© Springer Nature Singapore Pte Ltd. 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

# Preface

This volume contains the selected papers from the 6th International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec 2020), which was held in Tianjin, China, during September 26–27, 2020. The event was organized by Nankai University.

The purpose of SocialSec 2020 was to provide a leading-edge forum to foster interactions between researchers and developers with the security and privacy communities in social networks and big data, and to give attendees an opportunity to interact with experts in academia, industry, and government. Social networks and big data have pervaded all aspects of our daily lives. With their unparalleled popularity, social networks have evolved from the platforms for social communication and news dissemination, to indispensable tools for professional networking, social recommendations, marketing, and online content distribution. Social networks, together with other activities, produce big data that is beyond the ability of commonly used computer software and hardware tools to capture, manage, and process within a tolerable elapsed time. It has been widely recognized that security and privacy are the critical challenges for social networks and big data applications due to their scale, complexity, and heterogeneity. The SocialSec 2020 Organizing Committee presented 5 keynote speakers from the international leading researchers, and more than 10 invited talks from the distinguished experts on the front line of the security domain. To enhance communication between young researchers and renowned scholars, we held a special panel session entitled “How to Publish Papers in Top Conferences/Journals? Questions and Answers.” All accepted and presented papers at SocialSec 2020 were contested for the paper awards in a variety of tracks. SocialSec 2020 received 111 papers and all the submitted manuscripts were peer reviewed in a double-blind fashion by at least three qualified reviewers chosen from our Technical Committee members based on their qualifications. Eventually, 38 papers were finally accepted for publication, yielding an acceptance ratio of about 34.23%.

The editors would like to express their sincere appreciation and thanks to all the members of the SocialSec 2020 Organizing Committee and the Technical Program Committee for their tremendous efforts. Without their dedication, it would have been impossible to have a successful SocialSec 2020. The editors would also like to thank all the authors for their contributions. Finally, we express special thanks to Springer for publishing the proceedings of SocialSec 2020.

September 2020

Yang Xiang  
Zheli Liu  
Jin Li

# Organization

## General Chairs

Yang Xiang	Swinburne University of Technology, Australia
Zheli Liu	Nankai University, China
Jin Li	Guangzhou University, China

## Program Committee Chairs

Changyu Dong	Newcastle University, UK
Ding Wang	Nankai University, China
Xiaochun Cheng	Middlesex University, UK
Moayad Aloqaily	Analytics Inc., Canada

## Publication Chairs

Jian Xu	Northeastern University, China
Siu-Ming Yiu	Chinese University of Hong Kong, Hong Kong, China

## Publicity Chairs

Hao Peng	Zhejiang Normal University, China
Il-sun You	Soonchunhyang University, South Korea

## Local Chairs

Ying Zhang	Nankai University, China
Zhi Wang	Nankai University, China

## Webmaster

Yu Wang	Guangzhou University, China
---------	-----------------------------

## Program Committee

Aniello Castiglione	University of Naples Parthenope, Italy
Wun-She Yap	Universiti Tunku Abdul Rahman, Malaysia
Seonghan Shin	National Institute of Advanced Industrial Science and Technology (AIST), Japan
Steve Furnell	University of Plymouth, UK
Kwok Yan Lam	Nanyang Technological University, Singapore
Georgios Kambourakis	University of the Aegean, Greece

Julian Jang-Jaccard	Massey University, New Zealand
Willy Susilo	University of Wollongong, Australia
Muhammad Khurram Khan	King Saud University, Saudi Arabia
Rongmao Chen	National University of Defense Technology, China
Yizhi Ren	Hangzhou Dianzi University, China
Fuchun Guo	University of Wollongong, Australia
Jinguang Han	University of Surrey, UK
Genhard Hancke	City University of Hong Kong, HongKong, China
Romain Laborde	IRIT/SIERA, France
Kwok Yan Lam	Nanyang Technological University, Singapore
Yan Li	Singapore Management University, Singapore
Joseph Liu	Monash University, Australia
Kaitai Liang	University of Surrey, UK
Rongxing Lu	University of New Brunswick, Canada
RIshikesh Sahay	Technical University of Denmark, Denmark
Seonghan Shin	National Institute of Advanced Industrial Science and Technology, Japan



# Contents

## Big Data Security

Efficient and Secure Two-Party Distributed Signing Protocol for the GOST Signature Algorithm . . . . .	3
<i>Yunru Zhang, Min Luo, Kim-Kwang Raymond Choo, Li Li, and Debiao He</i>	
A Novel Web Anomaly Detection Approach Based on Semantic Structure. . .	20
<i>Zishuai Cheng, Baojiang Cui, and Junsong Fu</i>	
A Malicious URL Detection Model Based on Convolutional Neural Network . . . . .	34
<i>Zhiqiang Wang, Shuhao Li, Bingyan Wang, Xiaorui Ren, and Tao Yang</i>	
Security-Enhanced Timed-Release Encryption in the Random Oracle Model . . . . .	41
<i>Ke Yuan, Yahui Wang, Yingming Zeng, Wenlei Ouyang, Zheng Li, and Chunfu Jia</i>	
Generating Adversarial Malware Examples with API Semantics-Awareness for Black-Box Attacks . . . . .	52
<i>Xiaowei Peng, Hequn Xian, Qian Lu, and Xiuqing Lu</i>	
An Elastic Data Processing Method Based on Data-Center-Platform . . . . .	62
<i>Zhang Pan, Lai Fenggang, Du Jing, Ying Zhangchi, Kong Rui, Zhou Yi, and Yu Xiao</i>	
Adaptive Adversarial Attack on Graph Embedding via GAN . . . . .	72
<i>Jinyin Chen, Dunjie Zhang, and Xiang Lin</i>	
Research on LDoS Attack Detection and Defense Mechanism in Software Defined Networks . . . . .	85
<i>Shengxu Xie, Changyou Xing, Guomin Zhang, Xianglin Wei, and Guyu Hu</i>	
Unsupervised Analysis of Encrypted Video Traffic Based on Levenshtein Distance . . . . .	97
<i>Luming Yang, Yingming Zeng, Shaojing Fu, and Yuchuan Luo</i>	
An Efficient and Revocable Auditing Scheme for the Internet of Things . . . .	109
<i>Tian Jun-Feng and Guo Rui-Fang</i>	

A Configurable off-Policy Evaluation with Key State-Based Bias Constraints in AI Reinforcement Learning . . . . . 120  
*Shuoru Wang, Jiqiang Liu, Tong Chen, He Li, Wenjia Niu, Endong Tong, Long Li, and Minglu Song*

An Efficient Framework for Text Document Security and Privacy. . . . . 132  
*Umair Khadam, Muhammad Munwar Iqbal, Leonardo Mostarda, and Farhan Ullah*

**Social Networks**

A Real-Time Audio and Video Streaming Transmission Scheme for Social Media. . . . . 143  
*Jianping Yu, Gang Zhao, Xiaohui Kuang, and Ruyun Zhang*

UAV-Enabled Social Internet of Vehicles: Roles, Security Issues and Use Cases . . . . . 153  
*Chaogang Tang, Xianglin Wei, Chong Liu, Haifeng Jiang, Huaming Wu, and Qing Li*

An Efficient Influence Maximization Algorithm Based on Social Relationship Priority in Mobile Social Networks. . . . . 164  
*Xinxin Zhang, Li Xu, and Min Gao*

Key Nodes Recognition in Opportunistic Network . . . . . 178  
*Zhifei Wang, Gang Xu, Fengqi Wei, Zhihan Qi, and Liqiang He*

A Novel Measure to Quantify the Robustness of Social Network Under the Virus Attacks . . . . . 189  
*Bo Song, Zhengjun Jing, Y. Jay Guo, Ren Ping Liu, and Qian Zhou*

**Privacy-Preserving and Security**

Effectiveness Analysis of Traditional Chinese Medicine for Anti-Alzheimer’s Disease Based on Machine Learning . . . . . 203  
*Jingwen Lu, Peng Tang, Weidong Qiu, Hao Wang, and Jie Guo*

A Methodology of Fake Cell Test Based on the RRC Redirection or Reselection Priorities from the 5G Network . . . . . 215  
*Yanan Zhang, Chao Ma, Dong Wang, Tianyu Liu, and Zhi Wu*

Keeping Privacy Data Secure Under Factory Recovery . . . . . 224  
*Wang Lianfang, Wang Ye, Zhao Gang, Liu Lu, and Kuang Xiaohui*

An Intelligent File Transfer Optimization for Poor Network Conditions . . . . . 234  
*Ming Yan, Bo Zhang, Gang Zhao, Xiaohui Kuang, Lu Liu, and Ruyun Zhang*

A VirtualXposed-Based Inline Hooking Framework for Android Native Methods . . . . . 245  
*Shuo Feng, Yu-an Tan, Gang Zhao, Xiaohui Kuang, Xiao Yu, and Juan Wang*

An Android Data Protection Scheme for System-as-Root Architectures . . . . . 254  
*Kai Yang, Ling Pang, Bo Zhang, Gang Zhao, and Xiaohui Kuang*

A Two-Fitness Resource Scheduling Strategy Based on Improved Particle Swarm Optimization . . . . . 263  
*Xueming Qiao, Meng Chen, Xiangkun Zhang, Weiyi Zhu, Yanhong Liu, Zhixin Huo, Ruiqi Sun, and Dongjie Zhu*

Privacy-Preserving Nonlinear SVM Classifier Training Based on Blockchain . . . . . 278  
*Nan Jia, Shaojing Fu, and Ming Xu*

A Novel Defending Scheme for Graph-Based Classification Against Graph Structure Manipulating Attack . . . . . 289  
*Quanyu Zhao, Fengqian Zhang, and Yuan Zhang*

Privacy-Preserving Graph Operations for Social Network Analysis . . . . . 303  
*Peng Li, Fucai Zhou, Zifeng Xu, Yuxi Li, and Jian Xu*

The Movie Recommendation System Based on Differential Privacy . . . . . 318  
*Min Li, Yingming Zeng, Yue Guo, and Yun Guo*

Reliability Analysis of Heterogeneous CPS Under Different Swapping Inter-links Strategies . . . . . 329  
*Hao Peng, Can Liu, Dandan Zhao, Zhaolong Hu, Jianmin Han, and Jianfeng Lu*

Adapt Swarm Path Planning for UAV Based on Artificial Potential Field with Birds Intelligence Extensions. . . . . 340  
*Yifei He, Jiqiang Liu, Endong Tong, Wenjia Niu, Xinyu Huang, Ying Zhou, Chenyang Li, and Liang Chang*

Automatic Counting of Railway Tools Based on Deep Learning . . . . . 355  
*Wei Wei, Jin Yang, Sikai Wang, Deng Chen, Yanduo Zhang, Zihang Zhang, Wei Liu, Gonghao Duan, Chaohui Zheng, Jianping Ju, and Jianyin Tang*

Risk Assessment of Heterogeneous CPS Systems Under Different Proportions of Links . . . . . 369  
*Hao Peng, Zhe Kan, Dandan Zhao, Jianmin Han, and Zhaolong Hu*

Automatic Classification Analysis of Tibetan Folk Music Based on Adaboost Algorithm . . . . . 379  
*Ma Ying, Li Kaiyong, and Hou Jiayu*

On the Security of a Certificateless Public Verification Scheme for Cloud-Based Cyber-Physical-Social Systems . . . . . 387  
*Jing Wang, Hongjie Zhang, Lishong Shao, Li Li, and Min Luo*

Advertising Strategy for Maximizing Profit Using CrowdSensing Trajectory Data. . . . . 395  
*Kaihao Lou, Shuqiu Li, Funing Yang, and Xingliang Zhang*

A Method for Community Partition Based on Information Granularity. . . . . 407  
*Tianchu Hang, Yang Bai, and Guishi Deng*

Forward Calculation for Improving the Sensitivity of Multiple Perturbations in Magnetic Induction Tomography Based on Brain Tissue Structure. . . . . 420  
*Yi Lv*

Pair-Wise Convolution Network with Transformers for Sequential Recommendation . . . . . 433  
*Jiangpeng Shi, Xiaochun Cheng, and Jianfeng Wang*

**Author Index** . . . . . 447

# **Big Data Security**



# Efficient and Secure Two-Party Distributed Signing Protocol for the GOST Signature Algorithm

Yunru Zhang<sup>1,2</sup>, Min Luo<sup>2(✉)</sup>, Kim-Kwang Raymond Choo<sup>3</sup>, Li Li<sup>2</sup>,  
and Debiao He<sup>1,2</sup>

<sup>1</sup> Cyberspace Security Research Center, Peng Cheng Laboratory,  
Shenzhen 518055, China  
yunruzhang@qq.com, hedebiao@163.com

<sup>2</sup> School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China  
{mluo, lli}@whu.edu.cn

<sup>3</sup> Department of Information Systems and Cyber Security and the Department  
of Electrical and Computer Engineering, University of Texas at San Antonio,  
San Antonio, TX 78249, USA  
raymond.choo@fulbrightmail.org

**Abstract.** Mobile devices, such as Android and iOS devices, are often used for electronic/mobile commerce (e.g. payments using WeChat Pay and Bitcoin wallet). Hence, ensuring the security of a user's private key stored on the device is crucial. To reduce the risk of private key leakage, a number of  $(t, n)$  threshold secret sharing protocols have been proposed in the literature. However,  $(t, n)$  threshold secret sharing protocols are generally not designed to mitigate key reconstruction attacks. Hence, in this paper we propose an efficient and secure two-party distributed signing protocol for the GOST signature algorithm, a Russian cryptographic standard algorithm. This allows us to separate a single private key to two mobile devices and generate a valid signature without reconstructing the entire private key. We then prove that our protocol is secure under non-standard assumption. Moreover, we implement our protocol using MIRACL Cryptographic SDK and evaluate its performance on two Android devices and two personal computers.

**Keywords:** GOST signature · Two-party signing · Wireless environment · Provable secure

## 1 Introduction

Electronic commerce using mobile devices (also known as mobile commerce) is increasingly commonplace. For example, according to the statistics from [18], the global mobile payment income is reportedly 450 billion dollars in 2015 and it is expected to exceed 1 trillion dollars in 2019. From October to December 2016, mobile e-commerce spending in U.S. is approximately 22.7 billion dollars.

Digital signature plays a key role in wireless communication [5, 11, 13, 20], as it can facilitate the validation of user’s identity and message and to achieve non repudiation. However, this necessitates the protection of the user’s private key. Generally, the user’s private key is stored in a single device such as an Android or iOS device, or a hardware such as a smart card. In other words, if an attacker gets access to this single device then the attacker can potentially gain access to the private key, for example by exploiting software/hardware vulnerabilities or using digital forensic techniques.

$(t, n)$  threshold secret sharing protocol [1–3, 6, 7, 17, 19] can be used to protect the security of the private key. Specifically, in a  $(t, n)$  threshold secret sharing protocol, the private key is split into  $n$  parties, where the private key can only be reconstructed by at least  $t$  participants. This raises the difficulty of an attacker in seeking to obtain the private key, as the attacker now has to corrupt  $t$  or more parties (or devices).

There is, however, a limitation in such  $(t, n)$  secret sharing protocols, in the sense that the private key needs to be reconstructed in its entirety in order for the group to successfully decrypt or sign a message. Once a party who holds the private key colludes with an adversary (e.g. compromised by the attacker), the private key is exposed to the adversary. While a number of  $(t, n)$  secret sharing protocols [4, 15] can decrypt or sign a message without the need to reconstruct the entire private key, the computing cost is significant; therefore such schemes are not suitable for the wireless environment. To mitigate this limitation, a number of two-party signing protocols have also been designed. However, these protocols incur significant computational costs.

Inspired by Lindell’s fast secure two-party ECDSA signing protocol (presented in *CRYPTO 2017*) [10], we construct an efficient and secure two-party distributed signing protocol for the GOST signature algorithm. GOST [9] is a set of technical standards maintained by the Euro-Asian Council for Standardization, Metrology and Certification (EASC). As part of its national standardization strategy, the GOST standard was originally developed by the Soviet government. Currently, the collection of GOST standards includes over 20,000 titles used extensively in conformity assessment activities in 12 or more countries<sup>1</sup>.

In this paper, we present a two-party distributed signing protocol for the GOST signature algorithm. At the time of this research, this is the first fast threshold cryptography protocol designed for GOST signature algorithm. Our protocol can generate a valid signature without the need to recover the entire private key. In addition, a valid signature cannot be generated if one of the participators is not involved.

In order to be assured of its security, we combine the zero-knowledge proof analysis and demonstrate the protocol’s security under non-standard assumption.

---

<sup>1</sup> <https://delta-engineering.be/gost>, last accessed July 24th, 2018.

We also implement the protocol using the MIRACL cryptographic SDK on two Android devices and two personal computers (PCs), and demonstrate that our protocol is more secure, efficient and practical in the wireless environment. Specifically, in terms of performance, the findings from the evaluation show that our protocol achieves better performance, and the signature operation on its BN curve consumes approximately 152.34 ms when running on a PC.

The remainder of the paper is organized as follows. We describe the relevant notations, GOST signature algorithm, zero-knowledge proof and the homomorphism of Paillier Encryption in Sect. 2. We present the proposed protocol, and describe its construction in Sect. 3. We analyze the security of the protocol under non-standard assumption combined with the zero-knowledge proof analysis in Sect. 4. And in Sect. 5, we present our evaluation findings. Finally, we conclude this paper in the last section.

## 2 Preliminaries

In this paper, let  $n$  and  $\mu(\cdot)$  respectively denote the security parameter and a negligible function for any polynomial  $p$ , having  $\mu(n) = O(1/p(n))$ . P.P.T denotes a probabilistic-polynomial time algorithm, and  $H$  and  $h$  are two secure hash functions, such that  $H : \{0, 1\}^* \rightarrow \{0, 1\}^{256}$ , and  $h : \{0, 1\}^* \rightarrow \mathbb{Z}_n$ .

### 2.1 Zero-Knowledge Proof

Let  $\mathcal{F}_{zk}$  denotes the standard ideal zero knowledge functionality, and  $\mathcal{F}_{zk} : ((\chi, \omega), \vartheta) \rightarrow (\vartheta, (\chi, R(\chi, \omega)))$ , in which  $\vartheta$  means the empty string.

**Definition 1.** Defines  $\mathcal{F}_{zk}^R$  for relation  $R$ : Upon receiving (prove,  $sid, \chi, \omega$ ) from  $P_i$  ( $i \in \{1, 2\}$ ), checks whether  $sid$  has been used before or  $(\chi, \omega) \notin R$ . If yes, then stops the function. Otherwise, sends (proof,  $sid, \chi$ ) to  $P_{3-i}$ . The non-interactive zero-knowledge proof of knowledge satisfying  $\mathcal{F}_{zk}$  [8] can be achieved under the random oracle model.

In our protocol, we use the following zero-knowledge functionalities:  $\mathcal{F}_{zk}^{RP}$ ,  $\mathcal{F}_{zk}^{RDL}$ ,  $\mathcal{F}_{zk}^{RPDL}$ . The three functionalities are based on the following relations:

1. The key of Paillier public encryption is correctly generated, and the relationship is defined such that

$$R_p = \{(N, (p_1, p_2)) | N = p_1 p_2 \text{ and } p_1, p_2 \text{ are prime}\}$$

2. The Elliptic curve discrete logarithm problem is defined such that

$$R_{DL} = \{(G_1, n, P, r | P = r \cdot G)\}$$

We use Schnorr zero-knowledge proof [16] in our protocol.



3. The ciphertext is a Paillier encryption of a discrete log, such that:

$$R_{PDL} = \{((C, ppk, Q_1, \mathbb{G}, G, n), (k_1, psk)) \mid k_1 = Dec_{psk}(C) \text{ and } Q_1 = k_1 \cdot G \text{ and } k_1 \in \mathbb{Z}_n\},$$

where  $(ppk, psk)$  is a Paillier public and private key pair.

## 2.2 Paillier Encryption

In our protocol, we use Paillier cryptosystem [14] for encryption. Paillier cryptosystem is defined as follows:

### 1. Key Generation:

- (a) Randomly chooses two equivalent length large prime numbers  $p$  and  $q$ .
- (b) Computes  $g = pq + 1$ , let  $\alpha = \phi(n)$  and  $\beta = (\alpha)^{-1}$ , where  $\phi(n) = (p-1)(q-1)$ .
- (c) Let  $ppk = (n, g)$  and  $psk = (\alpha, \beta)$ , where  $ppk$  and  $psk$  are public and private key pair.

### 2. Encryption:

- (a) Randomly selects a number  $r$ , in which  $r \in [0, n]$ .
- (b) Computes ciphertext  $c = Enc_{ppk}(m) = g^m \cdot r^n \pmod{n^2}$ , where  $0 \leq m < n$ .

### 3. Decryption:

- (a) Decrypts ciphertext  $m = Dec_{psk}(c) = L(c^\alpha \pmod{n^2}) \cdot \beta \pmod{n}$ , where  $L(x) = \frac{x-1}{n}$ .

In our protocol,  $Enc_{ppk}(\cdot)$  denotes the encrypt operation using public key  $pk$ , and  $Dec_{psk}(\cdot)$  denotes the decrypt operation using private key  $sk$ . In the Paillier cryptosystem, there is a notable feature which is its homomorphic property:

1.  $Dec_{psk}(Enc_{ppk}(m_1) \cdot Enc_{ppk}(m_2)) = m_1 + m_2$ .
2.  $Dec_{psk}(Enc_{ppk}(m_1)^{m_2}) = m_1 m_2$ .

Let  $c_1 = Enc_{ppk}(m_1)$ ,  $c_2 = Enc_{ppk}(m_2)$ , then  $c_1 \oplus c_2 = Enc_{ppk}(m_1 + m_2)$ ,  $m_2 \otimes c_1 = Enc_{ppk}(m_1)^{m_2}$ .

## 2.3 Network Model

Following the definition of our distributed signature generation protocol, the network model of our protocol is shown in Fig. 1. There are an administrator  $Adm$ , and the user's devices  $P_i$  in the protocol.

- $Adm$ : It is a trusted third party, which generates the system parameters.
- $P_i$ : It interacts with another one, uses the system parameters to generate public key, and finally outputs the signature.

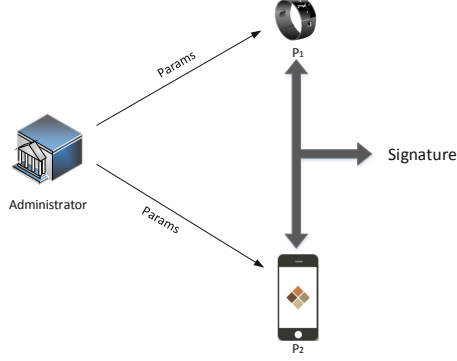


Fig. 1. Network model

### 3 Proposed Two-Party Distributed GOST Signing Protocol

We present our two-party distributed GOST signing protocol in this section. In our protocol, there are two main phases: distributed key generation (see Sect. 3.1), and distributed signature generation (see Sect. 3.2).

#### 3.1 Distributed Key Generation

In the distributed key generation phase, two parties  $P_1$  and  $P_2$  interact with each other to generate the public key  $Q$ . The steps are described as follows.

##### Phase 1: Distributed Key Generation

1.  $P_1$  randomly chooses a number  $d_1$ , computes  $Q_1 = d_1P$ ,  $C_{key} = Enc_{pk}(d_1)$ , and sends (prove, 1,  $(Q_1, C_{key}), (d_1, sk)$ ) to  $\mathcal{F}_{zk}^{RPDL}$ . Finally,  $P_1$  generates a Paillier public and private key pair  $(pk, sk)$ , and sends (prove, 1,  $N, (p_1, p_2)$ ) to  $\mathcal{F}_{zk}^{RP}$ , where  $pk = N = p_1p_2$ .
2. If  $P_2$  receives (proof, 1,  $(Q_1, C_{key})$ ) from  $\mathcal{F}_{zk}^{RPDL}$  and (proof, 1,  $N$ ) from  $\mathcal{F}_{zk}^{RP}$ , then it performs the following steps:
  - (a) Computes  $Q_2 = d_2P$  where  $d_2$  is a random integer, and sends (prove, 2,  $Q_2, d_2$ ) to  $\mathcal{F}_{zk}^{RDL}$ .
  - (b) Computes  $Q = d_2Q_1$ , and stores  $(d_2, Q, pk, C_{key})$ . Otherwise, it aborts.
3.  $P_1$  computes  $Q = d_1Q_2$  and stores  $(d_1, Q, pk, sk)$ .

It is trivial to check that  $Q = d_1Q_2 = d_2Q_1 = d_1d_2P$ .

#### 3.2 Distributed Signature Generation

In the distributed signature generation phase,  $P_1$  and  $P_2$  interact with each other to generate a GOST signature using their partial secret keys, which are generated in the preceding phase. The phase is described as follows.

## Phase 2: Distributed Signature Generation

1.  $P_1$  chooses  $k_1 \in \mathbb{Z}_q^*$ , computes  $R_1 = k_1P$ , and encrypts  $k_1$  under Paillier encryption key that  $C_{ran} = \text{Enc}_{pk}(k_1)$ . Then,  $P_1$  sends  $(\text{prove}, 1, (R_1, C_{ran}), (k_1, sk))$  to  $\mathcal{F}_{zk}^{RPDL}$ .
2. If  $P_2$  receives  $(\text{proof}, 1, (R_1, C_{ran}))$  from  $\mathcal{F}_{zk}^{RPDL}$ , then it executes the following steps; otherwise, it aborts.  $P_2$  chooses  $k_2 \in \mathbb{Z}_q^*$  randomly, computes  $R_2 = k_2P$ , and sends  $(\text{prove}, 2, R_2, k_2)$  to  $\mathcal{F}_{zk}^{RPDL}$ . Then,  $P_2$  computes  $R_2 = k_2P$  and  $r = C_x \bmod q$  in which  $C_x$  is the x-coordinate of  $R_2$ , due to the homomorphism property of the Paillier Encryption,  $P_2$  can compute  $C_1 = rd_2 \otimes C_{key} \oplus ek_2 \otimes C_{ran} \oplus \text{Enc}_{pk}(\rho \cdot q)$ , where  $\rho \in \mathbb{Z}_q$  is chosen randomly. Finally,  $P_2$  sends  $C_1$  to  $P_1$ .
3. If  $P_1$  receives  $(\text{proof}, 2, R_2)$  from  $\mathcal{F}_{zk}^{RPDL}$ , then it executes the following steps; otherwise, it aborts.  $P_1$  computes  $C = k_1R_2 = k_1k_2P$ , and  $r = C_x \bmod q$ . Then,  $P_1$  decrypts  $C_1$  using its private key:  $s = \text{Dec}_{sk}(C_1) \bmod q = rd_1d_2 + ek_1k_2$ . Finally,  $P_1$  verifies signature  $(r, s)$  using the public key  $Q = d_1Q_2$ . If the signature is valid, then outputs  $(r, s)$

**Correctness.** Due to  $C_{key} = \text{Enc}_{pk}(d_1)$ ,  $C_{ran} = \text{Enc}_{pk}(k_1)$ , we can compute that  $C_1 = \text{Enc}_{pk}(rd_1d_2 + ek_1k_2)$ . Thus, we have the following equations:

$$\begin{aligned} r &= C_x \bmod q \\ s &= \text{Dec}_{sk}(C_1) \\ &= (rd_1d_2 + ek_1k_2) \end{aligned}$$

Therefore, the correctness of the proposed signing protocol is demonstrated.

## 4 Security Analysis

### 4.1 Security Model

**Definition 2.** Let  $\pi$  be a secure digital signature protocol  $\pi = (\text{Gen}, \text{Sign}, \text{Verify})$ , we define an experiment  $\text{Sign}_{\mathcal{A}, \pi}(1^n)$  as follows:

1.  $(pk, sk) \leftarrow \text{Gen}(1^n)$ .
2.  $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\text{Sign}_{sk}(\cdot)}(1^n, pk)$ .
3. The set of all  $m$  that  $\mathcal{A}$  queries to its oracle is  $\mathcal{M}$ . An adversary  $\mathcal{A}$  can query oracle with  $m$ . Then, the experiment outputs 1, when  $m^* \notin \mathcal{M}$  and  $\text{Verify}_{pk}(m^*, \sigma^*) = \text{valid}$ .

**Definition 3.** We define the signature protocol  $\pi$  is existentially unforgeable under chosen message attack, if for any P.P.T adversary  $\mathcal{A}$  there exists a negligible function  $\mu(\cdot)$  for every  $n$ ,

$$\Pr[\text{Sign}_{\mathcal{A}, \pi}(1^n) = 1] \leq \mu(n)$$

We consider an experiment  $\text{DistSign}_{\mathcal{A},\Pi}^b(1^\lambda)$  in distributed signing protocol. An adversary  $\mathcal{A}$  can control a party  $P_b$  ( $b \in \{1, 2\}$ ).  $\Pi_b(\cdot, \cdot)$  is a stable oracle which executes the instructions from the honest party  $P_{3-b}$ . In this definition, distributed key generation phase should run first, then distributed signature generation phase can be executed between the two parties.  $\mathcal{A}$  queries the oracle with two inputs: a session identifier ( $sid$ ) and a next incoming message. It works as follows:

1. When  $P_{3-b}$  receives a query  $(0, 0)$  from  $P_b$  for the first time, party  $P_{3-b}$  instructs the oracle to initialize a machine  $M$ . If  $P_{3-b}$  sends the first message during distributed key generation phase, oracle will reply with this message.
2. When receiving a query  $(0, m)$ , oracle sends  $m$  to  $M$  as the next incoming message and return the output of  $M$  if the public and private keys have not been generated. Otherwise, oracle return  $\perp$ .
3. When receiving a query  $(sid, m)$  and  $sid \neq 0$ , but  $M$  has not completed generating the public and private keys, oracle return  $\perp$ .
4. When receiving a query  $(sid, m)$  and the distributed key generation has executed and the  $sid$  has not been queried, then party  $P_{3-b}$  instruct the oracle to invoke a new machine  $M_{sid}$  with  $(sid, m)$ . If party  $P_{3-b}$  sends the first message in the distributed key generation phase, then the oracle reply with this message.
5. When receiving a query  $(sid, m)$  and the distributed key generation has executed and the  $sid$  has been queried already, then the oracle sends  $m$  to  $M_{sid}$  as the incoming message and returns  $M_{sid}$ 's output as the next message. If  $M_{sid}$  completes, then return  $M_{sid}$ 's output.

The adversary  $\mathcal{A}$  can control a party  $P_b$  ( $b \in \{0, 1\}$ ) with the oracle access to  $\Pi_b$  in this experiment.  $\mathcal{A}$  can win the game if the message used to the forged signature is not be queried.

**Definition 4.** We define an experiment  $\text{DistSign}_{\mathcal{A},\Pi}^b(1^\lambda)$  as follows:

1.  $(m^*, \sigma^*) \leftarrow \mathcal{A}^{(\Pi_b(\cdot, \cdot))}(1^\lambda)$ .
2. We define  $\mathcal{M}$  is the collection of any  $m$  which can be queried.  $\mathcal{A}$  can query oracle with  $(sid, m)$ . Then, the experiment outputs 1 when  $m^* \notin \mathcal{M}$  and  $\text{Verify}_{pk}(m^*, \sigma^*) = \text{valid}$ , where  $\text{Verify}_{pk}$  is an algorithm of  $\pi = (\text{Gen}, \text{Sign}, \text{Verify})$ , and the verification key is output by  $P_{3-b}$  in the distributed key generation phase.

**Definition 5.** We define the signature protocol  $\Pi$  is a secure two-party protocol for distributed signature generation for  $\pi$  only if for any probabilistic polynomial time adversary  $\mathcal{A}$  and every  $b \in \{0, 1\}$ , there exists a negligible function  $\mu(\cdot)$  such that for every  $n$ :

$$\Pr[\text{DistSign}_{\mathcal{A},\Pi}^b(1^n) = 1] \leq \mu(n)$$

**Definition 6.** A function  $\mathcal{F}_{GOST}$  consists of two functions: one is key generation and the other is signature generation. The key generation function can be

queried only once, after key generation function has been executed, the signature generation function can be queried arbitrary times.  $P_1$  and  $P_2$  execute the function  $\mathcal{F}_{GOST}$  as follows:

1. When receiving  $KeyGen$  from both  $P_1$  and  $P_2$ :
  - (a) Generate a GOST key pair  $(d, Q)$ . Randomly choose a number  $d \in \mathbb{Z}_q^*$ , and compute  $Q = d \cdot G$ . Choose a hash function  $H_q : \{0, 1\}^n \rightarrow \{0, 1\}^{\log|q|}$ , then store  $(q, H_q, d)$ .
  - (b) Send  $(Q, H_q)$  to  $P_1$  and  $P_2$ .
  - (c)  $KeyGen$  will no longer be queried.
2. When oracle receives  $Sign(sid, m)$  from  $P_1$  and  $P_2$ , if  $KeyGen$  has queried already and  $sid$  has not been queried, the adversary computes an GOST signature  $(r, s)$  using message  $m$ . The signature is generated as follows:
  - (a) Compute  $H = H_q(m)$ , and compute an integer  $\alpha$  of the vector  $H$ , set  $e = \alpha \bmod q$ .
  - (b) Choose  $k \in \mathbb{Z}_q^*$  randomly, compute  $C = k \cdot P = (C_x, C_y)$  and  $r = C_x \bmod q$ .
  - (c) Compute  $s = (rd + ke) \bmod q$ .
  - (d) Output  $(r, s)$ .

**Definition 7.** Assumption that there exists a negligible function  $\mu(\cdot)$  satisfying  $\Pr[\text{Paillier} - EC_{\mathcal{A}}(1^n) = 1] \leq \frac{1}{2} + \mu(n)$ , we say for any P.P.T adversary  $\mathcal{A}$ , it is negligible for them to solve the Paillier-EC assumption. Suppose  $G$  is a generator of a group  $\mathbb{G}$ , and the order is  $q$ ,  $\text{Paillier} - EC_{\mathcal{A}}(1^n)$  is defined as follows:

1. Generate a Paillier public and private key pair  $(pk, sk)$ .
2. Select  $r_0, r_1 \xleftarrow{r} \mathbb{Z}_q$  and compute  $R = r_0 \cdot G$ .
3. Select  $b \in \{0, 1\}$  and encrypt the random number  $r_b$ ,  $C = \text{Enc}_{pk}(r_b)$ .
4. Let  $b' = \mathcal{A}^{\mathcal{O}_C(\cdot, \cdot)}$ , if  $\text{Dec}_{sk}(C') = \alpha + \beta \cdot r_b \bmod q$ ,  $\mathcal{O}_C(C', \alpha, \beta) = 1$ .
5. When  $b' = b$ , the experiment output 1.

## 4.2 Proof of Security

We prove the protocol  $\Pi$  of our proposed distributed GOST signature generation protocol is a secure two-party signing protocol.

**Theorem 1.** *If Paillier encryption is indistinguishable under CPA, and GOST signature is existentially-unforgeable under chosen message attack, then our two-party signing protocol of GOST is secure.*

*Proof.* In our proposed protocol, if  $\mathcal{A}$  can break the protocol in zero-knowledge model with the probability  $\epsilon$ , then the protocol can be broken with probability  $\epsilon \pm \mu(n)$  where  $\mu(\cdot)$  is a negligible function.

We prove the security of corrupted  $P_1$  and corrupted  $P_2$  separately. When  $\mathcal{A}$  attacks our protocol, we create a simulator  $\mathcal{S}$ , the probability that he forges the GOST signature in Definition 2 is similar to the probability that  $\mathcal{A}$  forges the signature in Definition 4. We now prove that for every P.P.T  $\mathcal{A}$  and

$b \in \{0, 1\}$ , there exists a P.P.T  $\mathcal{S}$  and a negligible function  $\mu(\cdot)$  can make the Paillier encryption is IND-CCA secure, it has:

$$|\Pr[\text{Sign}_{\mathcal{S}, \pi}(1^n) = 1] - \Pr[\text{DistSign}_{\mathcal{A}, \Pi}^b(1^n) = 1]| \leq \mu(n) \quad (1)$$

In Eq. 1,  $\Pi$  denotes our proposed protocol,  $\pi$  denotes the GOST signature protocol. If GOST signature protocol is secure, there exists a negligible function  $\mu'(\cdot)$  for every  $n$  that  $\Pr[\text{Sign}_{\mathcal{S}, \pi}(1^n) = 1] \leq \mu'(n)$ . According to the Eq. 1, we have that  $\Pr[\text{Sign}_{\mathcal{A}, \pi}^b(1^n) = 1] \leq \mu(n) + \mu'(n)$ . Now we prove Eq. 1 from two aspects:  $b = 1$  and  $b = 2$ .

When  $b = 1$ , the adversary  $\mathcal{A}$  corrupted party  $P_1$ , we define  $\mathcal{A}$  as a P.P.T adversary in  $\text{DistSign}_{\mathcal{A}, \Pi}^1(n)$ , we create a P.P.T simulator  $\mathcal{S}$  in  $\text{Sign}_{\mathcal{S}, \pi}(n)$ .  $\mathcal{S}$  simulates the execution for  $\mathcal{A}$  as follows:

1. In  $\text{Sign}$ ,  $\mathcal{S}$  receives  $(1^n, Q)$ , in which  $Q$  is user's public key.
2.  $\mathcal{S}$  inputs  $1^n$  and calls  $\mathcal{A}$ , and simulates oracle  $\Pi$  for  $\mathcal{A}$  in  $\text{DistSign}$  as follows:
  - (a) When  $\mathcal{A}$  queries all  $(\text{sid}, \cdot)$  to  $\Pi$  before the public and private key pair has been generated,  $\mathcal{S}$  output  $\perp$ . If  $\mathcal{A}$  queries anything before it has queried  $(0, 0)$ ,  $\mathcal{S}$  output  $\perp$ .
  - (b) If  $\mathcal{A}$  had queried  $(0, 0)$  to  $\Pi$ ,  $\mathcal{S}$  will receive  $(0, m_1)$  from  $P_1$  in the distributed key generation phase, the process of  $\mathcal{S}$  replies  $\mathcal{A}$  as follows:
    - i.  $\mathcal{S}$  parses  $m_1$  as  $(\text{prove}, 1, (Q_1, C_{key}), d_1)$  which is the message that  $P_1$  sends to  $\mathcal{F}_{zk}^{RPDL}$ .
    - ii.  $\mathcal{S}$  verifies the equation  $Q_1 = d_1 \cdot P$  and  $C_{key} = \text{Enc}_{pk}(d_1)$ . If the equations hold, it computes  $Q_2 = (d_1)^{-1} \cdot Q$ , otherwise it aborts.
    - iii.  $\mathcal{S}$  sets the oracle's output that  $(\text{prove}, 2, Q_2)$  and sends it to the adversary  $\mathcal{A}$ .
  - (c)  $\mathcal{S}$  receives  $(0, m_2)$  and works as follows:
    - i.  $\mathcal{S}$  parses  $m_2$  as  $(\text{prove}, 1, N, (p_1, p_2))$  which is the message that  $\mathcal{A}$  sends to  $\mathcal{F}_{zk}^{RP}$ .
    - ii.  $\mathcal{S}$  verifies the equation  $pk = N = p_1 \cdot p_2$ , if the equation does not hold,  $\mathcal{S}$  simulates  $P_2$  to abort.
    - iii. The experiment completes if  $\mathcal{S}$  has simulated  $P_2$  to abort, and  $P_2$  quits this protocol. Due to  $P_2$  does not output the verification key  $pk$ , then  $\mathcal{S}$  does not output anything. Otherwise,  $\mathcal{S}$  stores  $(d_1, Q, sk, pk)$  and the distributed key generation is completed.
  - (d) When receiving a query  $(\text{sid}, m)$  and  $\text{sid}$  has not been queried before.  $\mathcal{S}$  queries its signing oracle with message  $m$ , then it receives a signature  $(r, s)$  from the signing oracle.  $\mathcal{S}$  can compute  $C$  in the verification algorithm of GOST signature.  $\mathcal{S}$  works as follows when receiving queries from  $\mathcal{A}$ :
    - i. The first message  $(\text{sid}, m_1)$  is parsed as  $(\text{prove}, 1, (R_1, C_{ran}), k_1, sk)$ . If  $R_1 = k_1 \cdot P$  and  $C_{ran} = \text{Enc}_{pk}(k_1)$ , then  $\mathcal{S}$  sets  $R_2 = (k_1)^{-1} \cdot C$ , and sets the oracle reply with  $(\text{prove}, 2, R_2)$  to  $\mathcal{A}$ . Otherwise,  $\mathcal{S}$  simulates  $P_2$  to abort.
    - ii.  $\mathcal{S}$  randomly chooses  $\rho \in \mathbb{Z}_q$ , and computes the ciphertext  $C_1 = \text{Enc}_{pk}(s + \rho \cdot q)$ , where  $s$  is the value of signature output from  $\mathcal{F}_{GOST}$ , and sets the oracle reply  $C_1$  to  $\mathcal{A}$ .
3. Once  $\mathcal{A}$  halts and outputs  $(m^*, \sigma^*)$ ,  $\mathcal{S}$  outputs  $(m^*, \sigma^*)$  and halts.

Now we prove the Eq. 1 holds. We show that  $\mathcal{A}$ 's view in  $\mathcal{S}$ 's simulation is identical to its view in the real execution of the proposed protocol.

In the distributed key generation phase, the generation of  $Q_2$  is the only difference between the real execution and the simulation. In the real execution,  $P_2$  chooses  $d_2 \in \mathbb{Z}_q^*$  and computes  $Q_2 = q_2 \cdot P$ . In the simulation,  $\mathcal{S}$  computes  $Q_2 = d_1^{-1} \cdot Q$ , where  $\mathcal{S}$  received the public key  $Q$  in the experiment *Sign*. Due to  $Q$  is chosen randomly, therefore, the distributions of  $x_2 \cdot P$  and  $x_1^{-1} \cdot Q$  are identical. If  $P_2$  does not abort, in both real execution and the simulation, the verification key is  $d_1 \cdot P_2 = Q$ . Therefore,  $\mathcal{A}$ 's view of the real execution and the simulation is identical, and  $Q$  is the public verification key.

In the distributed signature generation phase, the only difference between the real execution and the simulation of  $\mathcal{A}$ 's view is the generation of  $C_1$ .  $C$  is generated by  $\mathcal{F}_{GOST}$ , so that the distribution between  $k_1^{-1} \cdot C$  and  $k_2 \cdot P$  is identical. Therefore, the only difference is  $C_1$ , in the simulation, it is a ciphertext of  $s + \rho \cdot q$ , in the real execution, it is the ciphertext of  $s = rd_1d_2 + ek_1k_2 + \rho \cdot q$ .

We prove that the difference of  $\mathcal{A}$ 's view between the real execution and the simulation is indistinguishable. In the GOST signature protocol,  $s = rd + ek \pmod q = r(d_1d_2) + e(k_1k_2) \pmod q$ . Therefore, there exists some  $l \in \mathbb{Z}_q$  such that  $r(d_1d_2) + e(k_1k_2) = s + l \cdot q$ . Therefore, the difference between the real execution and the simulation is:

1. Real execution: the plaintext of  $C_1$  is  $s \pmod{q + l \cdot q + \rho \cdot q}$ .
2. Simulation: the plaintext of  $C_1$  is  $s \pmod{n + \rho \cdot q}$ .

Due to the distribution of  $C_1$  in the real execution and the simulation is statistically close. Thus, we prove that the Eq. 1 holds for  $b = 1$ .

When  $b = 2$  i.e. the adversary  $\mathcal{A}$  corrupted  $P_2$ . We construct  $\mathcal{S}$  that  $\mathcal{S}$  simulates  $P_1$  aborting at some random point. Let  $\mathcal{S}$  choose  $i \in \{1, 2, \dots, p(n)+1\}$  randomly, where  $p(n)$  is the upper bound of the query times made by  $\mathcal{A}$ . Due to probability of  $\mathcal{S}$  chooses the correct  $i$  is  $\frac{1}{p(n)+1}$ , so the probability that  $\mathcal{S}$  simulates  $\mathcal{A}$ 's view is  $\frac{1}{p(n)+1}$ . Therefore, the probability of  $\mathcal{S}$  forges a signature in *Sign* is at least  $\frac{1}{p(n)+1}$  times of the probability that  $\mathcal{A}$  forges a signature in *DistSign*.

Let  $\mathcal{A}$  be a P.P.T adversary,  $\mathcal{S}$  works as follows:

1. In the *Sign* experiment,  $\mathcal{S}$  receives  $(1^n, Q)$ , where  $Q$  is the GOST public key.
2.  $p(n)$  denotes the upper bound of the query times that  $\mathcal{A}$  queries to  $\Pi$  in *DistSign*,  $\mathcal{S}$  chooses  $i \in \{1, 2, \dots, p(n) + 1\}$ .
3.  $\mathcal{S}$  invokes  $\mathcal{A}$  on input  $1^n$ , then simulates the oracle  $\Pi$  in *DistSign*:
  - (a) When  $\mathcal{A}$  queries all  $(sid, \cdot)$  to  $\Pi$  before the completion of the distributed key generation phase,  $\mathcal{S}$  output  $\perp$ . If  $\mathcal{A}$  queries anything before it queries  $(0, 0)$ ,  $\mathcal{S}$  output  $\perp$ .
  - (b) After  $\mathcal{A}$  queries  $(0, 0)$  to  $\Pi$ ,  $\mathcal{S}$  generates a valid Paillier encryption key-pair  $(pk, sk)$  and computes the oracle reply with  $(\text{proof}, 1, N)$ .
  - (c)  $\mathcal{S}$  receives the message  $(0, m_1)$  and works as follows:
    - i.  $\mathcal{S}$  parses  $m_1$  as  $(\text{prove}, 2, Q_2, x_2)$  which is the message that  $P_2$  sends to  $\mathcal{F}_{zk}^{RPDL}$ .

- ii.  $\mathcal{S}$  verifies that  $Q_2 = d_2 \cdot P$ , if not, it simulates  $P_1$  to abort.
- iii.  $\mathcal{S}$  sets  $(\text{proof}, 1, (Q_1, C_{key}))$  as the oracle's response, where  $Q_1 = d_2^{-1} \cdot Q$ .

$\mathcal{S}$  stores  $(d_2, Q)$ , and distributed key generation phase is completed.

- (d) When receiving a query  $(sid, m)$ , where  $sid$  has not been queried before.  $\mathcal{S}$  sets the oracle reply with  $(\text{proof}, 1, R_1, C_{ran})$ , where  $R_1 = k_2^{-1} \cdot C$ .  $\mathcal{S}$  queries its signing oracle with message  $m$  in experiment  $Sign$ , then it receives a signature  $(r, s)$ .  $R$  can be computed in the verification algorithm of GOST signature.  $\mathcal{A}$  queries  $\mathcal{S}$  with identifier  $sid$ , and works as follows:
  - i. The first message  $(sid, m_1)$  is parsed as  $(\text{prove}, 2, R_2, k_2)$ ,  $\mathcal{S}$  checks the equation  $R_2 = k_2 \cdot P$ , if the equation does not hold, it simulates  $P_1$  aborts.
  - ii. The second message  $(sid, m_2)$  is parsed  $m_2$  as  $C_1$ . If this is the  $i$ th query by  $\mathcal{A}$ , then  $\mathcal{S}$  simulates  $P_1$  aborts. Otherwise, it continues.
- 4. Once  $\mathcal{A}$  halts and outputs  $(m^*, \sigma^*)$ ,  $\mathcal{S}$  outputs  $(m^*, \sigma^*)$  and halts.

Let  $j$  be the first query to oracle  $\Pi$  with  $(sid, m_2)$ , and  $P_1$  does not receive the valid that corresponds to the public key  $Q$ . If the equation  $j = i$  holds, the difference between the distribution of  $\mathcal{A}$ 's view in real execution and the simulation is the ciphertext  $C_{ran}$ . Since  $\mathcal{S}$  does not hold the Paillier private key in the simulation, the indistinguishability of the simulation follows from a reduction of indistinguishability of the encryption protocol under CPA.

We can learn that

$$|\Pr[Sign_{\mathcal{S}, \pi}(1^n) = 1 | i = j] - \Pr[DistSign_{\mathcal{A}, \Pi}^2(1^n) = 1]| \leq \mu(n)$$

so

$$\Pr[DistSign_{\mathcal{A}, \Pi}^2(1^n) = 1] \leq \frac{\Pr[Sign_{\mathcal{S}, \pi}(1^n) = 1]}{1/(p(n) + 1)} + \mu(n)$$

i.e.

$$\Pr[Sign_{\mathcal{S}, \pi}(1^n) = 1] \geq \frac{DistSign_{\mathcal{A}, \Pi}^2(1^n) = 1}{1/(p(n) + 1)} - \mu(n)$$

It means that if  $\mathcal{A}$  can forge a signature in  $DistSign_{\mathcal{A}, \Pi}^2(1^n)$  with a non-negligible probability, then  $\mathcal{S}$  can forge a signature in  $Sign_{\mathcal{S}, \pi}(1^n)$  with a non-negligible probability. According to that the GOST signature is existentially unforgeable, then our protocol is secure.  $\square$

**Theorem 2.** *If the Pailler-EC assumption is hard. Then, distributed signature generation phase securely computes  $\mathcal{F}_{GOST}$  in presence of a malicious static adversary.*

*Proof.* We prove the security for corrupted  $P_1$  and corrupted  $P_2$  respectively.

First, we consider  $P_1$  is corrupted by an adversary  $\mathcal{A}$ , we construct a simulator  $\mathcal{S}$ , and analysis the simulation of distributed key generation phase and distributed signature generation phase respectively.



In the distributed key generation phase, it works as follows:

1. When receiving  $KeyGen$ ,  $\mathcal{S}$  sends  $KeyGen$  to  $\mathcal{F}_{GOST}$  and receives  $Q$ .
2.  $\mathcal{S}$  invokes  $\mathcal{A}$  when receiving  $KeyGen$  and receives  $(\text{prove}, 1, (Q_1, C_{key}), (d_1, sk))$  from  $\mathcal{A}$ .
3.  $\mathcal{S}$  verifies the equations  $Q_1 = d_1 \cdot P$  and  $C_{key} = \text{Enc}_{pk}(d_1)$ . If the equations hold, it computes  $Q_2 = d_2 \cdot P$ , otherwise it aborts.
4.  $\mathcal{S}$  sends  $(\text{proof}, 2, Q_2)$  to  $\mathcal{A}$ .
5.  $\mathcal{S}$  receives  $(\text{prove}, 1, N, (p_1, p_2))$  from  $\mathcal{A}$ , and verifies  $pk = N = p_1 \cdot p_2$ , checks the length of  $pk$  that  $|pk| = N$ . If it does not hold, the simulator  $\mathcal{S}$  simulates  $P_2$  to abort.
6.  $\mathcal{S}$  sends  $\text{continue}$  to  $\mathcal{F}_{GOST}$ , and stores  $(d_1, Q, pk, sk)$ .

The only difference between the simulation and a real execution is the generation of  $Q_2$ . In real execution,  $P_2$  chooses  $d_2$  randmoly and computes  $Q_2 = d_2 \cdot P$ , in simulation,  $\mathcal{S}$  computes  $Q_2 = d_1^{-1} \cdot Q$ . Since  $Q$  is chosen randomly, then the distribution between  $d_2 \cdot P$  and  $d_1^{-1} \cdot Q$  is identical. Therefore, the distribution between  $\mathcal{A}$ 's view and  $P_2$ 's output is identical.

In distributed signature generation phase,  $P_1$  can just receive a ciphertext  $C_1$  from  $P_2$ , the simulator can receive a result that equals to  $C$  from  $\mathcal{F}_{GOST}$ . Therefore, the main challenge is to prove that  $\mathcal{S}$  can generate  $P_1$ 's view of decryption of  $C_1$ , where  $\mathcal{S}$  only has the signature  $(r, s)$  from  $\mathcal{F}_{GOST}$ .

1. When receiving  $Sign(sid, m)$ ,  $\mathcal{S}$  sends  $Sign(sid, m)$  to  $\mathcal{F}_{GOST}$  and receives a signature  $(r, s)$ .
2.  $\mathcal{S}$  computes the point  $C$  by using GOST verification algorithm.
3.  $\mathcal{S}$  invokes  $\mathcal{A}$  when receiving  $Sign(sid, m)$  and simulates the following message to ensure the result is  $C$ .
  - (a)  $\mathcal{S}$  receives  $(\text{prove}, 1, (R_1, C_{key}), (k_1, sk))$  from  $\mathcal{A}$ .
  - (b)  $\mathcal{S}$  checks the equations  $R_1 = k_1 \cdot P$  and  $C_{key} = \text{Enc}_{pk}(k_1)$ , if the equations do not hold, then  $\mathcal{S}$  simulates  $P_2$  to abort and sends  $\text{abort}$  to  $\mathcal{F}_{GOST}$ . Otherwise, it continues.
4.  $\mathcal{S}$  randomly chooses  $\rho \in \mathbb{Z}_q$ , and computes  $C_1 = \text{Enc}_{pk}(s + \rho \cdot q)$ , where  $s$  is a signature value that received from  $\mathcal{F}_{GOST}$ , then sends  $C_1$  to  $\mathcal{A}$ .

The only difference of  $\mathcal{A}$ 's view between the real execution and the simulation is  $C_1$ . In the simulation,  $C_1 = \text{Enc}_{pk}(s + \rho \cdot q)$ , in the real execution,  $C_1 = \text{Enc}_{pk}(rd_1d_2 + ek_1k_2 + \rho \cdot q)$ . It has been already proved that the two distribution is statistically close. Therefore, the situation that  $P_1$  is corrupted has proved.

Then, we consider the situation that  $P_2$  is corrupted by an adversary  $\mathcal{A}$ . We construct a simulator  $\mathcal{S}$  works as follows:

1. When receiving  $KeyGen$ ,  $\mathcal{S}$  sends  $KeyGen$  to  $\mathcal{F}_{GOST}$ , then receives  $Q$  from  $\mathcal{F}_{GOST}$ .
2.  $\mathcal{S}$  generates a valid Paillier encryption key-pair  $(pk, sk)$ , and sends  $(\text{proof}, 1, N)$  to  $\mathcal{A}$ .
3.  $\mathcal{S}$  receives  $(\text{prove}, 2, Q_2, d_2)$  that  $\mathcal{A}$  intends to send to  $\mathcal{F}_{zk}^{RDL}$ .

4.  $\mathcal{S}$  verifies the equation that  $Q_2 = d_2 \cdot P$ , if the equation does not hold, it simulates  $P_1$  to abort.
5.  $\mathcal{S}$  computes  $Q_1 = d_2^{-1} \cdot Q$  and  $C_{key} = \text{Enc}_{pk}(C_{key})$ , then sends (proof, 1,  $Q_1, C_{key}$ ) to  $\mathcal{A}$ .
6.  $\mathcal{S}$  sends continue to  $\mathcal{F}_{GOST}$ , and stores  $(d_2, Q, C_{key}, pk)$ .

We can see that the view of adversary  $\mathcal{A}$  and the honest party  $P_1$  is indistinguishable, due to the honest party always outputs  $Q = d_1 \cdot Q_2 = d_2 \cdot Q_1$  in the real execution. In the simulation, due to  $Q_1 = d_2^{-1} \cdot Q$  and  $d_2 \cdot Q_1 = Q$ , therefore, the distribution between the real execution and the simulation is identical.

In distributed signature generation phase, the simulator  $\mathcal{S}$  works as follows:

1. When receiving  $\text{Sign}(sid, m)$ ,  $\mathcal{S}$  sends  $\text{Sign}(sid, m)$  to  $\mathcal{F}_{GOST}$ , then receives a signature  $(r, s)$  from  $\mathcal{F}_{GOST}$ .
2.  $\mathcal{S}$  computes the point  $C$  by using GOST verification algorithm.
3.  $\mathcal{S}$  invokes  $\mathcal{A}$  when receiving  $\text{Sign}(sid, m)$ , selects  $\tilde{k}_1$ , and computes  $C_{key} = \text{Enc}_{pk}(\tilde{k}_1)$ , then sends (proof, 1,  $R_1, C_{key}$ ) to  $\mathcal{A}$ , where  $R_1 = k_2 \cdot R$ .
4.  $\mathcal{S}$  receives (prove, 2,  $R_2, k_2$ ) from  $\mathcal{A}$ , then  $\mathcal{S}$  verifies the equation  $R_2 = k_2 \cdot P$ , if the equation does not hold, then  $\mathcal{S}$  simulates  $P_1$  to abort.
5.  $\mathcal{S}$  receives  $C_1$  from  $P_2$ , and decrypts  $C_1$  and reduces the result by modulo  $q$ .  $\mathcal{S}$  checks if it equals to  $(rd_1d_2 + ek_1k_2) \bmod q$ . If the equation holds, then  $\mathcal{S}$  sends continue to  $\mathcal{F}_{GOST}$ . Otherwise, it simulates  $P_1$  to abort.

We now modify  $\mathcal{S}$  to a simulator  $\mathcal{S}'$  who is hold an oracle  $\mathcal{O}_c(c', \alpha, \beta)$ . The oracle  $\mathcal{O}_c(c', \alpha, \beta)$  outputs 1 if and only if  $\text{Dec}_{sk}(c', \alpha, \beta) = \alpha + \beta \cdot k_1 \bmod q$ .  $\mathcal{S}'$  can simulate  $\mathcal{S}$  as follows:

1. Compute  $\alpha = rd_1d_2 \bmod q$ .
2. Compute  $\beta = ek_2 \bmod q$ .
3. Query  $\mathcal{O}_c(c', \alpha, \beta)$  and receive a response  $b$ .
4. If  $b = 1$  then  $\mathcal{S}'$  continues to simulate  $\mathcal{S}$ .

$\mathcal{S}$  accepts if  $\mathcal{S}'$  accepts, since these checks by  $\mathcal{S}$  and  $\mathcal{S}'$  are equivalent. Due to the Paillier-EC assumption is hard, we conclude that the output generated by  $\mathcal{S}'$  in the ideal model is computationally indistinguishable from the real execution. Since the output distribution of  $\mathcal{S}$  and  $\mathcal{S}'$  are identical in ideal model; therefore, the output generated by  $\mathcal{S}$  in the ideal model is computationally indistinguishable from the real execution.  $\square$

### 4.3 Zero-Knowledge Proof Analysis

In our protocol, the main zero-knowledge proof for the relations are  $\mathcal{F}_{zk}^{RP}$ ,  $\mathcal{F}_{zk}^{RDL}$ ,  $\mathcal{F}_{zk}^{RPDL}$ , which are defined in [10]. We use this zero-knowledge proof directly; thus, we omit the constructions here.

### 4.3.1 Proof that $r$ is a Discrete Log of $R$

In this section, we propose the constructions of zero-knowledge proof for the relation  $R_{DL}$  such that

$$R_{DL} = \{(G, Q, x) | Q = x \cdot G\}$$

We use Schnorr Zero Knowledge Proof [16] to achieve this requirement. In the distributed signing phase, if a malicious  $P_2$  sends (prove,  $Q_2, x_2$ ) to  $\mathcal{F}_{zk}^{R_{DL}}$ , it also receives a correct message (proof,  $Q_1$ ) from  $\mathcal{F}_{zk}^{R_{DL}}$ . However,  $P_1$ 's message in the protocol is assumed to be zero knowledge and hence does not reveal any information about the random integer  $x_1$ . The detailed zero-knowledge proof protocol is described as follows:

The joint statement is  $(G, Q)$ , the prover has a witness  $x$  and wishes to prove that  $Q = x \cdot G$ .

**Schnorr Zero Knowledge Proof Generation Algorithm:**

**Input:** Public parameter  $params = (q, a, b, n, G)$ , signer's identity  $ID$ , secret value  $x$ , and public value  $Q = x \cdot G$ .

**Output:**  $(z, e)$

1. Select  $k \xleftarrow{r} \mathbb{Z}_n$ , compute  $K = k \cdot G$ .
2. Compute  $e = H(params, ID, K, V)$
3. Compute  $z = K - xe$

**Schnorr Zero Knowledge Proof Verification Algorithm:**

**Input:** Public parameter  $params = (q, a, b, n, G)$ , public value  $Q$ , signer's identity  $ID$ , Schnorr zero knowledge proof values  $(z, e)$ .

**Output:** Valid or invalid

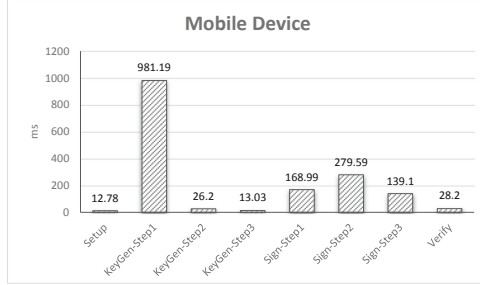
1. Perform public key validation for  $V$ .
2. Compute  $e = H(params, ID, K, V)$ .
3. If  $V = zP + eQ$ , then the signature is verified.

## 5 Performance Analysis

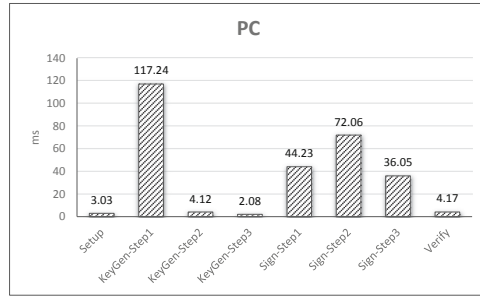
We implement our protocol using MIRACL Cryptographic SDK [12]. The proposed protocol is deployed on two Android devices (Samsung Nexus with a dual-core 1.2 GHz processor, 2G bytes memory and the Google Android 4.4.2 operating system, and two PCs (with an i7-6700 processor, 8G bytes memory and the Microsoft Windows 10 Professional operating system). The experimental curve we choose is the BN curve, as it achieves AES-128 security.

We analyze the computation cost of each progress in both distributed key generation and distributed signing phases. The experimental results is shown in Figs. 2 and 3. In the distributed key generation phase, KeyGen-Setp1 denotes the progress executed by P1 before P1 sends the first message to P2, KeyGen-Setp2 denotes the progress executed by P2, and KeyGen-Setp3 denotes the progress executed by P1 after receiving the message from P2. In the distributed signature

generation phase, Sign-Step1 denotes the progress executed by P1 before P1 sends the first message to P2, Sign-Step2 denotes the progress executed by P2, and Sign-Step3 denotes the progress executed by P1 after receiving the message from P2.



**Fig. 2.** Runtime on Android devices



**Fig. 3.** Runtime on PCs

## 6 Conclusion

As the number of mobile devices owned by any individual increases (e.g. one or more smart phones, one or more wearable devices that are capable of pairing with the smart phones, and Internet of Things devices in a smart home), splitting a single private key into multiple keys for storage on these different devices will be useful to ensure the protection of the user’s private key. This also removes the need for  $n$  number of private keys for  $n$  devices.

In this paper, we proposed a two-party distributed signing protocol for the GOST signature algorithm, which allows us to generate a valid signature without the need to reconstruct the private key in its entirety. The security of the

protocol is demonstrated, as well as its performance. Specifically, we demonstrated that the proposed protocol achieves better performance than other competing protocols.

Future research includes implementing and evaluating the protocol in collaboration with mobile device manufacturers.

**Acknowledgments.** The work was supported the National Natural Science Foundation of China under Grant 61972294, and Grant 61932016. The work was also supported the Natural Science Foundation of Hubei Province (No. 2020CFA052), the Wuhan Municipal Science and Technology Project (No. 2020010601012187), the Major Scientific and Technological Projects of Hubei Province (No. 2020AEA013). Kim-Kwang Raymond Choo was supported only by the Cloud Technology Endowed Professorship.


## References

1. Alam, M.K., et al.: An approach secret sharing algorithm in cloud computing security over single to multi clouds (2013)
2. Beimel, A.: Secret-sharing schemes: a survey. In: Chee, Y.M. (ed.) IWCC 2011. LNCS, vol. 6639, pp. 11–46. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-20901-7\\_2](https://doi.org/10.1007/978-3-642-20901-7_2)
3. De Prisco, R., De Santis, A.: Cheating immune  $(2, n)$ -threshold visual secret sharing. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 216–228. Springer, Heidelberg (2006). [https://doi.org/10.1007/11832072\\_15](https://doi.org/10.1007/11832072_15)
4. Gennaro, R., Goldfeder, S., Narayanan, A.: Threshold-optimal DSA/ECDSA signatures and an application to Bitcoin wallet security. In: Manulis, M., Sadeghi, A.-R., Schneider, S. (eds.) ACNS 2016. LNCS, vol. 9696, pp. 156–174. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-39555-5\\_9](https://doi.org/10.1007/978-3-319-39555-5_9)
5. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.* **17**(2), 281–308 (1988)
6. Harn, L.: Comments on fair  $(t, n)$  threshold secret sharing scheme. *IET Inf. Secur.* **8**(6), 303–304 (2014)
7. Harn, L., Lin, C.: Authenticated group key transfer protocol based on secret sharing. *IEEE Trans. Comput.* **59**(6), 842–846 (2010)
8. Hazay, C., Lindell, Y.: Efficient Secure Two-Party Protocols - Techniques and Constructions. Information Security and Cryptography. Springer, Heidelberg (2010)
9. ietf.org. GOST r 34.10-2012: Digital signature algorithm (2012). <https://tools.ietf.org/html/rfc7091>
10. Lindell, Y.: Fast secure two-party ECDSA signing. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10402, pp. 613–644. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-63715-0\\_21](https://doi.org/10.1007/978-3-319-63715-0_21)
11. Lipmaa, H., Wang, G., Bao, F.: Designated verifier signature schemes: attacks, new security notions and a new construction. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 459–471. Springer, Heidelberg (2005). [https://doi.org/10.1007/11523468\\_38](https://doi.org/10.1007/11523468_38)
12. Miracl: Miracl library (2017). <https://www.miracl.com/>
13. Nishioaka, T., Hanaoka, G., Imai, H.: A new digital signature scheme on id-based key-sharing infrastructures. ISW 1999. LNCS, vol. 1729, pp. 259–270. Springer, Heidelberg (1999). [https://doi.org/10.1007/3-540-47790-X\\_22](https://doi.org/10.1007/3-540-47790-X_22)

14. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). [https://doi.org/10.1007/3-540-48910-X\\_16](https://doi.org/10.1007/3-540-48910-X_16)
15. Pedersen, T.P.: A threshold cryptosystem without a trusted party. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 522–526. Springer, Heidelberg (1991). [https://doi.org/10.1007/3-540-46416-6\\_47](https://doi.org/10.1007/3-540-46416-6_47)
16. Schnorr, C.P.: Efficient signature generation by smart cards. *J. Cryptology* **4**(3), 161–174 (1991). <https://doi.org/10.1007/BF00196725>
17. Shamir, A.: How to share a secret. *Commun. Assoc. Comput. Mach.* **22**(11), 612–613 (1979)
18. Statista: Total revenue of global mobile payment market from 2015 to 2019 (in billion U.S. dollars). <https://www.statista.com/statistics/226530/mobile-payment-transaction-volume-forecast/>. Accessed 2018
19. Tassa, T.: Hierarchical threshold secret sharing. *J. Cryptology* **20**(2), 237–264 (2007)
20. Zhang, F., Safavi-Naini, R., Susilo, W.: An efficient signature scheme from bilinear pairings and its applications. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 277–290. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-24632-9\\_20](https://doi.org/10.1007/978-3-540-24632-9_20)



# A Novel Web Anomaly Detection Approach Based on Semantic Structure

Zishuai Cheng<sup>(✉)</sup> , Baojiang Cui, and Junsong Fu

School of Cyberspace Security, Beijing University of Posts and Telecommunications,  
Beijing 100087, China  
{chengzishuai, cuibj, fujs}@bupt.edu.cn

**Abstract.** In recent years, various machine learning, deep learning based models have been developed to detect novel web attacks. These models are mostly use NLP methods, like N-gram, word-embedding, to process URLs as the general strings composed of characters. In contrast to natural language which consist of words, the URL is composed of characters and hardly decomposes into several meaning segments. In fact, HTTP requests have its inherent patterns, which so-called semantic structure, such as the request bodies have fixed type, request parameters have fixed structure in names and orders, values of these parameters also have special semantics such as username, password, page id, commodity id. These methods have no mechanism to learn semantic structure. They roughly use NLP techniques like DFA, attention techniques to learn normal patterns from dataset. And, they also need a mount of dataset to train. In this paper, we propose a novel web anomaly detection approach based on semantic structure. Firstly, a hierarchical method is proposed to automatically learn semantic structure from training dataset. Then, we learn normal profile for each parameter. The experimental results showed that our approach achieved a high precision rate of 99.29% while maintaining a low false alarm rate of 0.88%. Moreover, even on a small training dataset composed of hundreds of samples, we also achieved 96.3% accuracy rate.

**Keywords:** Anomaly-based detection · Semantic structure · Machine learning

## 1 Introduction

Web-based applications are more and more popular and provide various services for individuals and organizations [2]. Daily tasks, such as E-commerce, E-government, E-mail, and social networking, are mostly processed via Web-based applications. Meanwhile, users usually store sensitive data in those applications. The importance and sensitiveness of Web-based applications attract a lot of interest from attacks. Web-based applications are suffering from many types of web attacks such as SQL injection, Cross-site scripting (XSS) attack, Web-Shell attack, etc. [1]

Defending Web-based applications from attacks is a challenging task. Cyber-defense is an asymmetric warfare as attackers have great advantages [27]. Intrusion detection systems are continuously identifying attacks relying on the up-to-date signatures or models, while attackers only need a single vulnerability for victory. Anomaly-based intrusion detection approaches provide an ability to detect attacks by identifying abnormal behaviors with deviating from the normal behaviors which have been profiled in training phase [12].

A great number of anomaly-based detection methods have been developed by researchers in recent years. Igino et al. [5] model the sequence and values for each attribute of queries based on Hidden Markov Model (HMM). Davide et al. [3] propose an intrusion detection approach based on HMM that models the character sequences of the HTTP payload. Wen et al. [9] propose an adaptive anomaly detection model based on HMM. Those researchers introducing the semantic structure in the anomaly detection approach, but these works use HMM mainly to learn the character sequence of URLs which is only a part of semantic structure.

Moreover, Deep learning technique has been used in anomaly-based detection model to learn higher-level features. Qin et al. [19] propose a model which learns semantic of malicious segments in payload using Recurrent Neural Network (RNN) with attentional mechanism. Yu et al. [26] propose a method that uses Bidirectional Long Short-Term Memory (Bi-LSTM) with attention mechanism to model HTTP traffic. Although the attention mechanism can learn the semantic of attack patterns, these models still have drawbacks. URLs are treated as the meaningless general string composed by characters and ignore the semantic structure in HTTP-request scenario. And also, training the deep learning-based models need lots of samples, but high-quality training data is difficult to obtain in the real world [28].

In this paper, we propose a novel anomaly detection approach based on semantic structure of URLs. Firstly, we propose an algorithm that automatically learns semantic structure information from training dataset. We use pattern-tree, logical parts and trivial parts to represent the semantic structure [17]. Each path of this tree is a piece of semantic structure which illustrates a specific structure. Next, we build anomaly detection model for each node of pattern-tree using machine learning technique based on length, characteristic distribution, structure inference. Finally, we classify URL as normal or abnormal using semantic structure and anomaly detection model. This approach considers entire semantic structure of URLs and produces a very precise normal behavior model. Our approach is very sensitive and is able to detect malicious such as web-shell, SQL intrusion, XSS. This approach has a very low false positive rate in despite the fact that it has high sensitiveness. Even on the small training dataset, it still has a good performance.

The contributions of this paper are summarized as follows.

- An efficient Web intrusion detection approach is proposed, based on semantic structure. Compared with previous research which treats the URL as



meaningless string composed by letters, we treat the URL as the meaningful combination of parts.

- We improved the Markov detection model to decrease the size and improve learning ability.
- We evaluated our approach on CSIC-2010 [11] dataset and achieve better performance than previously published results.

The rest of this paper is organized as follows. In Sect. 2 we introduce the related work, focusing on anomaly-based detection research and semantic structure research. The framework of our novel anomaly detection approach is introduced in Sect. 3. In Sect. 4, we report the simulation environment and results. Finally, we draw conclusions and future points in Sect. 5.

## 2 Related Work

Since anomaly-based intrusion detection was first introduced in 1987 by D. Denning et al. [8], the research associated with this field has been rapidly developed. Kruegel et al. [13], [14] proposed an anomaly detection system for Web-attacks, which takes advantage of the particular structure of HTTP queries that contains parameter-value pairs. kruegel et al. assemble separated models to detect attacks. Each model is built on different features, such as attribute’s length, character distribution, structural inference, token finder attribute presence or absence and attribute order and separately outputs the anomaly probability value. The request is marked as malicious if one or more features’ probability exceed the defined threshold. Cho et al. [4] proposed a model which uses Bayesian parameter estimation to detect anomalous behaviors. PAYL [24] used the frequency of n-grams in the payload as features. A recent version of PAYL is proposed [23], which add some functionalities such as multiple centroids, and ingress/egress correlation, to the original version. These authors focus their efforts on solving the problem of how to build the behavior models that significantly distinguish abnormal behavior from normal behavior.

More recently, some anomaly detection methods based on feature selection are proposed [6, 18, 21, 22, 29]. In [18, 22], authors combined expert knowledge with n-gram feature for reliable and efficient web attack detection and use the Generic-FeatureSelection (GeFS) measure to eliminate redundant and irrelevant features. Zhou et al. [29] proposed an ensemble learning approach to detect XSS attack. They use a set of Bayesian networks, which each Bayesian network is built with both domain knowledge and threat intelligence. All these authors defined features based on their expert knowledge. Nevertheless, the selected features are well fitting with the specific environment such as training dataset and not adaptive to various network environments.

To the best of our knowledge, there is few web instruction detection method using the semantic structure. In other research areas, researchers have taken advantage of this information. Lei et al. [17] propose a concept of pattern-tree that leverages the statistic information of the training set to learn URL patterns. This paper uses a top-down strategy to build a tree and uses statistic information

to make the learning process more robust and reliable. Yang et al. [25] propose an unsupervised incremental pattern-tree algorithm to construct a pattern-tree and extract main patterns from it to classify Web page.

### 3 Framework of Our Approach

Without loss of generality, in this paper, we mainly focus on the HTTP request-URLs which using GET method. Although we focus on the GET requests here, our method also can be extended to all request methods, such as POST, HEAD, PUT, by converting the request data or parameters as parameter-value format.

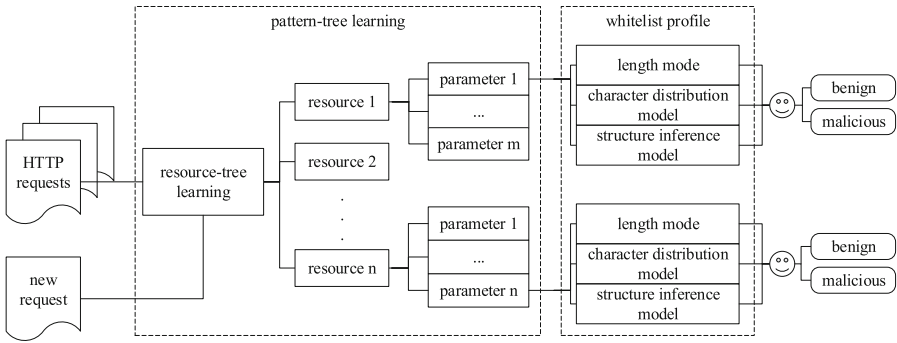


Fig. 1. The framework of our approach.

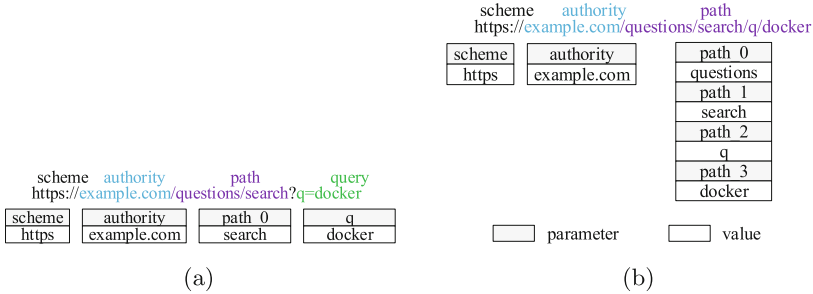
As shown in Fig. 1, our approach consists of learning phase and detection phase. In learning phase, we mainly learn semantic structure of request URL for website and build anomaly-based detection model for each trivial logical part. In detection phase, we propose an approach which based on semantic structure and anomaly detection model to classify new HTTP request as normal or abnormal.

#### 3.1 Learn Semantic Structure of a URL

We denote the collection of URLs dataset as  $U = \{u_1, u_2, \dots, u_m\}$ , in which  $u_i$  is the  $i$ -th request URL. According to HTTP protocol [10], each request URL  $u_i$  can be decomposed into several components (e.g. scheme  $sch$ , authority  $auth$ , path  $path$ , optional path information component  $pinfo$ , optional query string  $query$ ) by delimiters like ‘:’, ‘/’ and ‘?’. As shown in Fig. 2, URLs can be decomposed into  $sch, auth, path, query$ .

Components before ‘?’ are called static parts (i.e.,  $scheme, authority, path, pinfo$ ) and the rest components (i.e.,  $query$ ) are dynamic parts.  $path$  usually identifies the requesting resource and has a hierarchical structure. It can be further decomposed into a collection which composed of logical parts  $\{(p_1, v_1), \dots, (p_n, v_n)\}$ ,  $v_i$  is the  $i$ -th value in  $path$  split by ‘/’ and  $p_i$  is

corresponding index of  $v_i$ . The query string *query* always contains parameters and corresponding values submitted to server-side programs by users. As same to *path*, *query* also can decompose into a collection  $\{(p_1, v_1), \dots, (p_n, v_n)\}$ , in which  $p_i$  is the name of  $i$ -th parameter in *query*,  $v_i$  is the corresponding values of  $i$ -th parameter,  $n$  is the numbers of parameter-value pairs in *query*.

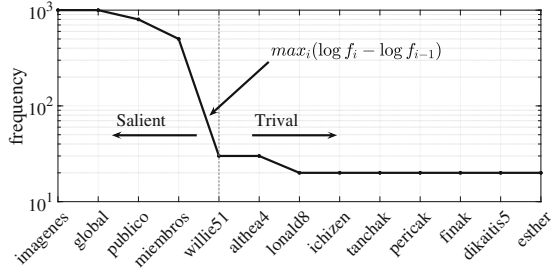


**Fig. 2.** The syntax structure of URL. a is dynamic URL which *path* can present as  $\{(path_0, question), (path_1, search)\}$  and *query* can present as  $\{(q, docker)\}$ . b is a pseudo-static URL.

However, in the real world, *path* not only identifies the requesting resource but also contains parameter-value pairs. Most Web-based applications use pseudo-static technique [7, 20] to make the Web application more friendly to search engine. As Fig. 2b, the pseudo-static is a technique that translates the dynamic parts, like *query*, as a static format and append them as a part of the static part. The pseudo-static technology poses a challenge to learn semantic structure information. We need to learn the function or meaning of each logical part, that is, is this part whether identifying the requesting resource or just a value of parameter submitted by a user.

We use a specialized tree, named pattern-tree [17], to learn the semantic structure of request URLs. We keep salient values in pattern-tree’s node and generalize trivial values with regular expressions ‘\*’. We determine whether a value is salient or trivial based on its frequency cure and entropy. As shown in Fig. 3, when values’ appearance frequencies are stored in descending order, there exists a position in the frequency-curve that has the max frequency of descent. Values on the left-hand side of this position are considered to be salient, on the contrary, values on the right-hand side of this position are considered to be trivial. The position is calculated as:  $pos_{dec} = max_i(\log f_i - \log f_{i-1})$ , where  $f_i$  is the appearance frequency of the  $i$ -th logical part.

As shown in Algorithm 1, we use a top-down splitting strategy to divide the URLs into subgroups and build a pattern-tree. First, we determine the first logical part of all URLs as salient or trivial. Each salient value is reserved and all trivial values are generalized as ‘\*’. According to these salient values and ‘\*’, we can split URLs into subgroups. Then, we further classify the next logical



**Fig. 3.** A example of the values. From the frequency curve, it is clear that the maximum decline point can help distinguish salient values from trivial ones.

part as salient/trivial on each subgroup. We repeat to determine logical part as salient or trivial and divide URLs into subgroups recursively, until the subgroup is empty. Finally, we build a pattern-tree, each path of the tree is a piece of semantic structure information. Each node in the pattern-tree is a logical part in the URL and can illustrates the type of this logical part as salient or trivial.

We retrieve a path of the pattern-tree using the key-value collection  $kv_i$ . For example, for a request-URL ‘/question/search?q=docker’, we retrieve the path according to its key-value collection,  $kv = \{(p_0, question), (p_1, search), (q, docker)\}$ . We examine the first key-value pair  $\{p_0, questions\}$  on pattern-tree. If the key-value pair exists, the search is valid and we further examine the next key-value pair in  $kv$  on the corresponding child-tree. If the key-value pair does not exist, we replace the value of this key-value pair with ‘\*’ and re-examine it. This process is repeated until all key-value pairs in  $kv$  are examined or sub-tree is null. For this request-URL shown in Fig. 4, the retrieval path is marked with an arrow. This path shows that the semantic structure is ‘/question/search?q=\*’, where the parameter q is trivial and the value of q can consist malicious payload to launch attacks.

### 3.2 Build Anomaly-Based Detection Model of a Logical Part

In building anomaly detection model phase, we first divide URLs  $U$  into several subsets  $\{U_1, U_2, \dots, U_n\}$  based on semantic structure (also is pattern-tree), where  $n$  is the number of subsets that equal to the number of semantic structure of the Web application. The subset  $U_i$  has the following characters:

1.  $\forall u \in U_i$ , URL  $u$  has the same semantic structure knowledge.
2.  $\forall i \neq j, U_i \cap U_j = \emptyset$ .
3.  $\sum_{i=1}^n U_i = U$ ,  $n$  is the number of subsets.

According to semantic structure, we can extract the values of each trivial logical part for URL  $u$  and combine these values as a vector  $pv = \{(p_1, v_1), (p_2, v_2), \dots, (p_q, v_q)\}$ , where  $p_i$  is the index for the  $i$ -th logical part and  $v_i$  the value of this logical part,  $q$  is the number of trivial logical parts in

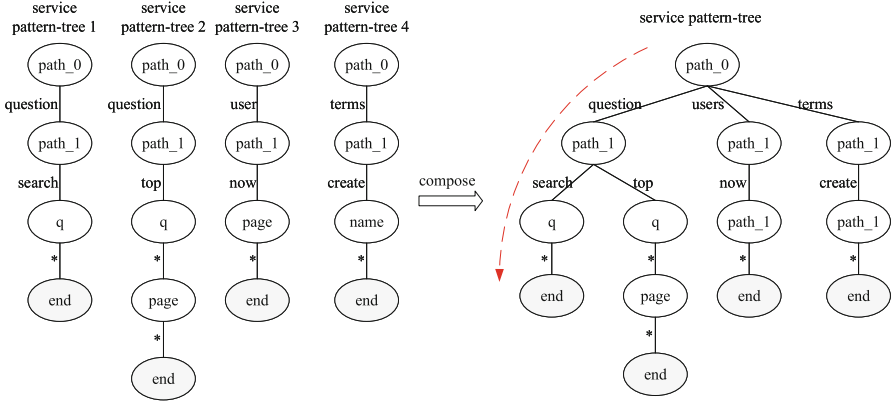


Fig. 4. A example of pattern-tree

---

**Algorithm 1.** *ConstructPatternTree* ( $U, j$ )

---

**Input:** Given a URL group  $U$  and initialize  $j$  as 0

**Output:** A tree node  $t$  for URLs in  $U$

---

- 1: create a new node  $n$
  - 2: **if**  $j >$  the number of parameter-value pairs for URLs in  $U$  **then**
  - 3:     **return** the node  $n$
  - 4: **end if**
  - 5: extract  $j$ -th parameter-value pair for each URL in  $U$
  - 6: calculate frequency-curve for parameter  $k$
  - 7: **for** URL  $u \in U$  **do**
  - 8:     **if** value  $v$  of  $k$  for  $u$  is salient **then**
  - 9:          $V_{k1} = V_{k1} \cup v$
  - 10:     **else**
  - 11:          $V_{k1} = V_{k1} \cup \text{'*'}$
  - 12:     **end if**
  - 13: **end for**
  - 14: calculate entropy  $H(k)$  for this  $j$ -th parameter  $k$
  - 15: **if**  $H(k) >$  threshold  $t$  **then**
  - 16:      $V_{k2} =$  the first *max\_number* values of  $U$
  - 17:      $V_k = (V_{k1} \cup \text{'*'}) \cap V_{k2}$
  - 18: **else**
  - 19:      $V_k = V_{k1}$
  - 20: **end if**
  - 21: split  $U$  into sub-groups  $\{U_1, U_2, \dots, U_t\}$  according to  $V_k$
  - 22: **for** all subgroup  $U_i$  **do**
  - 23:      $ch = \text{ConstructPatternTree}(U_i, j + 1)$
  - 24:     add  $ch$  to  $n$  as child node
  - 25: **end for**
  - 26: **return** the node  $n$
-

$u$ . Furthermore, we extract  $pv$  for each URL  $u$  in  $U_i$ , and combine these  $pv$  as a  $m \times q$  matrix  $PV_i$ :

$$PV_i = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1q} \\ v_{21} & v_{22} & \cdots & v_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mq} \end{bmatrix}$$

where  $m$  is the number of URLs in  $U_i$ . The  $j$ -th column  $[v_{1j}, v_{2j}, \dots, v_{mj}]$  is the values for the  $j$ -th trivial logical part for all URLs in  $U_i$ .

After extract matrix  $PV_i$  for each subset  $U_i$ . We observe the values column by column of each  $PV_i$  and automatically build profile  $pro_j$  for  $j$ -th column using statistical learning techniques highlighted in [13]. Then we get  $q$  profiles for  $PV_i$ . These  $q$  profiles consist the integral anomaly detection model  $model_i$  for a special piece of semantic structure, also the  $i$ -th path of pattern-tree. Each profile describes the normal behavior of request values for logical part in three aspects: length, character distributes and sequence structure of values. All these model  $\{model_1, model_2, \dots, model_n\}$  consist the entire anomaly detection model of this Web application.

### 3.3 Anomaly Detection

In detection phase, we determine incoming HTTP request as benign or malicious based on semantic structure and anomaly based detection model. When a new HTTP request coming, we first use pattern-tree to illustrate the request URL. If this URL is not successfully retrieved from pattern-tree, we classify this URL as malicious directly. Contrarily, this URL matches  $j$ -th path of pattern-tree. According to this path, we determine which logical part is trivial and extract the values of trivial logical parts as a collection  $pv$ . For each value  $v_i$  in  $pv$ , we use the corresponding  $pro_i$  in  $model_i$  to detect it whether is benign or malicious. If any value of  $pv$  is determined as malicious, the URL is classify as malicious. Otherwise, the URL is classify as benign.

## 4 Experiment

In order to evaluate the ability of our novel anomaly-detection approach, we conducted several experiments. To analyze the effect of semantic structure information on the performance of our model, we observe the change in length distribution and final classification performance when we control whether structure information is considered. To investigate the sensitivity of the model to the training data size, we tested the performance on different size training sets. And also we show the advantage of using character substitution approach. Finally, we compare our model with five existing models.

## 4.1 Experimental Settings

**Datasets.** The experiment was conducted on CSIC-2010 [11], which contains thousands of Web requests automatically generated by creating traffic to an e-commerce web application. The dataset consists of three subsets: 36,000 normal requests for training, 36,000 normal requests and 25,000 anomalous requests for the test. There are three types of anomalous request: static attacks that request for hidden(non-existent) resources, dynamic attacks that modify the valid request arguments, and unintentional illegal requests that have no malicious intention, however they do not follow the normal behavior of the web application and do not have the same structure as normal parameter values [19].

The dataset consists of HTTP requests for several resource and contains two request methods: *GET* and *POST*. According to the desired resource, dataset can divide into two types. One is requesting static resources, such as .jpg, .git, .css, .js format file stored on server. The other is requesting dynamic resources, which need to be processed by the server-side program and the response results are the execution results.

**Metrics.** There are numbers of performance metrics that can be used to evaluate the performance of anomaly-detection system. The most commonly used metrics in this field are precision, recall, F1-score and accuracy (ACC). In this paper, we use these metrics to evaluate our novel anomaly-detection approach:

- **Precision** is defined as the number of true positives divided by the number of true positives plus the number of false positives.

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

- **Recall** is defined as the percentage of positive cases you caught.

$$recall = \frac{true\ positives}{true\ positives + false\ negative}$$

- **F1-score** is the harmonic mean of precision and recall taking both metrics into account.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

- **Accuracy (ACC)** measures in percentage form, where instances are correctly predicted.

$$Accuracy(ACC) = \frac{TP + TN}{TP + TN + FP + FN}$$

## 4.2 The Comparative Performance of Using Semantic Structure or Not

To illustrate the effect of semantic structure on classification performance. We implement two systems, one using semantic structure and build profile for each

type of trivial logical part, and the another is build profile by observing all values in all request-URLs. Then we compare their classification performance.

As shown in Table 1, it is the confusion matrix of these two models. When not using semantic structure information, the detection model has poor performance. For 2500 benign request-URLs, the model predicts 249 request-URLs are malicious with 9.96% false-positive rate. For 2500 malicious request-URLs, the model predicts 906 request-URLs are malicious with 36.24% recall rate and 63.76% false-negative rate. When using semantic structure information, the performance of the anomaly-based detection model has a great improvement. False-positive has reduced to 0.76% and recall has improved to 96.12%. On f1-score, we have improved it from 0.4957 to 0.9722 with 96.12% improvement rate.

**Table 1.** Confusion Matrix of anomaly-based detection model whether using semantic structure information

(a) The performance of anomaly-based detection model without semantic structure.				(b) The performance of anomaly-based detection model with semantic structure.			
		Actual				Actual	
		Benign	Malicious			Benign	Malicious
Predicted	Benign	2251	1594	Predicted	Benign	2481	106
	Malicious	249	906		Malicious	19	2394

This result shows that using the same methods to build normal-based detection profile, semantic structure can tremendously help us building a precise model and improve the performance. Thus, it is necessary to use semantic structure to improve the performance of the detection model in Web attack detection field.

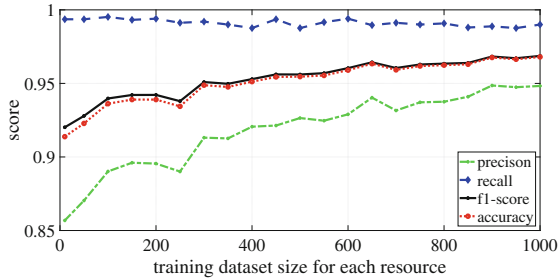
### 4.3 The Performance on Different Dataset Size

This experiment intends to measure the impact of the scale of training dataset. We construct several training datasets of different sizes by randomly choosing the request-URLs for each resource. The training datasets consist of 10, 50 to 1000 with 50 steps HTTP request examples for each resource. Then, we train and evaluate our model on each training dataset.

The result is shown in Fig. 5, as the size of training dataset increases, *precision*, *f1-score*, *acc* of this model also increases, although *recall* decreased. The precision score increase from 0.8568 to 0.9483 as the data size increase from 10 to 100. F1-score and accuracy are also increased from 0.9202 to 0.9687 and from 0.9138 to 0.963 separately. Only recall decreased from 0.9936 to 0.99. When feeding more training examples to model, the anomaly-based detection model can learn more precise normal behavior(length, character distributions, and structure). Thus, the classification performance is being better. When the data size



is greater than 900, the impact of increasing size of training dataset on classification performance is not obvious. Especially, on the very small training dataset that each resource only has 10 request-URLs examples, our model also achieves 96.2% accuracy.



**Fig. 5.** The performances of our approach to different scales of training datasets.

This result shows that our approach based on semantic structure not require large scale dataset in training phase. On the tiny training dataset, our method also achieves good performance. Compare with other anomaly-based detection methods, especially of deep learning based method need 5000 sample for each class to get an ideal classification performance [16], we can learn enough knowledge to achieve a acceptable performance from limited dataset. Our model solves the knotty problem that there not exists enough scale training dataset to train an anomaly-based detection model in real life.

#### 4.4 Compare with Other Approach

We compared our model with other anomaly-based detection approaches on CSIC-2010 Dataset. The results are described in Table 2 which includes classification performances of SOM, C4.5, Naive Bayes, X-means and EM approaches evaluated in [15]. Compared with all other models on the CSIC-2010 dataset, our model achieves the best performance in Precision, F1-score, ACC, False-Positive rate. Even through, X-means reported the highest recall, it does not perform well in precision, F1-score and accuracy. Our model is very sensitive to detect malicious request and also maintaining a low false-positive rate of 0.88%.

This comparison result shows that our novel detection method based on semantic structure achieves better performance than those methods not using semantic structure.

**Table 2.** Classification performance of our approach and other approaches

	Precision	Recall	F1-Score	Acc	FP
SOM	0.6980	0.9497	0.8046	0.9282	0.0503
C4.5	0.9654	0.8697	0.9150	0.9650	0.1303
Native Bayes	0.6696	0.5235	0.5876	0.8408	0.4765
X-means	0.4631	<b>0.9865</b>	0.6303	0.7493	0.0135
EM	0.4851	0.7516	0.6167	0.786	0.2484
Our approach	<b>0.9929</b>	0.9552	<b>0.9737</b>	<b>0.9742</b>	<b>0.0088</b>

## 5 Conclusion and Future Work

In this work, we proposed a novel anomaly detection approach for web applications that leveraging semantic structure knowledge. We proposed approach to learn semantic structure information and built an anomaly detection profile for each type of trivial parameter in three aspects: length model, character distribution model, and structure model. Then we used the detection results from each trivial parameter in URL to classify whether the incoming URL is malicious.

The proposed approach was tested on the CSIC-2010 dataset. Using semantic structure, we achieved 97.42% accuracy and 99.29% precision. And F1-score and recall increased 196.12% and 162.8% than without using semantic structure. Even on the small dataset that only contains 10 records for each type of URL, our approach also archives 88.94% accuracy.

In the future, we intend to research how to learn the changing of semantic structure information with an increment learning mechanism. To provide better services for users, Web-application is constantly evolved, such as adding new or removing old resources and changing the parameters of resources. Thus, the semantic structure information of the Web-application is changing frequently.

## References

1. Adhyaru, R.P.: Techniques for attacking web application security. *Int. J. Inf.* **6**(1/2), (2016)
2. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web services. In: *Web Services*, pp. 123–149. Springer (2004). [https://doi.org/10.1007/978-3-662-10876-5\\_5](https://doi.org/10.1007/978-3-662-10876-5_5)
3. Ariu, D., Tronci, R., Giacinto, G.: HMMPayl an intrusion detection system based on hidden Markov models. *Comput. Secur.* **30**(4), 221–241 (2011)
4. Cho, S., Cha, S.: Sad: web session anomaly detection based on parameter estimation. *Comput. Secur.* **23**(4), 312–319 (2004)
5. Corona, I., Ariu, D., Giacinto, G.: Hmm-web: a framework for the detection of attacks against web applications. In: *2009 IEEE International Conference on Communications*, pp. 1–6. IEEE (2009)
6. Cui, B., He, S., Yao, X., Shi, P.: Malicious URL detection with feature extraction based on machine learning. *Int. J. High Perform. Comput. Netw.* **12**(2), 166–178 (2018)

7. Cui, M., Hu, S.: Search engine optimization research for website promotion. In: 2011 International Conference of Information Technology, Computer Engineering and Management Sciences, vol. 4, pp. 100–103. IEEE (2011)
8. Denning, D.E.: An intrusion-detection model. *IEEE Trans. Software Eng.* **2**, 222–232 (1987)
9. Fan, W.K.G.: An adaptive anomaly detection of web-based attacks. In: 2012 7th International Conference on Computer Science & Education (ICCSE), pp. 690–694. IEEE (2012)
10. Fielding, R., et al.: Hypertext transfer protocol-HTTP/1.1. Technical report (1999)
11. Giménez, C.T., Villegas, A.P., Marañón, G.Á.: HTTP data set CSIC 2010. *Inf. Secur. Inst. CSIC (Span. Res. Nat. Coun.)* (2010)
12. Hawkins, D.M.: Identification of Outliers. Springer, Dordrecht (1980). <https://doi.org/10.1007/978-94-015-3994-4>
13. Kruegel, C., Vigna, G.: Anomaly detection of web-based attacks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 251–261. ACM (2003)
14. Kruegel, C., Vigna, G., Robertson, W.: A multi-model approach to the detection of web-based attacks. *Comput. Netw.* **48**(5), 717–738 (2005)
15. Le Jr, D.: An unsupervised learning approach for network and system analysis (2017)
16. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436 (2015)
17. Lei, T., Cai, R., Yang, J.M., Ke, Y., Fan, X., Zhang, L.: A pattern tree-based approach to learning URL normalization rules. In: Proceedings of the 19th International Conference on World Wide Web, pp. 611–620. ACM (2010)
18. Nguyen, H.T., Torrano-Gimenez, C., Alvarez, G., Petrović, S., Franke, K.: Application of the generic feature selection measure in detection of web attacks. In: Herrera, Á., Corchado, E. (eds.) *CISIS 2011*. LNCS, vol. 6694, pp. 25–32. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-21323-6\\_4](https://doi.org/10.1007/978-3-642-21323-6_4)
19. Qin, Z.Q., Ma, X.K., Wang, Y.J.: Attentional payload anomaly detector for web applications. In: Cheng, L., Leung, A.C.S., Ozawa, S. (eds.) *ICONIP 2018*. LNCS, vol. 11304, pp. 588–599. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-04212-7\\_52](https://doi.org/10.1007/978-3-030-04212-7_52)
20. Shi, J., Cao, Y., Zhao, X.J.: Research on SEO strategies of university journal websites. In: The 2nd International Conference on Information Science and Engineering, pp. 3060–3063. IEEE (2010)
21. Tang, P., Qiu, W., Huang, Z., Lian, H., Liu, G.: SQL injection behavior mining based deep learning. In: Gan, G., Li, B., Li, X., Wang, S. (eds.) *ADMA 2018*. LNCS (LNAI), vol. 11323, pp. 445–454. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-05090-0\\_38](https://doi.org/10.1007/978-3-030-05090-0_38)
22. Torrano-Gimenez, C., Nguyen, H.T., Alvarez, G., Franke, K.: Combining expert knowledge with automatic feature extraction for reliable web attack detection. *Secur. Commun. Netw.* **8**(16), 2750–2767 (2015)
23. Wang, K., Cretu, G., Stolfo, S.J.: Anomalous payload-based worm detection and signature generation. In: Valdes, A., Zamboni, D. (eds.) *RAID 2005*. LNCS, vol. 3858, pp. 227–246. Springer, Heidelberg (2006). [https://doi.org/10.1007/11663812\\_12](https://doi.org/10.1007/11663812_12)
24. Wang, K., Stolfo, S.J.: Anomalous payload-based network intrusion detection. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) *RAID 2004*. LNCS, vol. 3224, pp. 203–222. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-30143-1\\_11](https://doi.org/10.1007/978-3-540-30143-1_11)

25. Yang, Y., Zhang, L., Liu, G., Chen, E.: UPCA: an efficient URL-pattern based algorithm for accurate web page classification. In: 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 1475–1480. IEEE (2015)
26. Yu, Y., Yan, H., Guan, H., Zhou, H.: DeepHTTP: semantics-structure model with attention for anomalous HTTP traffic detection and pattern mining. arXiv preprint [arXiv:1810.12751](https://arxiv.org/abs/1810.12751) (2018)
27. Yurcik, W., Barlow, J., Rosendale, J.: Maintaining perspective on who is the enemy in the security systems administration of computer networks. In: In ACM CHI Workshop on System Administrators Are Users, p. 345. ACM Press, November 2003
28. Zhang, J., Zulkernine, M.: Anomaly based network intrusion detection with unsupervised outlier detection. In: 2006 IEEE International Conference on Communications, vol. 5, pp. 2388–2393. IEEE (2006)
29. Zhou, Y., Wang, P.: An ensemble learning approach for XSS attack detection with domain knowledge and threat intelligence. *Comput. Secur.* **82**, 261–269 (2019)



# A Malicious URL Detection Model Based on Convolutional Neural Network

Zhiqiang Wang<sup>1,2,3(✉)</sup>, Shuhao Li<sup>2</sup>, Bingyan Wang<sup>2</sup>, Xiaorui Ren<sup>2</sup>, and Tao Yang<sup>3</sup>

<sup>1</sup> State Information Center, Beijing, China  
wangzq@besti.edu.cn

<sup>2</sup> Beijing Electronic Science and Technology Institute, Beijing, China

<sup>3</sup> Key Lab of Information Network Security, Ministry of Public Security, Shanghai, China

**Abstract.** With the development of Internet technology, network security is facing great challenges. Malicious URL detection can defend against attacks such as phishing, spams, and malware implantation. However there are some problems on current malicious URL detection, for example the methods used to extract features are inefficient and hard to adapt to the current complex network environment. To solve these problems, this paper uses the word embedding method based on character embedding as the way of vector embedding to improve the deep convolutional neural network, and designs a malicious URL detection system. Finally, we carry out experiments with the system, the results prove the effectiveness of our system.

**Keywords:** Network security · Convolutional neural network · Malicious URL detection

## 1 Introduction

Malicious URLs are usually used by attackers to enticed users to clicking them through spam, phishing, etc. The malicious URLs detection system can help users identify malicious URLs and protect their money and against from attacks. Traditionally, researchers adopt blacklist-based methods to detect malicious URLs, however, hackers can use Domain Generation Algorithm to generate malicious domain names which can escape this kind of detection. This problem can be solved by the machine learning technique. Extracting features manually is needed when using the conventional way of machine learning. Hackers can also escape detection by designing these features. Faced with today's various attacks and threats, designing a more effective malicious URL detection system has become a research focus.

To solve above problems, this paper proposes a malicious URL detection model. The innovations and contributions of this paper are as follows:

- (1) Based on a dynamic convolutional neural network, it adopts a character embedding method based on word embedding to carry out feature extraction and representation.

- (2) This paper optimizes and improves the overall training model. We add multiple embedded layers to handle different URL fields and realize the expansion of the network model to fully obtain the information carried in the URL string, expand the input of the full connection layer to improve the detection effect.
- (3) We verify the validity and feasibility of the model through experiments. These experiments adopts different Embedding method and different network structure.

The rest of this paper is organized as follows. In Sect. 2, we present the malicious URL detection model and main modules designed in this paper. In Sect. 3, we conduct experiments on the malicious URL detection model and offer the experimental results. Finally, we offer a brief discussion in Sect. 4.

## 2 Malicious URL Detection Model

Our paper proposes a malicious URL detection model based on convolutional neural networks. The construction of the model is shown in Fig. 1. The model mainly includes three modules: vector embedding module, dynamic convolution module, and block extraction module.

The vector embedding module is used to represent the input URL sequence as a suitable vector form to facilitate the processing of subsequent modules.

The dynamic convolution module adopts a dynamic convolution network to extract features from the input data automatically.

The block extraction module extract different fields such as the subdomain name, domain name, and domain name suffix from URLs.

### 2.1 Detection Process

The detection process is as follows. First, the domain name, subdomain name, and domain name suffix are sequentially extracted from URLs. In the first branch of the detection model, each URL is padded to a fixed length, of which every word is marked with a specific number. The entire URL is represented as a sequence of numbers. Then, the sequences are input to the embedding layer, **and** trained together with other layers. These sequences will learn the appropriate vector expression during the training process. The data streams output by the embedding layer are subsequently input into a dynamic convolutional network. That is the outputs pass through two convolution layers, two folding layers and two pooling layers which lay out in two successive rounds. In the flatten layer, the data streams are flattened, and then wait for connections with data from the other branch. In another branch of the detection model, the domain name, subdomain name, and domain name suffix are marked firstly, and the different main domain name, subdomain name, or domain name suffix in each field is encoded as an independent expression. Then the marked data are directly input into the three newly added embedding layers, and obtain the appropriate vector expression. Then the information is transformed into a suitable shape in the Reshape layer and connected with the data of the first branch. The connected data are jointly input into the fully connected layer for training, after the dropout layer, the results are output into the output layer.

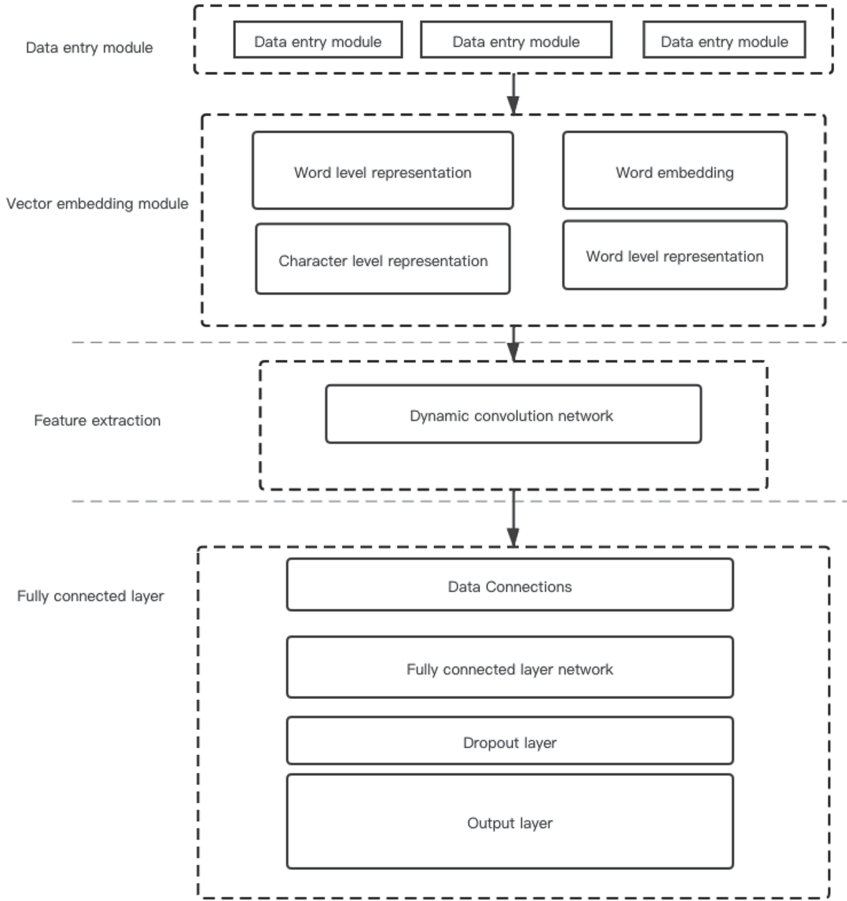


Fig. 1. The detection model

### 3 Experiment and Evaluation

#### 3.1 Experiment Data Set

A large amount of URL data from the network are collected in our paper, from [github.com](https://github.com), [kaggle.com](https://www.kaggle.com), and [uci.edu](https://uci.edu) websites to verify the validity and feasibility of the model. The data set division is shown in Table 1.

#### 3.2 Experiment Setting

Each URL's length was set to 200 words, and the vector embedding dimension was set to 32. The dynamic convolutional network included two convolutional layers, and the number of convolution kernel was set to 128. It was finally trained by one fully connected layer, and adopted the Adam algorithm as an optimization algorithm. The learning rate was set to 0.001, and the drop rate of the dropout layer was set to 0.5. In the process of

**Table 1.** Data set

	Training set	Validation set	Test set
Malicious URL	200k	25k	25k
Normal URL	200k	25k	25k
Total	400k	50k	50k

the experiment, we adopted batch training, and each batch contained 100 pieces of data, and totally 10 rounds are trained.

We designed comparative experiments. We tried to use different network structure in experiment 1 and experiment 2. Then we adopted different embedding methods in experiment 3 and experiment 4.

### 3.3 Results and Evaluation

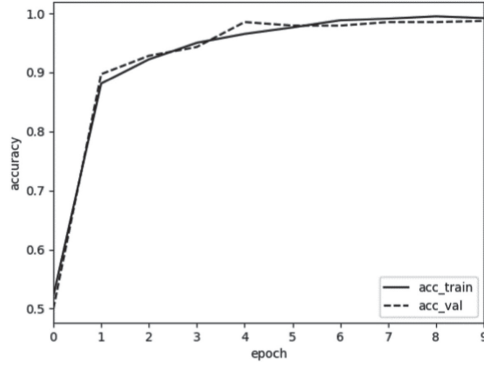
We measured the accuracy, F1-score, precision, and recall ratio to evaluate the test results. The final results of the detection model in this paper are shown in Table 2. The accuracy reaches 0.987, the precision reaches 0.993, the F1-score reaches 0.987, and the recall ratio is 0.981. The accuracy and loss during the training and verification process are shown in Fig. 2 and Fig. 3. As is shown in Fig. 2, as the number of iterations increased, the accuracy of the training increases continuously and the fitting degree of model is fairly ideal. At the same time, the loss has continued to decrease.

**Table 2.** Experimental results

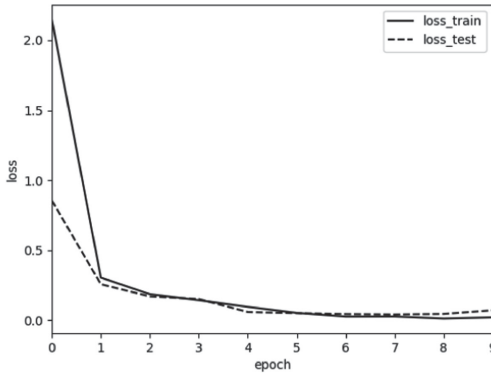
Detection Indicator	Meaning	Value
<i>accuracy</i>	$\frac{TP+TN}{P+N}$	0.987
<i>F1-score</i>	$\frac{2TP}{2TP+FP+FN}$	0.987
<i>recall</i>	$\frac{TP}{TP+FN}$	0.981
<i>precision</i>	$\frac{TP}{TP+FP}$	0.993

It can be seen that the our model has better effect than other models in the comparative experiments. High precision indicates that benign samples are less likely to be misjudged as malicious samples and intercepted. We hope that the amount of benign samples which are misjudged is as small as possible, so the high precision is required. Using the word embedding based on character embedding and the network structure of DCNN, the URL can be adequately expressed, and critical features can be extracted which can help to obtain better detection effect. Extracting different fields from the URL can make full





**Fig. 2.** Training accuracy and verification accuracy



**Fig. 3.** Training loss and validation loss

**Table 3.** Test results from experiments

NO.	Embedding method	Network structure	<i>accuracy</i>	<i>F1-score</i>	<i>recall</i>	<i>precision</i>
1	Word embedding based on character embedding	DCNN + Extracting fields	<b>0.987</b>	<b>0.987</b>	<b>0.981</b>	<b>0.993</b>
2	Word embedding based on character embedding	DCNN	0.961	0.960	0.936	0.984
3	Word embedding based on character embedding	Stacked CNN	0.958	0.959	0.976	0.942
4	Character embedding	Stacked CNN	0.923	0.926	0.964	0.890

use of keywords in the domain name, which can further improve detection accuracy and precision. In summary, the above experiments verify the feasibility of the detection model in this paper.

We also lists the test results of comparative experiments' results, as is shown in Table 2.

## 4 Conclusions

This paper aims to design a new malicious URL detection model based on deep learning. Firstly, A word embedding method based on character embedding is designed, and the vector expression of the URL is automatically learned by combining character embedding with word embedding. Secondly, we improve the deep convolution network and design a dynamic convolution network for the URL detection model. According to the length of the input vector and the depth of the current convolution layer, the parameters of the pooling layer are dynamically adjusted to extract features in a wider range automatically. Thirdly, We design the overall malicious URL detection model.

The malicious URL detection model achieves the expected effect in these experiments, which indicate that it is useful for practical application. However, considering the difference of the attack traffic between the testing environment and the real network environment. And with the development of Internet, malicious URLs are more diverse, to maintain the performance of the detection model, it is necessary to timely update the model in the actual application. Therefore, in the future, we plan to simplify the architecture of the detection model and shorten the training time while keeping the detection performance unchanged, so that the model could fit the requirements of complex application scenarios better.

**Acknowledgments.** This research was financially supported by the National Key Research and Development Plan(2018YFB1004101), Key Lab of Information Network Security, Ministry of Public Security(C19614), Special fund on education and teaching reform of Besti(jy201805), the Fundamental Research Funds for the Central Universities(328201910), China Postdoctoral Science Foundation(2019M650606), 2019 Beijing Common Construction Project-Teaching Reform and Innovation Project for Universities in Beijing, key laboratory of network assessment technology of Institute of Information Engineering, Chinese Academy of Sciences. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.



## References

1. Patil, D.R., Patil, J.: Survey on malicious web pages detection techniques. *Int. J. u-and e-Serv. Sci. Technol.* **8**(5), 195–206 (2015)
2. Yu, B., Pan, J., Hu, J., Nascimento, A., De Cock, M.: Character level based detection of DGA domain names. In: *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE (2018)
3. Choudhary, C., Sivaguru, R., Pereira, M., Yu, B., Nascimento A., De Cock M.: Algorithmically generated domain detection and malware family classification. In: *International Symposium on Security in Computing and Communication*, pp. 640–655. Springer, Singapore (2018). [https://doi.org/10.1007/978-981-13-5826-5\\_50](https://doi.org/10.1007/978-981-13-5826-5_50)

4. Garera, S., Provos, N., Chew, M., et al.: A framework for detection and measurement of phishing attacks. In: ACM Workshop on Recurring Malcode. ACM (2007)
5. Gupta, D., KM, M.: Behind phishing: an examination of phisher modi operandi. In: Usenix Workshop on Large-scale Exploits & Emergent Threats. DBLP (2008)
6. Ma, J., Saul, L.K., Savage, S., et al.: Beyond blacklists: learning to detect malicious Web sites from suspicious URLs. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28–July 1. ACM (2009)
7. Choi, H., Zhu, B.B., Lee, H.: Detecting malicious web links and identifying their attack types. In: Proceedings of the 2nd USENIX Conference on Web Application Development (2011)
8. Bartos, K., Sofka, M., Franc, V.: Optimized invariant representation of network traffic for detecting unseen malware variants. In: USENIX Security Symposium, pp. 807–822 (2016)
9. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning, p. 8. Beijing: Posts & Telecom Press, Beijing (2016)
10. Feng, Z., Shuo, C., Xiaochuan, W.: Classification for DGA-based malicious domain names with deep learning architectures. In: Second International Conference on Applied Mathematics and Information Technology, p. 5 (2017)
11. Xu, C., Shen, J., Du, X.: Detection method of domain names generated by DGAs based on semantic representation and deep neural network. *Comput. Secur.* **85**, 77–88 (2019)
12. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP (2014)
13. Zhang, X., Zhao, J., Lecun, Y.: Character-level Convolutional Networks for Text Classification (2015)
14. Wen, Z.: Research and design of malicious URL Detection algorithm on Deep learning. Beijing University of Posts and Telecommunications (2019)
15. Shibahara, T., Yamanishi, K., Takata, Y., et al.: Malicious URL sequence detection using event de-noising convolutional neural network. In: 2017 IEEE International Conference on Communications (ICC), pp. 1–7. IEEE (2017)
16. Woodbridge, J., Anderson, H.S., Ahuja, A., et al.: Predicting Domain Generation Algorithms with Long Short-Term Memory Networks (2016)
17. Dhingra, B., Zhou, Z., Fitzpatrick, D., et al.: Tweet2Vec: Character-Based Distributed Representations for Social Media (2016)



# Security-Enhanced Timed-Release Encryption in the Random Oracle Model

Ke Yuan<sup>1,2</sup>, Yahui Wang<sup>1,2</sup>, Yingming Zeng<sup>3</sup>, Wenlei Ouyang<sup>1</sup>,  
Zheng Li<sup>1</sup> (✉), and Chunfu Jia<sup>4</sup>

<sup>1</sup> School of Computer and Information Engineering, Henan University,  
Kaifeng 475004, China

yuanke\_hhhh@163.com, 2379087967@qq.com, 659695352@qq.com,  
lizheng@henu.edu.cn

<sup>2</sup> International Joint Research Laboratory for Cooperative Vehicular Networks  
of Henan, Kaifeng 475004, China

<sup>3</sup> Beijing Institute of Computer Technology and Applications, Beijing 100854, China  
yingmingblue@163.com

<sup>4</sup> College of Cyberspace Security, Nankai University, Tianjin 300350, China  
cfjia@nankai.edu.cn

**Abstract.** Timed-release encryption (TRE) is a cryptographic primitive that the sender specifies the future decryption time of the receiver. At present, most TRE schemes implement the control of decryption time based on non-interactive time server to publish time trapdoors periodically. However, the generation of a large number of time trapdoors depend on the generation of the fixed private keys of the time server, so a large number of public parameters about the private keys of the time server can then be used for cryptanalysis, which poses a great threat to the security of the private keys of the time server. To solve this problem, a concrete scheme of TRE in the random oracle model are proposed. In our scheme, time trapdoors published by the time server are generated by the private key of the time server and the random number generated in advance. Compared with the most efficient scheme in the random oracle model, our concrete scheme reduces the time consumption by about 10.8%, at the same time it has achieved the one-time-pad of the time trapdoor, which greatly enhances the security of the private key of the time server, and thus enhances the security and effectiveness of the TRE.

**Keywords:** Timed-release encryption · Security-enhanced · One-time-pad · Random oracle model

---

Supported by the National Key R&D Program of China under Grant 2018YFA0704703; the National Natural Science Foundation of China under Grant 61972215, 61802111 and 61972073; the Basic Research Plan of Key Scientific Research Projects in Colleges and Universities of Henan Province under Grant 18A413004.

© Springer Nature Singapore Pte Ltd. 2020  
Y. Xiang et al. (Eds.): SocialSec 2020, CCIS 1298, pp. 41–51, 2020.  
[https://doi.org/10.1007/978-981-15-9031-3\\_4](https://doi.org/10.1007/978-981-15-9031-3_4)

# 1 Introduction

Timed-release encryption (TRE)[18,22] is a cryptographic primitive in which the sender specifies the future decryption time. The goal is to send a message to the future, that is, the sender encrypts a message and sends it to the recipient. Any user, including the receiver, cannot decrypt the ciphertext before the specified time. Now TRE has become a basic cryptographic primitive, combined with a variety of other cryptographic primitives. It has been applied to more diversified and fine grain size scenarios to storage and access the time sensitive data. For example, the release of digital movies, the formulation and publication of business plans, online voting and so on.

Latest research shows that the TRE construction methods have expanded from mathematical problems [1–13,17,19–21,25,27,28] to physical problems [23,24] and block chain [14–16,26]. Most TRE schemes are currently constructed based on mathematical problems. Specifically, the model of non-interactive time server is mainly adopted, which is based on the bilinear pairings mathematical difficult problems. The non-interactive time server model is such that neither the sender nor the receiver interacts with the time server. The time server periodically publishes the time trapdoor, and all users in the non-interactive TRE system passively receive the time trapdoor. The receiver selects the time trapdoor corresponding to the decryption time to decrypt the ciphertext.

However, there are a lot of time trapdoor pairs about time server in the current schemes of the above non-interactive TRE model. Although it is difficult to solve mathematical problems of bilinear pairings at present, when the attacker has a large number of time trapdoor pairs, the choice of plaintext or ciphertext attack will seriously challenge the confidentiality of the time server's private key. Therefore, we attempts to propose a solution for this problem.

## 1.1 Related Work

TRE was first proposed by May [18] in 1993 and then discussed in detail by Rivest et al. [22] in 1996, which laid the foundation for TRE. Most of the previous TRE solutions fall into two categories: Time-lock puzzles [1, 16, 19, 22] and agents. Agents are mainly implemented based on time servers and are divided into interactive [11, 17, 21, 22] and non-interactive models [2–10, 12, 13, 20, 25, 27, 28]. The time server approach is initially constructed based on the quadratic residuosity problem [17], and the complexity of the problem is equivalent to the factorization problem of large integers. The subsequent schemes are almost based on bilinear pairings classes of difficult problems, such as bilinear Diffie-Hellman (BDH) class problem [2, 4–8, 10–13, 25], bilinear Diffie-Hellman inversion (BDHI) class problem [3] and bilinear Diffie-Hellman exponent (BDHE) class problem [28].

In the paper of the above non-interactive time server model, the hash function of time  $T$  uses the private key of the time server to perform an operation similar to encryption to generate the time trapdoor. As far as we know, there are a large number of plaintext-ciphertext pairs about the time server private key in the schemes of using the time server. Although it is difficult to solve mathematical

problems of bilinear pairings at present, when the attacker has a large number of time trapdoor pairs, the choice of plaintext or ciphertext attack will seriously challenge the confidentiality of the time server private key.

Quantum technique is a new field of physics and engineering. Its principle is based on the characteristics of quantum mechanics and is applied in quantum computing, quantum cryptography and quantum simulation and other fields. With the development of quantum computing technology, current cryptographic mechanisms based on various mathematically difficult problems will no longer be secure. A more secure solution to protect the time server private key is needed.

## 1.2 Our Contributions

The contributions of this paper are to re-examine the security problem caused by the heavy reuse of the time server private key and to propose a concrete scheme of security-enhanced timed-release encryption based on bilinear Diffie-Hellman problem (BDH-SE-TRE).

The plaintext  $m$  is encrypted using the key  $k$  to obtain the ciphertext  $c = E_k(m)$ ; conversely, the ciphertext  $c$  is decrypted using the key  $k$  to obtain the plaintext  $m = D_k(c)$ . Similarly, the hash function  $H(T)$  corresponding to time  $T$  is computed by using the private key  $s$  to obtain the corresponding time trapdoor  $S_T = E_s(H(T))$ , conversely, we can get  $H(T) = D_s(S_T)$ . Here,  $S_T$  corresponds to the ciphertext,  $H(T)$  corresponds to the plaintext. When the attacker has a large number of plaintext-ciphertext pairs, the security of the private key  $s$  of the time server decreases with the increase of the number of queries. If  $s$  is exposed, even though it is a difficult problem to try to recover  $H(T)$  by  $S_T$ , but due to the limited number of time trapdoors, the attacker can try according to the time sequence of the acquirement of the time trapdoors  $S_T$ , and will soon find the corresponding  $H(T)$ , causing the leak of  $H(T)$ . Therefore, once the receiver colludes with the attacker, the receiver can freely generate a time trapdoor for a specified time, so that the ciphertext can be decrypted in advance.

In our BDH-SE-TRE scheme, the time server takes a random number  $x$  corresponding to each current time, which is combined with the private key of the time server to generate the time trapdoor  $S_T = E_{(s,x)}(H(T))$ . Because each private key used to generate a trapdoor is different, it achieves one-time pad. In this way, the attacker has at most one chance to decrypt every time, even if the decryption succeeds, he will not get the private key. If the private key is exposed, the attacker is still unable to decrypt the ciphertext in advance without knowing the random number, nor can he generate a time trapdoor at any specified time. In this way, the probability that the private key of the time server being unravelled is greatly reduced and the anti-quantum requirement is satisfied, and then the ciphertext can not be decoded before the specified time arrives, which protects the plaintext.

## 2 Preliminary

In this section, we give a brief review on some cryptographic background.

## 2.1 Discrete Logarithm Problem

Let  $p, q$  be two prime numbers, the set  $F = \{0, 1, 2, \dots, p-1\}$  constitutes the finite field under multiplication, which is recorded as  $\mathbb{Z}_p$ .  $G = \{g^i : 0 \leq i \leq q-1, g \in \mathbb{Z}_p^*\}$  is the multiplicative group of the finite field  $\mathbb{Z}_p$  of order  $q$ . Every element in the group can be expressed as the power of an element  $g$ ,  $g$  is called a generator of the group.

**Definition 1.** DLP in multiplicative group of finite field. Given an element  $y \in G$ , we need to find the integer  $x \in \mathbb{Z}_q$ , which can satisfy  $y = g^x$ . This is what we called DLP in multiplicative group of finite field.

## 2.2 Elliptic Curve Discrete Logarithm Problem

Let  $p > 3$  be a prime number, elliptic curve  $y^2 = x^3 + ax + b$  over finite field  $\mathbb{Z}_p$  is a set  $E$  consisting of a special point called infinite point and all points satisfy congruence equation  $y^2 = x^3 + ax + b \pmod{p}$  in which  $a, b \in \mathbb{Z}_p, 4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ . Let  $E_p(a, b)$  denote the set

$$\{(x, y) : (x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p\} \cup \infty$$

of points on elliptic curves over finite fields. Addition in  $E_p(a, b)$  form additive groups  $\langle E_p(a, b), + \rangle$ .

**Definition 2.** ECDLP in additive group of finite field. Let  $p$  and  $q$  be two prime numbers,  $G_1 : \{kP : 0 \leq k \leq q-1\}$  be a subgroup of order  $q$  of the elliptical curve group  $\langle E_p(a, b), + \rangle$ , where  $P$  is a generator of order  $q$ . Given an element  $Q \in G_1$ , we need to find the integer  $x \in \mathbb{Z}_q$ , which satisfied  $Q = xP$ . This is called the ECDLP in additive group of finite field.

## 2.3 Bilinear Pairings Property

Different cryptography problems need to adopt different forms of bilinear pairings, which makes the mathematical description of the problem more concise and clear. The Definition 3 will give the definition of bilinear pairings that we adopt in this paper.

**Definition 3.** Bilinear pairings. Let  $G_1$  be a ECDLP additive group over a finite field,  $G_2$  be a DLP multiplicative group over a finite field, and the order of  $G_1, G_2$  is a prime number  $q$ . If the mapping  $e : G_1 \times G_2 \rightarrow G_2$  satisfies the following attributes:

- (1) Bilinear property. For any  $P, Q, R \in G_1$ , we have

$$\begin{aligned} e(P + Q, R) &= e(P, R)e(Q, R) \\ e(P, Q + R) &= e(P, Q)e(P, R) \end{aligned}$$

- (2) Nondegeneracy. If  $P$  is the generator of  $G_1$ , then  $e(P, P)$  is the generator of  $G_2$ .

(3) **Computability.** For any  $P, Q \in G_1$ , there is an efficient algorithm for computing  $e(P, Q)$ .

From the above basic bilinear properties, the bilinear properties of bilinear pairings

$$e(aP, bQ) = e(abP, Q) = e(P, abQ) = e(bP, aQ) = e(P, Q)^{ab}, a, b \in \mathbb{Z}_q^*$$

which is more commonly used in the construction of cryptographic schemes, can be derived.

By using bilinear pairings, the ECDLP in additive group of finite field can be reduced to the corresponding DLP in multiplicative group of finite field.

## 2.4 BDH Problem

Many difficult problems are constructed based on bilinear pairings, including BDH class problem, BDHI class problem and BDHE class problem. Here we only introduce the BDH problem in this paper.

**Definition 4.** BDH problem. Let  $G_1$  be a multiplicative elliptical curve group over finite field and  $P$  be the generator of  $G_1$ , given  $P, aP, bP, cP \in G_1^*$ , compute  $e(P, P)^{abc} \in G_2^*$ . If the advantage of the  $\mathcal{A}$  to solve the *BDH* problem is  $\mathcal{E}$ , then  $\Pr[\mathcal{A}(P, aP, bP, cP) = e(P, P)^{abc}] \geq \mathcal{E}$ , in which  $a, b$  and  $c$  are evenly distributed in  $\mathbb{Z}_q^*$ .

## 3 Construction of BDH-SE-TRE

In this section, we propose our concrete BDH-SE-TRE scheme and give the security assertion of our scheme.

### 3.1 Description of the Scheme

We build a non-interactive BDH-SE-TRE scheme from such a bilinear map defined above. Our BDH-SE-TRE scheme with random oracle works as follows:

**Setup.** Given a security parameter  $k$ , the algorithm outputs the system parameters  $params = \{G_1, G_2, q, e, P, H_1, H_2, n\}$ . Where  $G_1$  is an additive group,  $G_2$  is a multiplicative group, prime number  $q$  is the order of  $G_1$  and  $G_2$ ,  $e : G_1 \times G_1 \rightarrow G_2$  is a bilinear mapping satisfied the definition 3,  $P \in G_1^*$  is the generator of additive group  $G_1$  which is randomly selected by the time server, Hash function  $H_1 : \{0, 1\}^* \rightarrow G_1$ ,  $H_2 : G_2 \rightarrow \{0, 1\}^n$ , where  $n$  is the length of plaintext.

**TS-KeyGen.** The time server generates a random number  $s \in \mathbb{Z}_q^*$  as the time server's private key  $ts_{priv} = s \in \mathbb{Z}_q^*$ . Taking the public parameter  $P$  as the input, to compute the public key  $ts_{pub} = sP$  of the time server. Similarly, the time server generates the time server session private key set  $TS_{spriv} = \{x_1, x_2, \dots, x_l\} \in \mathbb{Z}_q^*$  and the corresponding time server session public key set  $TS_{spub} = \{x_1P, x_2P, \dots, x_lP\} \in G_1^*$  in sequence in the next ten years.



Where  $l \approx 175200$ , if we set the time server to generate a time trapdoor every half hour.

**User-KeyGen.** A system user generates a random number  $u \in \mathbb{Z}_q^*$  as the user's private key  $usk = u \in \mathbb{Z}_q^*$ . Taking the public parameter  $P$  as the input, to compute the public key  $upk = uP$  of the system user.

**Enc.** Given the message  $M$ , the recipient's public key  $upk_r = uP$ , the time server's public key  $ts_{pub} = sP$ , and a release time  $T \in \{0, 1\}^*$ , the sender retrieves the  $ts_{spub} = xP$  corresponding to the decryption time  $T$  which is designated by himself, and then performs the following operations:

- (1) Choose  $r \in \mathbb{Z}_q^*$  randomly, compute  $U = rP$  ;
- (2) Compute  $S_{pub} = ts_{pub} + ts_{spub} + upk = sP + xP + uP = (s + x + u)P$ ;
- (3) Compute  $K = e(rH_1(T), S_{pub}) = e(H_1(T), P)^{r(s+x+u)}$ ;
- (4) Get the ciphertext  $C$

$$C = \langle U, V \rangle = \langle rP, M \oplus H_2(K) \rangle$$

**TS-Rel.** The algorithm is implemented on the time  $T \in \{0, 1\}^*$ . The time server takes the current  $ts_{spriv} = x$  corresponding to the current time  $T$ , and generates the time trapdoor  $S_T = (s + x)H_1(T)$  .

**Dec.** Given a ciphertext  $C = \langle U, V \rangle$ , using the receiver's private key  $u$  and the corresponding time trapdoor  $S_T$  of  $T$ , the receiver performs the following operations:

- (1) Compute

$$\begin{aligned} K' &= e(U, S_T + uH_1(T)) \\ &= e(rP, (s + x)H_1(T) + uH_1(T)) \\ &= e(rP, (s + x + u)H_1(T)) \\ &= e(P, H_1(T))^{r(s+x+u)} \\ &= K \end{aligned}$$

- (2) Compute  $V \oplus H_2(K')$ , to recover message  $M$ .

If  $C$  is the correct ciphertext, then  $U = rP, V = M \oplus H_2(K)$ , in which,  $K = e(P, H_1(T))^{r(s+x+u)}$ . The decryption's correctness can be verified as follow:

$$\begin{aligned} K' &= e(P, H_1(T))^{r(s+x+u)} = K \\ V \oplus H_2(K') &= V \oplus H_2(K) \\ &= M \oplus H_2(K) \oplus H_2(K) \\ &= M \end{aligned}$$

### 3.2 Security of the Scheme

The concrete scheme above is a non-interactive BDH-SE-TRE scheme semantically secure against a chosen plaintext attack in the random oracle model.

**Theorem 1.** *If an adversary  $\mathcal{A}$  has advantage  $\epsilon$  in breaking the BDH-SE-TRE scheme above, then an challenger  $\mathcal{B}$  can be constructed to solve the BDH problem with probability at least  $\epsilon' = \epsilon/eq_sq_{H_2}$ . Where  $e$  is the base of the natural logarithm,  $q_s$  and  $q_{H_2}$  is the number of times that it is assumed that adversary  $\mathcal{A}$  can make time trapdoor queries and  $H_2$  hash function queries at most.*

We will give the rigorous proof in the full version of this paper.

## 4 Comparison and Experiments

In this subsection, we compare our BDH-SE-TRE scheme with two existing typical non-interactive server schemes: the classic scheme BC-TRE [4] proposed by Blake and Chan and the currently most efficient scheme AnTRE [3] proposed by Chalkias et al. The main advantages of the AnTRE scheme are low cost of computation and low cost of memory storage, but there is no random number in the time trapdoor, so as in other schemes, there are a large number of plaintext-ciphertext pairs about the release time.

To compare the computation time required for each scheme, we will let  $BP$  denote the bilinear pairing operation,  $PM_{ec}$  and  $PA_{ec}$  denote point multiplication and point addition operation in  $G_1$  respectively.  $Exp_{ec}$  denote exponentiation operation in  $G_2$ .  $H_1$  denotes to map a binary string of arbitrary length to  $G_1$ ,  $H_2$  denotes to map an element of  $G_2$  to a string of  $\log_2^q$  length consisting of 0 and 1,  $H_3$  denotes to map a binary string of arbitrary length to  $\mathbb{Z}_q^*$ ,  $Inv$  denotes modular inverse operation in  $\mathbb{Z}_q^*$ . We implement the above basic operations based on the MIRACL open-source library. In the process of implementation, the elliptic curve over the finite field  $F_p$  adopts a super singular elliptic curve ( $p$  is a large prime number of 512 bits, and its order  $q$  is a prime number of 160 bits). Bilinear mapping uses Tate pairing algorithm to map the above elliptic curve discrete logarithmic subgroup to the discrete logarithmic subgroup on  $F_{p^2}$  (the order is still prime  $q$ ).

Program running environment: Intel(R) Core(TM) i5-4210M CPU 2.60GHz processor, 64 bit, 8GB memory, Microsoft visual studio 2010. Running the program with 987654321 as the seed of random number. In order to make the results irrelevant to the performance of the computer, we take the time consumption of  $Exp_{ec}$  as the basic proportion, to compute the ratio of the time consumption of each operation to the time consumption of  $Exp_{ec}$ , and count the relative time consumption of each operation shown in Table 1.

The scenario we discussed is that the decryption time is known to the sender and the receiver. The sender encrypts the message at a certain time. When the decryption time arrives, the receiver will receive the time trapdoor to decrypt the ciphertext. However, the decryption time is confidential to the attacker, and the attacker can only guess the decryption time of the ciphertext.

**Table 1.** Cost of basic operations in relation to point multiplication operation.

Basic operation	Notation	Relatively time consuming
Bilinear Pairing	$BP$	3.4457
Point Multiplication in $G_1$	$PM_{ec}$	1
Point Addition in $G_1$	$PA_{ec}$	0.0072
Exponentiation in $G_2$	$Exp_{ec}$	0.3220
Modular Inverse in $\mathbb{Z}_q^*$	$Inv$	0.0030
Hash: $\{0, 1\}^* \rightarrow G_1$	$H_1$	0.3368
Hash: $G_1 \rightarrow \{0, 1\}^{\log_2^q}$	$H_2$	0.0782
Hash: $\{0, 1\}^* \rightarrow \mathbb{Z}_q^*$	$H_3$	0.0030

The TS-Rel phase of BDH-SE-TRE scheme requires the following operations: 1  $PM_{ec}$  and 1  $H_1$  to compute  $S_T = (s + x)H_1(T)$ , the total cost of the TS-Rel phase is 1.003. In the encryption phase requires the following operations: 1  $PM_{ec}$  for  $rP$ , 2  $PA_{ec}$  to compute  $S_{pub}$ , 1  $H_1$ , 1  $PM_{ec}$  and 1  $BP$  to compute  $e(rH_1(T), S_{pub})$ , 1  $H_2$  to compute  $H_2(K)$ ; The total cost of the Enc phase is 5.875. In the decryption phase, the recipient must perform 1  $H_1$ , 1  $PM_{ec}$ , 1  $PA_{ec}$  and 1  $BP$  to compute  $K' = e(U, S_T + uH_1(T))$ , 1  $H_2$  for  $M \oplus H_2(K)$ ; the total cost of the Dec phase is 4.868. Table 2 summarizes the comparison of the computational cost of BC-TRE, BDH-SE-TRE and AnTRE scheme. It should be noted that the hash functions  $H_1$  and  $H_2$  in AnTRE are roughly equivalent to the hash function  $H_3$  in Table 1, and the hash functions  $H_3$  and  $H_4$  in AnTRE are roughly equivalent to the hash function  $H_2$  in Table 1.

**Table 2.** Computation cost comparison of BC-TRE, BDH-SE-TRE and AnTRE.

Phase	Scheme		
	BC-TRE	BDH-SE-TRE	AnTRE
<i>TS - Rel</i>	$PM_{ec} + H_1 = 1.337$	$PM_{ec} + H_1 = 1.337$	$PM_{ec} + Inv + H_3 = 1.006$
<i>Enc</i>	$3BP + 2PM_{ec} + H_3 = 12.340$	$2PM_{ec} + 2PA_{ec} + H_1 + BP + H_2 = 5.875$	$4PM_{ec} + PA_{ec} + Exp_{ec} + BP + 2H_2 + 2H_3 = 7.934$
<i>Dec</i>	$BP + Exp_{ec} + H_1 + H_2 = 4.183$	$H_1 + PM_{ec} + PA_{ec} + BP + H_2 = 4.868$	$BP + PM_{ec} + 2H_2 + H_3 = 4.605$
<i>Total</i>	17.86	12.08	13.55

It can be seen from Table 2 that under the above discussion, compared with that of BC-TRE scheme, our BDH-SE-TRE scheme improves the efficiency by about 32.4%. Compared with the AnTRE scheme, our BDH-SE-TRE scheme improves the efficiency by about 10.8%. From the analysis of efficiency, the BDH-SE-TRE scheme reduces the time consumption. Moreover the scheme achieves

one-time pad about time trapdoor in terms of security, and the security performance is greatly improved compared to existing schemes. From the perspective of storage space, the proposed scheme needs to consider the storage space required by the time server at least. The time server always stores the public and private key pairs of the time server session in the next 10 years. The time server is set to release the time trapdoor every half an hour, so the total storage is  $24 \times 2 \times 365 \times 10 = 175200$ . The session private key of the time server is 160 bits each, and the session public key of the time server is the point on the elliptic curve, 1024 bits each, so the total storage space is about 24.7MB, and the time server can fully meet the storage space needs.

## 5 Conclusions

In order to enhance the security of the time server's private key in TRE, this paper proposes a concrete BDH-SE-TRE scheme in the random oracle model. In our scheme, the time server performs an "encryption-like" trapdoor generation operation at each time point using a different private key, which achieves the one-time pad about the time trapdoor, protect the time trapdoor from being exposed in advance, and ensure that the ciphertext is not decrypted in advance. In terms of efficiency, the scheme improves the time efficiency and the burden of space storage is very small.

## References

1. Bitansky, N., Goldwasser, S., Jain, A., Paneth, O., Waters, B.: Time-lock puzzles from randomized encodings. In: ACM Conference on Innovations in Theoretical Computer Science, pp. 345–356. ACM (2016)
2. Cao, H., Yuan, K., Wang, Y., Yan, Y., Zhou, L., Chai, X.: Bidding model based on timed-release encryption and blockchain. *J. Henan Univ. (Nat. Sci.)* **49**(2), 210–217 (2019)
3. Chalkias, K., Hristu-Varsakelis, D., Stephanides, G.: Improved anonymous timed-release encryption. In: Biskup, J., López, J. (eds.) ESORICS 2007. LNCS, vol. 4734, pp. 311–326. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-74835-9\\_21](https://doi.org/10.1007/978-3-540-74835-9_21)
4. Chan, A.F., Blake, I.F.: Scalable, server-passive, user-anonymous timed release cryptography. In: . Proceedings 25th IEEE International Conference on Distributed Computing Systems. ICDCS 2005, pp. 504–513. IEEE (2005)
5. Chen, W., Wang, Y., Qin, Z., Liu, X.: Research on timed access of sensitive data based on dual encryption. *J. Univ. Electr. Sci. Technol. China* **46**(3), 588–593 (2017)
6. Cheon, J.H., Hopper, N., Kim, Y., Osipkov, I.: Provably secure timed-release public key encryption. *ACM Trans. Inf. Syst. Secur.* **11**(2), 1–44 (2008)
7. Dent, A.W., Tang, Q.: Revisiting the security model for timed-release encryption with pre-open capability. In: Garay, J.A., Lenstra, A.K., Mambo, M., Peralta, R. (eds.) ISC 2007. LNCS, vol. 4779, pp. 158–174. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-75496-1\\_11](https://doi.org/10.1007/978-3-540-75496-1_11)

8. Fan, C.I., Chen, J.C., Huang, S.Y., Huang, J.J., Chen, W.T.: Provably secure timed-release proxy conditional reencryption. *IEEE Syst. J.* **11**(4), 2291–2302 (2017)
9. Fujioka, A., Okamoto, Y., Saito, T.: Generic construction of strongly secure timed-release public-key encryption. In: Parampalli, U., Hawkes, P. (eds.) *ACISP 2011*. LNCS, vol. 6812, pp. 319–336. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-22497-3\\_21](https://doi.org/10.1007/978-3-642-22497-3_21)
10. Hong, J., Xue, K., Xue, Y., Chen, W., Wei, D.S., Yu, N., Hong, P.: TAFC: Time and attribute factors combined access control for time-sensitive data in public cloud. *IEEE Trans. Serv. Comput.* **13**(1), 158–171 (2020). <https://doi.org/10.1109/TSC.2017.2682090>
11. Huang, Q., Yang, Y., Fu, J.: Secure data group sharing and dissemination with attribute and time conditions in public cloud. In: *IEEE Transactions on Services Computing ( Early Access ) PrePrints*, p.1 (2018). <https://doi.org/10.1109/TSC.2018.2850344>
12. Huang, S., Fan, C.I., Tseng, Y.: Enabled/disabled predicate encryption in clouds. *Future Gener. Comput. Syst.* **62**, 148–160 (2016)
13. Hwang, Y.H., Yum, D.H., Lee, P.J.: Timed-release encryption with pre-open capability and its application to certified e-mail system. In: Zhou, J., Lopez, J., Deng, R.H., Bao, F. (eds.) *ISC 2005*. LNCS, vol. 3650, pp. 344–358. Springer, Heidelberg (2005). [https://doi.org/10.1007/11556992\\_25](https://doi.org/10.1007/11556992_25)
14. Jia, L., Garcia, F., Ryan, M.: Time-release protocol from bitcoin and witness encryption for sat. *Korean Circulation J.* **40**(10), 530–535 (2015)
15. Li, C., Palanisamy, B.: Decentralized release of self-emerging data using smart contracts. In: *2018 IEEE 37th Symposium on Reliable Distributed Systems*, pp. 213–220. IEEE (2018). <https://doi.org/10.1109/SRDS.2018.00033>
16. Liu, J., Jager, T., Kakvi, S.A., Warinschi, B.: How to build time-lock encryption. *Des. Codes Cryptography* **86**(11), 2549–2586 (2018). <https://doi.org/10.1007/s10623-018-0461-x>
17. Marco Casassa, M., Keith, H., Martin, S.: The hp time vault service: Exploiting ibe for timed release of confidential information. In: *Proceedings of the 12th International Conference on World Wide Web*, pp. 160–169. ACM (2003)
18. May, T.: Timed-release crypto. Unpublished manuscript (1993). <https://www.mysite.org>
19. Mahmoody, M., Moran, T., Vadhan, S.: Time-lock puzzles in the random oracle model. In: Rogaway, P. (ed.) *CRYPTO 2011*. LNCS, vol. 6841, pp. 39–50. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-22792-9\\_3](https://doi.org/10.1007/978-3-642-22792-9_3)
20. Namasudra, S.: An improved attribute-based encryption technique towards the data security in cloud computing. *Concurrency Comput. Pract. Experience* **31**(9), e4364 (2017). <https://doi.org/10.1002/cpe.4364>
21. Patil, S.Y., N, A.J.: Conjunctive keyword search with designated tester and timing enabled proxy reencryption in health cloud. *Int. J. Innovative Res. Sci. Technol.* **4**(3), 78–85 (2017)
22. Rivest, R.L., Shamir, A., Wagner, D.A.: Time-lock puzzles and timed-release crypto. Technical Report MIT/LCS/TR-684, MIT LCS Tech, Cambridge, MA (1996)
23. Unruh, D.: Revocable quantum timed-release encryption. In: Nguyen, P.Q., Oswald, E. (eds.) *EUROCRYPT 2014*. LNCS, vol. 8441, pp. 129–146. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-642-55220-5\\_8](https://doi.org/10.1007/978-3-642-55220-5_8)
24. Wang, T., He, Y., Li, L.: New timed-release encryption based on indistinguishability obfuscation. *Appl. Res. Comput.* **34**(9), 2795–2798 (2017)

25. Watanabe, Y., Shikata, J.: Timed-release computational secret sharing and threshold encryption. *Des. Codes Cryptography* **86**(1), 17–54 (2017). <https://doi.org/10.1007/s10623-016-0324-2>
26. Wei-Jr, L., Chih-Wen, H., Ja-Ling, W.: A fully decentralized time-lock encryption system on blockchain. In: 2019 IEEE International Conference on Blockchain, pp. 302–307. IEEE (2019)
27. Xiong, J., Li, F., Ma, J., Liu, X., Yao, Z., Chen, P.S.: A full lifecycle privacy protection scheme for sensitive data in cloud computing. *Peer-to-Peer Networking Appl.* **8**(6), 1025–1037 (2014). <https://doi.org/10.1007/s12083-014-0295-x>
28. Yuan, K., Liu, Z., Jia, C., Yang, J., Lv, S.: Public key timed-release searchable encryption in one-to-many scenarios. *Acta Electronica Sinica* **43**(4), 760–768 (2015)



# Generating Adversarial Malware Examples with API Semantics-Awareness for Black-Box Attacks

Xiaowei Peng, Hequn Xian<sup>(✉)</sup>, Qian Lu, and Xiuqing Lu

Qingdao University, Qingdao 266071, China  
xianhq@qdu.edu.cn

**Abstract.** As deep learning plays a key role in malware detection, it is of great practical significance to study adversarial malware examples to evaluate the robustness of malware detection algorithm based on deep learning. A black-box attack is performed while malware authors are allowed only access to the input and output of the malware detection model. Due to the transferability of deep learning model, it is an effective way to train a substitute model to fit the black-box model. Generative adversarial network based models show good performance in generating adversarial examples and training substitute models. However, because of the discrete output, generative adversarial networks are unable to compute gradient for their back-propagation, which makes it difficult to update the weights of the neural network in the training process. In addition, APIs are important features in representing malware, but their potential semantic features are usually ignored. To address the above problems, a generative adversarial network based algorithm with API word embedding method is designed, which adopts CNN structure to train a substitute model. The substitute model is utilized to analyze the semantic association of sequential API calls. Then, we employ a long short-term memory framework to generate antagonistic examples. The experimental results show that the proposed scheme is efficient and effective.

**Keywords:** Adversarial malware examples · Black-box attack · Generative adversarial network(GAN) · API word embedding · Long short-term memory(LSTM)

## 1 Introduction

With the booming of Social Networks, tremendous amount of data is being produced world-widely. New methods and new software tools are continuously developed and dispersed to capture, manage, and process Big Data. Various categories of malwares have raised security issues and posed critical challenges for Social Networks and Big Data applications. Malware detection methods and techniques are considered to be of great importance. Deep learning has been widely used in computer vision, natural language processing and many other application realms [1, 2]. In malware detection, deep learning models outruns traditional tools with its outstanding ability to recognize the essential

features of examples and classify them effectively. Powerful as they are, deep learning models is vulnerable to disturbance from slightly tempered input [3, 4]. Adversarial examples, generated by adding perturbations to normal examples, can easily compromise the working of a deep learning model. They reveal basic blind spots in deep learning algorithms [5].

Researchers try to generate adversarial examples through countermeasure techniques. In some early research works, gradient based methods were used to generate adversarial examples, which fool the detection model through exemplifying the prediction error of the neural network and optimizing the input [7]. However, all these researches are carried out based on a fundamental assumption that malware authors have been granted full access privileges to the target model and can train detection models of their own. Actually, under most circumstances, malware detection algorithms based on deep learning are integrated into an antivirus software or deployed in the cloud. From the perspective of malware authors, the malware detection model is a black box without any information about the internal structure.

With the transferability of adversarial examples, an adversary in a confrontational attack can compromise the target detection model by simply training and attacking a substitute model. This type of attack is also recognized as the black-box attack [7]. Black-box attacks are usually built with supervised learning mechanisms [8]. The substitute model is trained with examples labeled by the target model. Then, adversarial examples can be generated according to the substitute model which is fully transparent to the attacker. However, once the algorithm of the target model is updated or replaced, malware authors will have to train another substitute model. There is another substitute model training method which is based on the generative adversarial network (GAN) [6]. It trains the substitute model with help of the multi-cognitive network within GAN, which trains a generation model simultaneously. The generation model and the substitute model train each other and account for each, as stated in the game theory, until a Nash equilibrium can be achieved. Eventually adversarial examples generated by the generation model can fool the substitution model. The GAN based model can enter a new round of adversarial training once the algorithm in the target model is changed. However, API is adopted as the character of malware examples, and sparse vectors are used to represent APIs in a software program [9], which is directly fed into the multi-cognitive network. The Euclidean distance between APIs is rendered identical, and the potential semantic features between APIs are overlooked [10].

According to the problems above, we propose a generative adversarial network model and adopt the method of word vector expression in natural language processing [11, 12], so that the neural network can extract characters from the API word list. Convolutional neural network (CNN) is used to train the substitute model, which analyzes the semantic association of sequential API calls, and explains the latent semantic features obtained by APIs. Then, we employ a long short-term memory framework to generate antagonistic examples.

The contribution of this paper can be summarized as follows:

- We devised a method to map the behavior of API calls into a vector space, so that the connection between APIs can be evaluated through measurement. The behavior pattern of the malwares can be effectively analyzed with machine learning method.



- We design a deep learning model that can extract the semantic association of APIs, which can make the most out of the semantic features to produce dedicated malware API sequences.

## 2 Adversarial Example Generation

### 2.1 Black-Box Attack and Substitution Model

From the perspective of an attacker, black-box attack may be the most challenging situation. Black-box attack can be divided into ordinary black-box attack and completely restricted black-box attack according to the degree of the restrictions [7]. Access to the input and output are allowed in an ordinary black-box attack, while in a completely restricted black-box attack, no information can be acquired. Neither kind of black-box attack can use the back propagation of the target model, so the question falls in the attack transferability from the self-trained model to the restricted black-box model [3]. Due to the enormous challenge of the completely restricted black-box attack, it falls out of the scope of this paper. In fact, attackers do have the right to inquire about the target model to obtain useful information for generating adversarial examples in many scenarios. For example, malware classifiers (usually trained by deep neural networks) allow attackers to input any binary file and they output the classification results, such as confidence score or category of classification. Then the attacker can use the classification results to design more effective adversarial examples to fool the target classifier. In this kind of black-box attack, the back propagation for gradient calculation of the target model is still prohibited, because the back propagation needs to understand the internal configuration of the target model, which is not available for a black-box. A better way to realize the attack transferability of the adversarial examples is to use the ability of free query to train the substitute model [13, 14]. Then, any white-box attack technology can be used to attack the substitute model, and the generated adversarial examples can be used to attack the target model.

The main advantage of the training a substitute model is that it is completely transparent to the attacker [15]. Therefore, the basic attack process of the target model, such as the back propagation of gradient calculation, can be achieved with the substitute model which is used to generate adversarial examples.

### 2.2 API Word Vectors

Malware programs call different APIs to implement corresponding malicious actions. We focus on the API call types distinguished from the naming, regardless of their parameters and return values. It can be seen that each API is represented by a number of English words or character combinations, which is similar to the text analysis problem in natural language processing [11]. Therefore, we try to extract all the API calls, regard the API sequence as a text sequence, and use the technology of word vectors to digitize each API.

First, we regard each API in the API sequence as an independent “word”, and generate a word vector for it. A complete API sequence is denoted as  $W$ . In order to facilitate the

learning with neural networks, we set the length of each API sequence to the maximum API sequence length  $T$  in the sample set,  $T$  is the number of APIs contained in the sequence. For a sequence shorter than  $T$ , we make tail padding with 0. The sequence  $W$  composed of  $T$  API “words”  $w_t (1 \leq t \leq T)$  is represented as:

$$W = (w_1, w_2, \dots, w_T) \quad (1)$$

Each API “word”  $w_t$  is then encoded using one-hot encoding. Suppose that the size of the input API dictionary is  $V_I$ . In this way, an API to one hot encoding mapping can be established. Each API uses a  $V_I$  dimensional one-hot vector representation, that is,

$$w_t = (o_1, o_2, \dots, o_{V_I})^T \quad (2)$$

There are  $t$  ones and the rest of the elements are zeros. If the one-hot encoded vector is directly feed into the deep learning model for training, the number of model parameters will be huge and extremely sparse [10]. Moreover, the one-hot encoding ignores the semantic relationship between words. In natural language processing, the one-hot encoded character text sequence needs to go through the word embedding layer [16] to generate the word vector whose dimension is much lower than  $V_I$ .

We feed the one-hot code of the API sequence into the word embedding layer, then obtaining a word embedding matrix  $W_e \in R^{V_I \times K}$  which is a matrix of parameters. According to the classification label of the API sequence, a word vector can be acquired by supervised learn, in which  $W_e$  is the table of word vectors. Each line of  $W_e$  is the  $K$  dimension word vector  $x_t (1 \leq t \leq T)$  of every API.

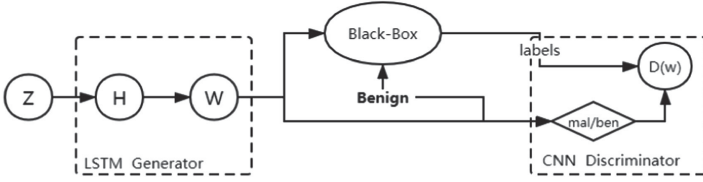
$$x_t = w_e^T w_t \quad (3)$$

Given the one-hot vector of any API word in an API sequence,  $o_t = 1, o_i = 0 (1 \leq i \leq T, i \neq t)$ , the word vector of the API is the  $t^{\text{th}}$  line of  $W_e$ . A word sequence  $X$  can be acquired from an API call sequence  $W$ .

$$X = (x_1, x_2, \dots, x_T) \in R^{K \times T} \quad (4)$$

### 2.3 Model Architecture

Our designing goal is to capture the dynamic behavior characteristics of malware calling API, and to reveal the collaborative relationship between API calling sequence and malicious purpose from a large number of training data. Our model consists of three components, the LSTM generator, the CNN substitute detector and the black-box. The architecture is shown in Fig. 1.



**Fig. 1.** The architecture of adversarial malware examples generation model.

### LSTM Generator

The generator is based on long short-term memory network, known as LSTM. An LSTM unit is made of an input gate  $i_t$ , an output gate  $o_t$ , and a forget gate  $f_t$ . The input gate controls the extent of the new input value, the forget gate is used to manage how much a value should be left in the unit, the output gate decides which values are used to activate the LSTM unit. Repeated LSTM units can record some status of each moment, which in our design is the position of each API in the generated API sequence. An API sequence is synthesized by the LSTM generator with random noise vectors. Every LSTM unit is determined by its previous hidden status  $h_{t-1}$ , previous output  $y_{t-1}$  and the noise vector  $z$ .  $i_t$  and  $f_t$  are computed and then the current unit  $c_t$  and current hidden status  $h_t$  are computed. Functions within a LSTM unit is defined as follows.

$$i_t = \sigma(W_i y_{t-1} + U_i h_{t-1} + C_i z) \quad (5)$$

$$f_t = \sigma(W_f y_{t-1} + U_f h_{t-1} + C_f z) \quad (6)$$

$$o_t = \sigma(W_o y_{t-1} + U_o h_{t-1} + C_o z) \quad (7)$$

$$\tilde{c}_t = \tanh(W_c y_{t-1} + U_c h_{t-1} + C_c z) \quad (8)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \quad (9)$$

$$h_t = o_t \otimes \tanh(c_t) \quad (10)$$

$W_{\{i,f,o,c\}}$ ,  $U_{\{i,f,o,c\}}$  and  $C_{\{i,f,o,c\}}$  are weight matrices,  $\sigma(\cdot)$  is sigmoid function,  $\tanh(\cdot)$  is hyperbolic tangent function,  $\otimes$  stands for multiplication of elements. To simplify the computation, we omit the deviation term. Similar to the seq2seq (sequence to sequence) model [17], we use LSTM as the decoding network. Noise  $z$  is used as the input vector to control the generation of each API, which has the same dimension as the API word vector.

### Substitute Detector

The substitute detector is constructed with convolutional neural network (CNN), which is supposed to simulate the target black box. An API sequence of length  $T$  contains  $T$  consecutive word vector, it goes through the CNN and produces the prediction result. In

the convolution layer, the hidden patterns in different API sequences can be found automatically by the convolution window sliding on the word vector sequence and detecting the features in different positions. A window is set up at the position of each kernel word in the API sequence. The window size is the same as that of the convolution kernel. That is to say, convolution operation is performed on consecutive  $k$  APIs to generate a feature map. In the convolution neural network, multiple convolution kernels are often used at the same time. Here we record the number of convolution kernels as  $n_c$ . In this way, each convolution kernel generates a dimension's characteristic graph, and  $n_c$  convolution kernels will generate  $n_c$  dimension's characteristic graph, and connect them by columns, and finally get the characteristic graph set of API sequence.

### Training

In the training processes, the generated API sequence and the benign API sequence are used to train the substitute detector together with the labels given by the black box. By optimizing the loss function, the substitute detector simulates the black box, and the same classification is given to the input API sequence as the black box.

The ultimate goal of the generator is to generate malware API sequences, send them to the substitute detector for detection, and let them be recognized as benign API sequences.

By optimizing the loss function, the probability of generated malware API sequences being recognized as malicious is reduced. When the adversarial training between the generator and the substitute detector reaches a balance point [7], the malware API sequence generated by the generator can fool both the substitute detector and the target black box simulated.

## 3 Experiments

Our experiments are implemented with Keras and Tensorflow. Keras is a high-level deep learning programming framework, integrating the implementation of a variety of neural networks. Based on python programming language and tensorflow backend, Keras supports accelerated training model on GPU.

### 3.1 Dataset and Configuration

We collected 6946 malware from the malware sample website ZOO and some open source projects of malware. The benign software comes from different types of common applications, such as complete free software, Softonic, Microsoft Windows system files, etc., totaling 2749. We use APIs as the dynamic characteristics of the samples. Cuckoo is chosen as the sandbox tools, which can extract malware behavior data such as Windows API call sequence.

We set the virtual machine VMware 15.5 in Ubuntu 16.04. The malicious samples are executed within Windows 7 through the virtual box. The running time threshold is 2 min. According to the output JSON file, we extract the API call sequence of the program. In order to offset the unnecessary behaviors on the call data and increase the generalization ability of the training model on other data sets, it is necessary to de

duplicate the continuous repeated API calls. After deduplication, we truncate the length of the sequence used for the experimental data to 400, and shorter API call sequences are padded with zeros. The API call sequence is represented as 400 \* 300 dimensional API word vector matrix by word embedding.

For all of the sequence of the API calls dataset, we split 80% of the dataset as the training set and the remaining 20% as the test set. Then we randomly select 25% of the training set as the validation set.

Our experiments are carried out on a GPU workstation, which is equipped with NVIDIA Tesla V100 GPU, Intel Xeon 10 core CPU, 64 GB memory, CUDA 9.0 and cuDNN7.1 library, Cuckoo Sandbox with Windows7 X64.

Black box classification algorithm is usually not open-source to the public, and there is no tracking version available for experiments. The deep learning black-box malware classifier based on API call is difficult to obtain in the actual application scenarios. Therefore we train the malware classification algorithm separately as the black-box classifier needed in the experiment.

### 3.2 Evaluation

We create our own black-box malware classifier for the adversarial training process, which also allows us to evaluate the attack performance against many classifier types. The input of all the classifiers is a vector of 400 API calls in word embedding, each with dimension of 300. The output is binary: malicious or benign.

#### Black-Box Classifier performance

For all neural networks, we use the Adam optimizer. The output layer is fully-connected with sigmoid activation function, and a rectified linear unit RELU is chosen as the activation function of input layer and hidden layer due to its fast convergence compared with sigmoid () or tanh (). Dropout is used to improve the generalization potential of the network. We conduct training for a maximum of 100 epochs, but convergence is usually reached after 15-20 epochs, which depends on the type of classifier. Batch size of 256 samples is used.

We measured the performance of the classifiers using the accuracy ratio on the test set. The performance of all the black-box classifiers is shown in Table 1.

**Table 1.** Classifier performance

Classifier Type	Accuracy (%)	Classifier Type	Accuracy (%)
LR	90.59	MLP	95.08
DT	90.55	CNN	95.04
RF	92.49	LSTM	94.96
SVM	90.51	BiLSTM	95.39

As can be seen in Table 1, the classifier based on deep learning have a good performance in detecting malwares. BiLSTM is one of the classifier most resistant to our proposed attack based on the semantic features of API sequence.

**Attack Performance of Generated API Sequence** We evaluate the efficiency and effectiveness of our scheme by comparing it with Hu’s work on true positive rate (TPR), which is the percentage of the number of malicious examples detected and the number of all examples samples. After adversarial attack, the reduction of TPR can effectively reflect the ability of adversarial examples to successfully bypass the black-box detector. The result is shown in Table 2.

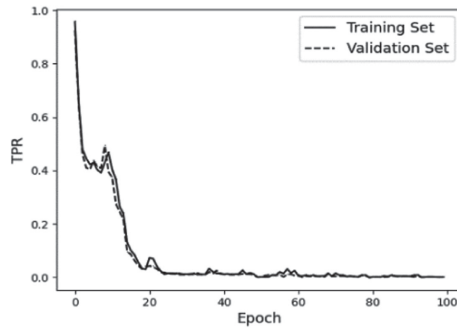
**Table 2.** True positive rate on original malware samples and adversarial examples for both the train set and test set.

Classifier Type	Training Set		Test Set	
	Hu et al. (%)	Generated (%)	Hu et al. (%)	Generated (%)
LR	0.00	0.00	0.00	0.00
DT	0.16	0.10	0.16	0.12
RF	0.20	0.18	0.19	0.18
SVM	0.00	0.02	0.00	0.02
MLP	0.00	0.00	0.00	0.00
CNN		0.00		0.00
LSTM		0.01		0.01
BiLSTM		0.06		0.06

As can be seen in Table 2, we achieve comparable attack results to five black-box classification algorithms used in Hu’s proposed MALGAN. Although the TPR is not decreased completely to zero for SVM, 0.02% is enough to be ignored. For random forest and decision trees, those are quite different with the structure of neural networks, our proposed attack is able to decrease the TPR on generated adversarial examples to the range of 0.10% to 0.18% for both the training set and the test set. In addition, for the deep learning based three classifiers, the TPRs is also reduced to nearly zero, while the malware detection accuracy ratio on the original samples range from 94.96% to 95.39%.

We also plot the convergence curve of the TPR on the training set and the validation set during the training process of our work, with using random forest as the black-box classifier. The result is shown in Fig. 2.

As can be seen from the result, the TPR in our scheme decreases dramatically with the increasing of the training epoch. The overall performance of our scheme is as good as that in Hu’s work, while the decreasing rate is greater than the opponent due to our design that reveal more of the semantic association of APIs.



**Fig. 2.** The change of the true positive rate on the training set and the validation set over time.

## 4 Conclusions

In this paper, we study the problem of generating adversarial malware examples for black-box attacks. Traditional generative adversarial network based models are unable to compute gradient for their back-propagation due to the discrete output, which makes it difficult to update the weights of the neural network in the training process. Potential semantic features of APIs are usually ignored in existing schemes. A novel generative adversarial network based algorithm with API level word embedding method is proposed, which adopts CNN structure to train a substitute model, which is utilized to analyze the semantic association of composite API calls and sequential API calls. It can reveal the latent semantic features obtained by APIs more extensively. We employ a long short-term memory framework to generate antagonistic examples to enhance the utilization of semantic feature information between APIs. Experimental results show that the proposed scheme is efficient and effective.

## References

1. Kai, Z., Shifei, D.: Advances in image super-resolution reconstruction. *Comput. Eng. Appl.* **53**, 29–35 (2017)
2. Lin, Y., Han, X., Xie, R., Liu, Z., Sun, M.: Knowledge Representation Learning: A Quantitative Review (2018). arXiv e-prints [arXiv:1812.10901](https://arxiv.org/abs/1812.10901)
3. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks (2013). arXiv preprint [arXiv:1312.6199](https://arxiv.org/abs/1312.6199)
4. Demetrio, L., Biggio, B., Lagorio, G., Roli, F., Armando, A.: Explaining vulnerabilities of deep learning to adversarial malware binaries (2019)
5. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2014). arXiv preprint [arXiv:1412.6572](https://arxiv.org/abs/1412.6572)
6. Hu, W., Tan, Y.: Generating adversarial malware examples for black-box attacks based on gan (2017). arXiv preprint [arXiv:1702.05983](https://arxiv.org/abs/1702.05983)
7. Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.-J.: Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26 (2017)

8. Duarte-Garcia, H.L., Morales-Medina, C.D., Hernandez-Suarez, A., Sanchez-Perez, G., Toscano-Medina, K., Perez-Meana, H., Sanchez, V.: A Semi-supervised learning methodology for malware categorization using weighted word embeddings. In: 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 238–246. IEEE (2019)
9. Johnson, R., Zhang, T.: Supervised and semi-supervised text categorization using one-hot LSTM for region embeddings. *Stat.* 1050 (2016)
10. Du Peng, D.S.: A DGA domain name detection method based on deep learning models with mixed word embedding. *J. Comput. Res. Dev.* **57**, 433 (2020)
11. Zhang, Y., Gan, Z., Carin, L.: Generating text via adversarial training. In: NIPS workshop on Adversarial Training (2016)
12. Weigelt, S., Landhäußer, M., Blersch, M.: How to Prepare an API for Programming in Natural Language (2019)
13. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 506–519 (2017)
14. Papernot, N., McDaniel, P., Goodfellow, I.: Transferability in machine learning: from phenomena to black-box attacks using adversarial samples (2016). arXiv preprint [arXiv:1605.07277](https://arxiv.org/abs/1605.07277)
15. Martins, N., Cruz, J.M., Cruz, T., Abreu, P.H.J.I.A.: Adversarial machine learning applied to intrusion and malware scenarios: a systematic review. *IEEE Access* **8**, 35403–35419 (2020)
16. Wang, S., Zhou, W., Jiang, C.: A survey of word embeddings based on deep learning. *Computing* **102**(3), 717–740 (2019). <https://doi.org/10.1007/s00607-019-00768-7>
17. Wang, Z., Liu, Z., Chen, Z., Hu, H., Lian, S.: A neural virtual anchor synthesizer based on seq2seq and gan models. In: 2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), pp. 233–236. IEEE (2019)





# An Elastic Data Processing Method Based on Data-Center-Platform

Zhang Pan<sup>1</sup>, Lai Fenggang<sup>1</sup>, Du Jing<sup>1</sup>, Ying Zhangchi<sup>2</sup>, Kong Rui<sup>1</sup>, Zhou Yi<sup>1</sup>,  
and Yu Xiao<sup>3,4</sup>✉

<sup>1</sup> State Grid Information and Telecommunication Co., Ltd, Beijing 100761, China

<sup>2</sup> State Grid Zhejiang Electric Power Co., LTD, Hangzhou 310000, China

<sup>3</sup> Department of Computer Science and Technology, Shandong University of Technology,  
Zibo Shandong 255022, China  
Yuxiao8907118@163.com

<sup>4</sup> School of Computer Science and Technology,  
Beijing Institute of Technology, Beijing 100081, China

**Abstract.** This paper analyzes the business value and application prospect of data-center-platform as a big data service platform. Based on the existing technology, it describes the organizational structure and data management process of data-center-platform, and studies the business interaction logic and construction elements of data-center-platform. The goal of data-center-platform is to build the center of data sharing in the whole domain, and to provide the services of data collection, data extraction, data service and provide a data intelligent platform for business applications. We analyze the problem of database synchronization in isolated networks faced by data-center-platform during data collection, and the existing database replication technology in detail, and design the process of maintaining data consistency in the same or different databases between different sites in a certain period of time. Finally, we take an equipment data integration system as an example to analyze the specific application of data-center-platform.

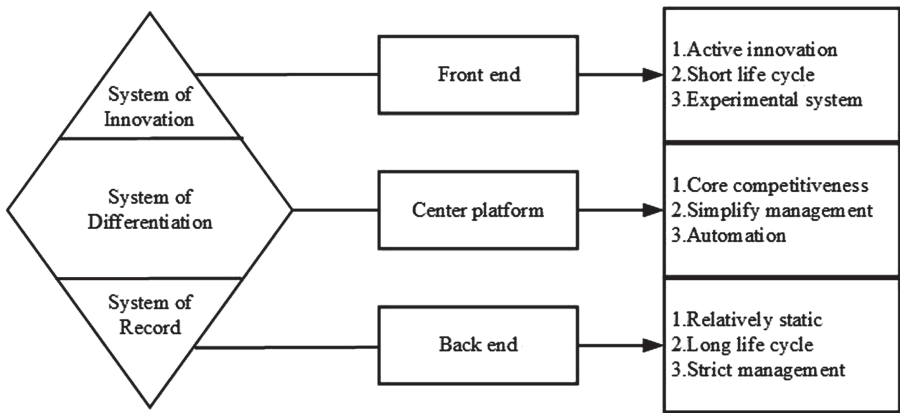
**Keywords:** Data-Center-Platform · Big data processing · Database synchronization · Isolated network

## 1 Introduction

Data-center-platform is a big data platform (service platform), it provides the function of collecting, cleaning, managing and analyzing big data, supports the standardization and rapid customization of business applications, helps to reduce the I/O throughput of data, the possibility of unnecessary data redundancy and data errors, achieves multiplexing

of calculation results, and improves data usage efficiency. The essence of data-center-platform is to build the center of global data sharing, to provide data collection, data extraction, data services and other integrated services, and to provide data intelligent platform for business applications [1]. The data-center-platform is closer to the specific business, provides faster services for the business, and builds on the existing data platform and data warehouse. So, it can be regarded as the middle layer from the specific data to the business value realization process.

Data-center-platform is a capability platform between the innovative front-end system and the recording back-end system. It has the ability to precipitate differentiation and the ability to accelerate the response of the front-end business, provide support for active and innovative front-end business, and improve the response speed of enterprises to market changes, as shown in Fig. 1.



**Fig. 1.** The conceptual diagram of data-center-platform

Data-center-platform is based on the aggregation of multiple types of large amounts of data and supported by a cloud platform. Through rich data tags, it provides unified data services for front-end applications and efficient data services for data analysis applications. Data-center-platform includes the data technology layer, unified data layer, unified service engine, data asset management, and data operation management [2]. The overall architecture is shown in Fig. 2.

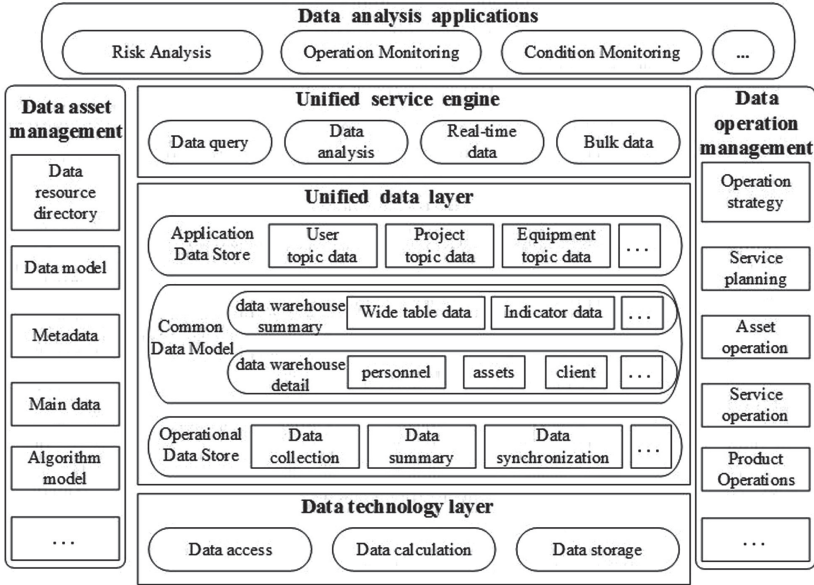


Fig. 2. The overall architecture diagram of data-center-platform

## 2 Data Synchronization of Isolated Network

Data collection and summarization are the foundation for realizing data-center-platform, which determines the composition of upper-layer business applications. Data synchronization in isolated network is one of the key issues that should be solved in data collection and summarization services. Therefore, in this section we discuss how to realize data synchronization in isolated network based on data-center-platform.

When the internal and external networks are completely isolated, it can protect the internal network from attacks that from the external network, can delimit a clear security boundary for the internal network, can enhance the controllability of the network, and facilitate internal management. However, in this kind of network environment, if there is no data exchange, the network application will be seriously affected.

Therefore, the information exchange between internal and external networks, especially the exchange of database information, has become an urgent problem to be solved. In addition to the extensive application of current database products, it can be said that where there is network isolation, there is a need for database synchronization.

Database synchronization refers to the process of keeping data consistent in a certain period of time between homogeneous or heterogeneous databases in different sites [3]. There are many methods of database synchronization, such as table replication, master-slave replication, peer-to-peer replication and cascading replication.

### 2.1 Database Synchronization Process

During the synchronization process from the source database to the target database, the change information captured from the source database is first saved in a certain

location, then forwarded to the site where the target database is located, and applied to the target database, so the whole process can be divided into the following three steps: change capture, data distribution and data update [4], they constitute the three functional modules of the database synchronization process, as shown in the following Fig. 3

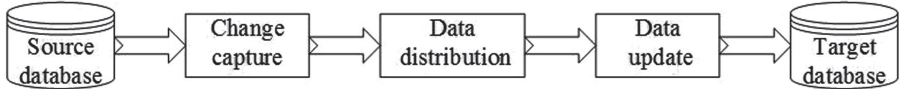


Fig. 3. Database synchronization process

### Change Capture Technology.

The change capture of synchronization objects is the basis of data synchronization, which directly determines the update and timing methods of data synchronization. There are many ways to capture change data, which can be summarized into six basic methods, namely snapshot method, trigger method, timestamp method, log method, API method and shadow table method [5].

### Data Distribution Technology.

Distribution, also known as propagation, is responsible for sending and receiving data between nodes. It transmits the change information of the original database from the source node to the target node. Generally, it can be divided into three types: push type, pull type and push-pull combination.

If the source node is responsible for the distribution task, the source node pushing the communication content into the target node, which is the “push” model [6].

If the target node is responsible for the distribution task, it means that the target node applies for the communication content from the source node, which is the “pull” model.

Compared with the two models, the advantage of push model is high efficiency, while pull model is easy to schedule. These differences will become more obvious as the number of nodes increases. If the distribution task is not undertaken by the source node or the target node, but by the third node, it constitutes a “push-pull combination” model.

### Data Update Technology.

Data update refers to modifying the target database according to the change sequence information of the synchronization object, so that the target database obtains a state consistent with the source database. The updated data content can be a complete copy of the synchronization object, a sequence of changes, or a net change.

For the heterogeneous database synchronization system, data transformation must be carried out during the process of data updating, so that the change sequence transmitted from the source database can be applied to the heterogeneous target database. The following mainly consider the data transformation ideas between relational heterogeneous databases.

Different databases define different data types. One of the key problems of data conversion between the two databases is to solve the matching problem of various data

types. Firstly, we must determine the data definition and corresponding relationship of the two kinds of database management systems, and then consider how to realize data conversion.

Generally, there are two ways to realize this conversion [7]. One is direct conversion, that is, to define the direct conversion mapping relationship between the two database data types, directly convert one database data type to another according to the mapping relationship. The second is indirect conversion, which is converted by third-party data types.

The advantages of direct conversion are fast conversion speed, high conversion accuracy, and the disadvantage is poor extensible. The advantage of indirect conversion is that it is easy to realize the conversion between heterogeneous databases, which requires the least conversion modules and has strong scalability. The disadvantage of indirect conversion is that the conversion time is longer and the conversion accuracy is reduced due to the increase of conversion times.

### 3 Case Analysis

In this section, we take an equipment data integration system as an example to analyze the specific application of data-center-platform. In order to avoid data inconsistency and realize standardized and reusable data integration, a data integration architecture based on data-center-platform is designed by the system in combination with business practice, as shown in Fig. 4.

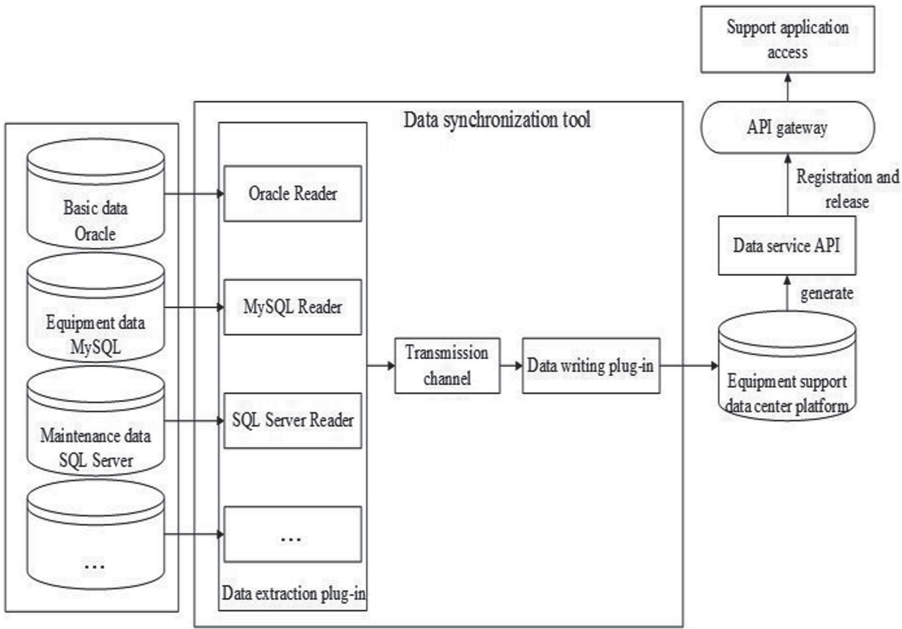


Fig. 4. Equipment support data integration architecture

### 3.1 Data Model Hierarchy

The data table in the data-center-platform can be divided into three levels: operation data store, common data model and application data store, as shown in Fig. 5.

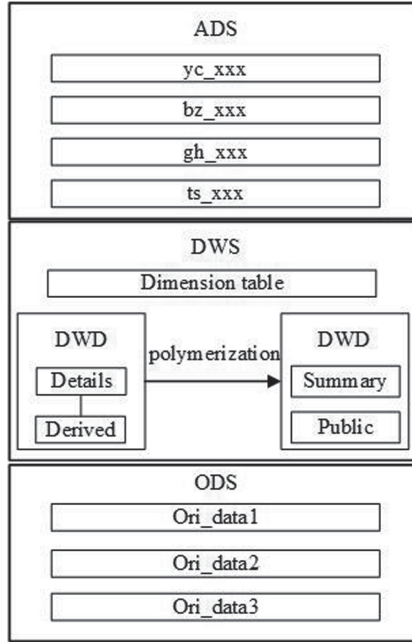


Fig. 5. Hierarchical structure of data-center-platform

#### Operation Data Store (ODS).

The data of all business systems on the integration platform is unified at this layer through data synchronization tools, which shields the heterogeneity of various data sources and provides a unified view for users. Users can save historical data and clean data according to data business requirements and audit requirements.

#### Common Data Model (CDM).

The common data model can be divided into data warehouse detail (DWD) and data warehouse summary (DWS). The common data model uses the data modeling method based on dimension modeling to build the data model [8]. For the business module, it divides and defines the data fields, business processes, dimensions, atomic indicators, modification types, time periods, modifiers and derived indicators. It uses the detailed wide table, reuses the associated calculation, and reduces the data scanning [9].

On the basis of data cleaning, filtering, recording history and other operations on the data from the operation data store, the data warehouse detail completes data fusion for multi-source and same subject data, forming a detailed fact table with the most original

granularity. The data warehouse summary performs statistical summary or algorithm to form summary fact table for detailed data according to different granularity and dimensions.

### **Application Data Store (ADS).**

The data of application data store is processed and generated according to the common data model and operation data store, and the personalized statistical indicator data is stored. It is suitable for the indicators of nonutility and complexity (index type, ratio type, ranking type). The application data store data mainly corresponds to the data, which is the comprehensive business analysis information.

## **3.2 Data Management**

In the data-center-platform, the most original and massive business data are calculated and processed to obtain the high-value data results required by the business, which are open to users of the system integration platform in the form of services, so as to realize that the data comes from the business and is ultimately used by the business.

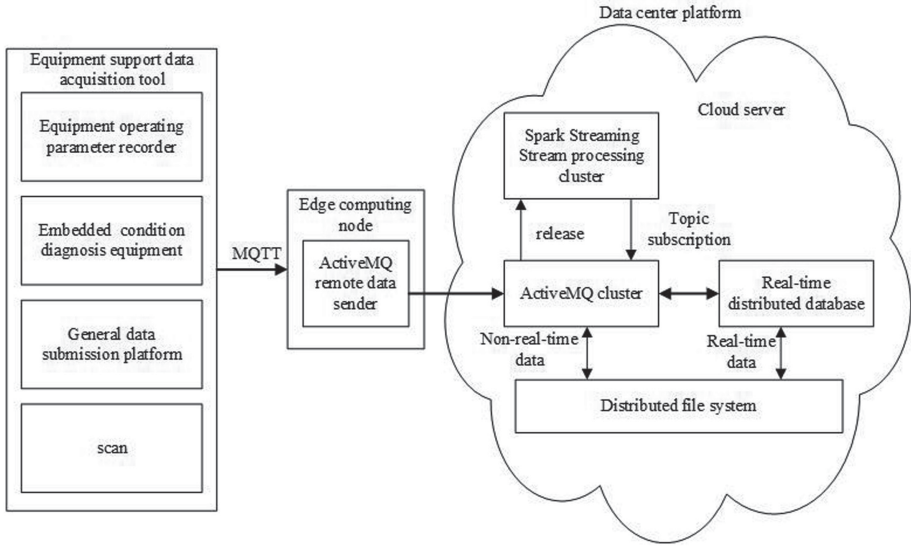
Data-center-platform provides unified, standardized and shared data services, and provides external services in the form of fixed application programming interface (API) for business calculation results. The data service adopts the Serverless architecture, users only need to pay attention to the query logic of API itself, do not need to care about the infrastructure such as running environment, and the API supports elastic expansion. After the data service generates the API, it registers in RESTful style and publishes it to API Gateway [10] for calling.

## **3.3 Data Synchronization Process**

The data acquisition architecture is shown in Fig. 6, through the data collection tools such as equipment operating parameter recorder, embedded condition diagnosis equipment and general data collection and reporting platform, using message queuing telemetry transport (MQTT) protocol to send the collected equipment basic data, business data, and real-time status data such as usage and consumption to the corresponding theme or queue cache of ActiveMQ cluster [11], storing the status data that needs to be processed in real time to real-time distributed databases such as HBase, and storing the non-real time data to the Hadoop distributed file system (HDFS).

Finally, all data will be stored in the distributed file system for extraction. Users can use Spark Streaming and other processing frameworks [12] to process big data by subscribing to the data of the corresponding topic, so as to meet the service application requirements. In addition, the high availability design of cluster server [13] can prevent data transmission from being affected by a server outage.

In the data integration architecture based on data-center-platform, select the corresponding data extraction plugins according to the data type, and synchronize all business data to data-center-platform. According to the needs of business, each application subsystem adopts different data storage structures, including structured data, semi-structured data and unstructured data.



**Fig. 6.** Equipment support data collection architecture

For structured data, synchronizing the data to the data-center-platform. For semi-structured data and unstructured data, storing the data to the data-center-platform after structured processing. A tool named Maxcompute 2.0 from Alibaba provides Java extension mode to support the analysis and output of any semi-structured data and unstructured data [14]. Maxcompute uses StorageHandler as the unified logical entry of external tables, which can read the semi-structured data and unstructured data stored on the object storage service (OSS), and then in the Extractor, it can parse the data stream called InputStream into structured record data, and then decides how to implement output data through Outputer [15].

Take the processing procedure of the garage monitoring video stored in the OSS as an example, using video analysis and intelligent reading technology and methods in Extractor to identify the number of people entering and the posture of the personnel, and store the result in the data-center-platform. Data-center-platform converts the unstructured video data into structured and statistical data, which provides data support for the inspection and evaluation of the daily situation of the garage.

In the daily evaluation of the garage, firstly, the MaxCompute tool is used to convert unstructured surveillance video data into structured statistical information on the number and posture of people entering. Then, using multi-dimensional data analysis and mining tools, combined with the equipment activation data detected in the operating parameter recorder, a comprehensive analysis of information such as personnel entry and equipment activation is conducted to obtain the evaluation result of the garage, and released in the form of RESTful data services for calling, the specific process is shown in Fig. 7.



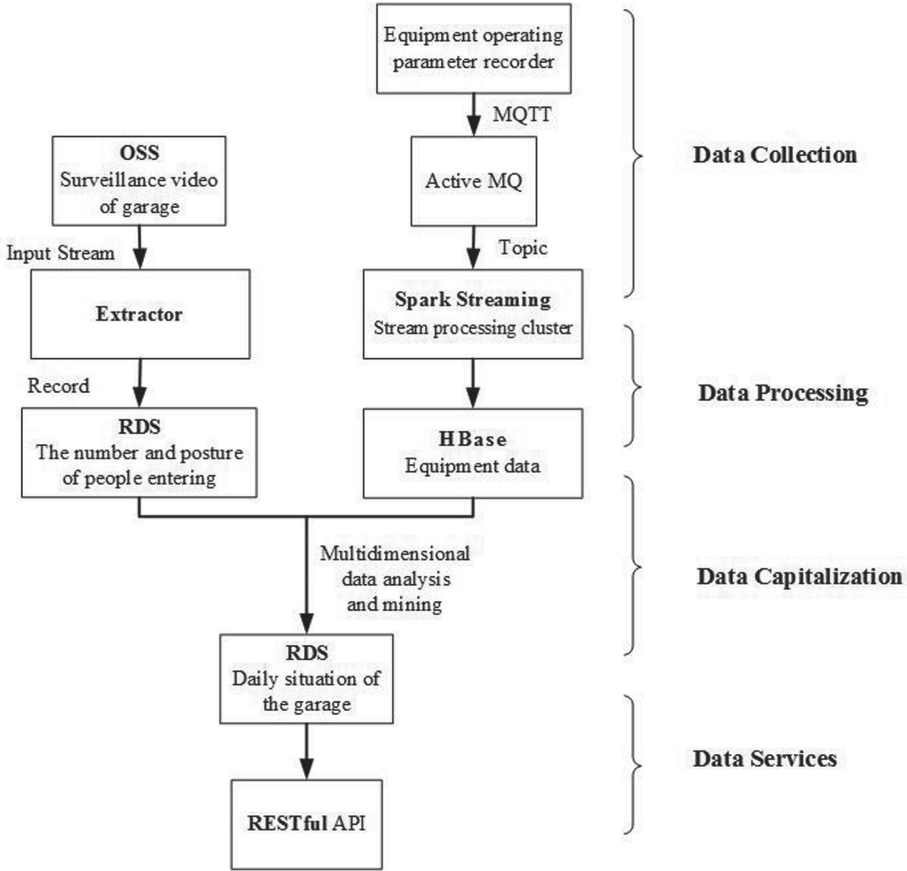


Fig. 7. Data processing of typical equipment support business

### 4 Conclusion

The data-center-platform has changed the status of self-collection and self-use of the current business system. Relying on the existing business system, the relevant data resources have been effectively integrated, and a data platform for overall management and centralized storage has been established to ensure safe storage and flexible call, and expand the innovative application of big data. Rely on data to scientifically analyze the problems in the daily business of various departments, can objectively propose targeted solutions, and finally achieves effective support for the construction of big data.

**Acknowledgments.** This research is sponsored by State Grid Technology Project “Research and Application of Key Technologies in Enterprise-level Data Center Platform based on Full-Service Unified Data Center” under grants number 5211XT190033.

## References

1. Tan, H.: Explain on ODPS of Ali. *China Information Weekly* (2019)
2. Zhang, J.: Design and realization of data center platform. *Inf. Secur. Technol.* **12**, 46–49 (2011)
3. Bell, D.L., Cozzolino, D.L., Lee, M.L., Hagen, R.L.: System and method for database synchronization (1998)
4. He, T., Tang, X., Zhang, B., Cui, S. and Xie, G.: Study the way of the security database synchronization between physical isolated networks of Web-based metallurgy database. *Comput. Appl. Chem.* (2014)
5. Gai, J.: Data replication technique analysis and application in distributed database system. *Comput. Appl. Softw.* **22**(7), 36–38 (2005)
6. Wang, Y., Rao, X., He, P.: Incremental database synchronization update mechanism under heterogeneous environment. *Comput. Eng. Des.* **32**(3), 948–951 (2011)
7. Zhang, Y., Xia, K., Zhang, F.: Research and implementation of data synchronization for distributed heterogeneous database. *J. Chin. Comput. Syst.* (2007)
8. Wang, T., Chen, M., Zhao, H., Zhu, L.: Estimating a sparse reduction for general regression in high dimensions. *Stat. Comput.* **28**(1), 33–46 (2016). <https://doi.org/10.1007/s11222-016-9714-6>
9. Koziel, S., Bekasiewicz, A.: Fast redesign and geometry scaling of multiband antennas using inverse surrogate modeling techniques. *Stat. Comput.* (2018)
10. Yu, Y., Silveira H., Sundaram, M.: A microservice based reference architecture model in the context of enterprise architecture. In: *Advanced Information Management, Communicates, Electronic & Automation Control Conference*, pp. 1856–1860 (2016)
11. Puripunpinyo, H., Samadzadeh, M.: Design, prototype implementation, and comparison of scalable web-push architectures on Amazon web services using the Actor model. In: *International Conference on Systems Engineering*, pp. 301–308 (2017)
12. Chintapalli, S., et al.: Benchmarking streaming computation engines: storm, flink and spark streaming. In: *IEEE International Parallel & Distributed Processing Symposium Workshops*, pp. 1789–1792 (2016)
13. Hou, Y., Xu, Q., Li, Y.: Monitoring system for mineral processing equipment based on IoT and industrial cloud computing. *Comput. Integr. Manuf. Syst.* **23**(9), 1972–1982 (2017)
14. Wang, C.: A brief talk on the research and recommendations of internet enterprise data central platform architecture and security. *Inf. Syst. Eng.* (2019)
15. Wang, J., Xu, D., Wang, Y., Gao, S., Hu, Q., Ding, X.: The design for electric marketing service data middle-platform. In: Atiquzzaman, M., Yen, N., Xu, Z. (eds.) *BDCPS 2019. AISC*, vol. 1117, pp. 1752–1760. Springer, Singapore (2020). [https://doi.org/10.1007/978-981-15-2568-1\\_247](https://doi.org/10.1007/978-981-15-2568-1_247)



# Adaptive Adversarial Attack on Graph Embedding via GAN

Jinyin Chen<sup>1,2(✉)</sup>, Dunjie Zhang<sup>2</sup>, and Xiang Lin<sup>2</sup>

<sup>1</sup> Institute of Cyberspace Security, Zhejiang University of Technology, Hangzhou 310023, China  
chenjinyin@zjut.edu.cn

<sup>2</sup> College of Information Engineering, Zhejiang University of Technology,  
Hangzhou 310023, China

**Abstract.** Graph embedding method learns the low-dimensional representation of graph data, which facilitates downstream graph analysis tasks, such as node classification, graph classification, link prediction and community detection. With the in-depth study of graph analysis tasks, the issues of excessive data mining by graph embedding methods have become increasingly prominent, a number of graph embedding attack methods have been put forward. Inspired by promising performance of generative adversarial network, this paper proposes an adaptive graph adversarial attack framework based on generative adversarial network (AGA-GAN). We use the game between a generator and two discriminators with different functions to iteratively generate the adversarial graph. Specifically, AGA-GAN generates the adversarial subgraph according to different attack strategies to rewire the corresponding parts in the original graph, and finally form the whole adversarial graph. To address the scalability problem of existing graph embedding attack methods, we consider the adaptively selected K-hop neighbor subgraph as the attack target instead of the original graph. Experimental study on real graph datasets verifies that the AGA-GAN can achieve state-of-the-art attack performance in most node classifications.

**Keywords:** Adversarial attack · Generative adversarial network · Graph embedding · Node classification

## 1 Introduction

Our lives are surrounded by various graph data, which used to represent data in a lot of fields, such as social networks, communication networks, biological networks, transportation networks and so on. Graph embedding methods [1–3] map information of nodes and links in the graph into low-dimensional Euclidean space, enabling the real-world graph analysis tasks such as node classification [4, 5], graph classification [6, 7], link prediction [8, 9], and community detection [10, 11]. The graph embedding methods usually learn the low-dimensional representation of graph structure, which directly determines the performance of downstream tasks, so it has received increasing attention recently.

With the widespread application of graph embedding methods in actual graph analysis tasks, many methods for downstream tasks have been proposed. Kipf et al. [12] proposed GCN as a basic graph convolution method for semi-supervised classification. This method is widely used in node classification and graph classification tasks. DIFF-POOL [6] uses a differentiable graph pooling module to adapt to various graph neural network architectures in a hierarchical and end-to-end manner. The DGCNN proposed by Zhang et al. [7] allows traditional neural networks to be trained on graph data. Deepwalk [13], node2vec [15], GCN [12] and other graph embedding methods also show superior performance over traditional algorithms in the task of link prediction. And new methods are constantly emerging in community detection tasks [16, 17].

The rapid emergence of graph embedding methods has also brought about the security problem of graph analysis tasks. In graph analysis task, failing the graph embedding model can also be achieved by faking the graph nodes, rewiring links or attributes modification, so as to protect the graph data from over-explorer. Zugner et al. [18] proposed the first adversarial attack on graph data, since then a number of attacks on node classification [19–21] have been studied. Some studies have focused more on other graph analysis tasks, such as community detection attack [22, 23] and link prediction attack [24, 25].

Generative adversarial network (GAN) [26] has achieved remarkable achievements in computer vision attacks [27, 28], natural language processing, audio recognition etc. GAN has also been used for graph data in recent years [29–31]. Since handling the graph data usually has scalability problem, it takes a lot of time and storage space to generate a full-size adversarial graph using GAN, which makes it difficult for GAN to achieve a fast and efficient attack on graph data. Most of the existing work focuses on how to better learn the embedding representation of graph data, and to our best knowledge, it is the first work on graph attacks via GAN.

In order to generate adversarial graph with minimal perturbation and maximal attack success rate, we propose an adaptive graph attack framework (AGA-GAN). Specifically, we design a multi-strategy attack generator (MAG), Similarity Discriminator (SD) and Attack Discriminator (AD) to form a three-player game. We consider the adaptively selected K-hop neighbor subgraph as the attack target instead of the original graph to reduce the cost of GAN in graph data attack. We attacked several graph embedding methods with node classification as downstream tasks and verified the effectiveness and universality of the proposed AGA-GAN on real graph data.

## 2 Related Work

### 2.1 Graph Attack Methods

The development of graph embedding models has given the graph analysis tasks a better theoretical basis. Massive real data are used for graph analysis, coming along security risks caused by excessive graph data analysis. In order to raise the security of graph data such as personal privacy [32], biomolecular structure [33], community structure [34] under the increasingly efficient graph data analysis methods, attack methods on graph data have been proposed to protect privacy from the excessive graph analysis.

Zugner et al. [18.] proposed the first adversarial attack against the graph data to generate the adversarial graph iteratively, namely NETTACK. This method focuses on the attack effect in the node classification, and achieves an effective attack within a limited budget. They further proposed Meta-Self [21], when the classification model and its training weights are unknown, regarding the graph as an optimizable hyper parameter, and using meta-gradients to solve the bi-level problem underlying training-time attacks. The FGA proposed by Chen et al. [19] extracts the gradients of node pairs based on the adversarial graph, and selects the node pairs with the largest absolute gradient to implement a fast gradient attack (FGA). It has strong attack transferability on various graph embedding methods. Chang et al. [20] built graph filters corresponding to the graph embedding models, and realized the attack in the black box environment through attacking the graph filters. Chen et al. [22] regarded community detection attacks as an optimization problem and proposed an attack strategy based on genetic algorithm and Q modularity. Yu et al. further proposed evolutionary perturbation attack (EPA) [23] based on genetic algorithm by rewiring the graph to achieve the attack.

## 2.2 GANs

Generative adversarial network (GAN) is a deep learning model proposed by Goodfellow et al. [26]. Since then it has become a powerful subclass of generative model [35] widely applied to image generation, text generation, semantic segmentation and other fields. A classical GAN is composed of a generator and a discriminator, using a two-player game idea, it can learn to deal with complex distribution problems through the mutual game between the generator and the discriminator.

In the field of graph data, the studies of GAN mostly focus on learning the embedding representation of graph data. For instance, GraphGAN [29] uses the generator to learn the potential connectivity distribution in the graph data, and predicts the probability of the existence of a link between a pair of nodes by the discriminator, unifying the generation of the adversarial graph into the GAN's minimax game. Bojchevski et al. [30] proposed NetGAN, applying Wasserstein GAN to the graph field, learns the distribution of biased random walks on graph and generates credible random walks in real graph. Pan [31] further combined the variational graph autoencoder with GAN and proposed a framework ARVGE for learning graph embedding and being able to reconstruct graph data. In conclusion, they all adopt GAN as an efficient embedding representation learning method.

## 3 Preliminary

In this section, we briefly formulate the graph embedding and node classification attack problem. A graph is represented as  $G = \{V, E, X\}$ , where  $V = \{v_1, \dots, v_n\}$  is the node set with  $|V| = N$ ,  $e_{i,j} = \langle v_i, v_j \rangle \in E$  denotes that there is a link between nodes  $v_i$  and  $v_j$ . The node topology of the graph is generally represented by the adjacency matrix  $A \in \{0, 1\}^{N \times N}$ ,  $A_{i,j} = 1$  if node  $v_i$  directly connected with  $v_j$ .  $X \in \{0, 1\}^{N \times D}$  is the node attributes matrix, and  $D$  denotes the dimension of  $X$ . Generally, the adjacency matrix  $A$  contains the information of  $V$  and  $E$  in the graph data, so we use  $G = (A, X)$  to represent a graph more concisely.

**Graph Embedding.** The graph embedding methods map the graph data  $G$  into an embedding matrix  $Z \in R^{N \times d}$  in a low-dimensional space, while retaining the information of the adjacency matrix  $A$  and the node attributes  $X$ . The dimension of  $d$  is much smaller than  $N$ , which allows graph data to use the embedding matrix to design downstream methods to implement graph analysis tasks such as node/graph classification, link prediction and community detection.

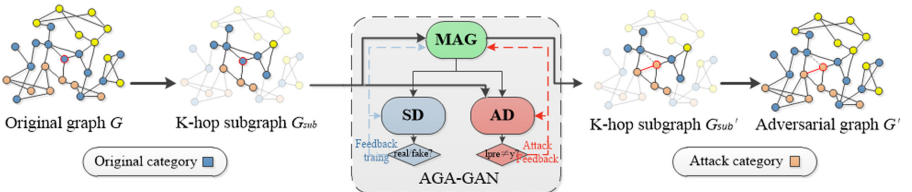
**Node Classification Attack.** Given a graph  $G$  and target node  $v_i$ .  $F = [\tau_1, \dots, \tau_{|F|}]$  is the category set of nodes,  $\tau_{i_{ori}} \in F$  denote the ground true category of the target node  $v_i$ . Our goal is to generate the adversarial graph  $G'$ , which makes the target node  $v_i$  can get a prediction category  $\tau_{i_{att}}$  with the largest distance from  $\tau_{i_{ori}}$  through the target node classifier  $f_{\theta}^{node}$ :

$$\arg \max_{\tau_{i_{att}} \neq \tau_{i_{ori}}} \text{In}Z_{v_i, \tau_{i_{att}}}^* - \text{In}Z_{v_i, \tau_{i_{ori}}}^* \quad (1)$$

where  $Z^* = f_{\theta}^{node}(G')$ ,  $\theta$  denotes the parameter of the target model training with the real graph  $G$ .

## 4 Method

Our proposed AGA-GAN attacks node classification by combining different attack strategies. Figure 1 shows the attack process of AGA-GAN, which consists of three parts: multi-strategy attack generator (MAG), similarity discriminator (SD) and attack discriminator (AD). We choose the  $K$ -hop neighbor subgraph of the target node in the original graph as the attack target. Through the alternating training of MAG, SD and AD, AGA-GAN chooses to generate adversarial subgraph structure  $A'$  or adversarial node attribute  $X'$  according to different attack strategies. Then we replace the corresponding part in the original graph with the adversarial subgraph, achieve an effective node classification attack.



**Fig. 1.** Process of AGA-GAN adaptive attacking original Graph  $G$ . We adaptively set the input size of AGA-GAN according to the  $K$ -hop( $K = 2$  in here) neighbor subgraph of the target node (normally colored nodes and links). AGA-GAN generates the adversarial subgraph and replaces the corresponding part of the original graph. (Color figure online)

#### 4.1 Multi-strategy Attack Generator

**Structure of MAG.** The MAG we propose achieves adaptive generation of adversarial graph data through different attack strategies. The MAG contains two parts, feature extractor and a graph reconstructor.

*Feature Extractor.* In order to learn graph structure  $A$  and node attributes  $X$  in our proposed AGA-GAN, we consider a two-layer graph convolution network (GCN) as the graph feature extractor. It maps the graph structure and node attribute information to a  $d$ -dimensional feature matrix. The low-dimensional features of graph data are defined as:

$$Z = f(X, A) = f(\hat{A}\sigma(\hat{A}XW^{(0)}))W^{(1)} \quad (2)$$

where  $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ ,  $A$  is the adjacency matrix and  $\tilde{A} = A + I_N$  is the adjacency matrix of the real graph  $G$  with the added self-connections.  $I_N$  is the identity matrix and  $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$  denotes the degree matrix of  $\tilde{A}$ .  $W^{(0)} \in R^{N \times H}$  and  $W^{(1)} \in R^{H \times d}$  denote the trainable weight matrix of hidden layer and output layer with  $H$  feature maps,  $N$  is the number of nodes in the graph, and  $d$  denotes the dimension of low-dimensional representation.  $f$  and  $\sigma$  are the softmax function and Relu active function.

*Graph Reconstructor.* After obtaining the low-dimensional representation  $Z$  of the graph data through the graph feature extractor, we use a dimension expansion matrix  $W_{ex}$  to reconstruct  $Z$  into the adversarial graph  $G'$ :

$$G' = \begin{cases} A' = \bar{\bar{}}[\mathcal{S}((ZW_{ex}^A + (ZW_{ex}^A)^T)/2)] \\ X' = \bar{\bar{}}[\mathcal{S}(ZW_{ex}^X)] \end{cases} \quad (3)$$

where  $Z \in R^{N \times d}$ ,  $W_{ex}^A \in R^{d \times N}$  and  $W_{ex}^X \in R^{d \times D}$  are the dimension expansion matrix of graph structure  $A$  and node attributes  $X$ . Sigmoid function  $\mathcal{S}$  maps the element values of generated data between  $[0-1]$ , then obtains discrete  $G'$  by the sign function  $\bar{\bar{}}$ .

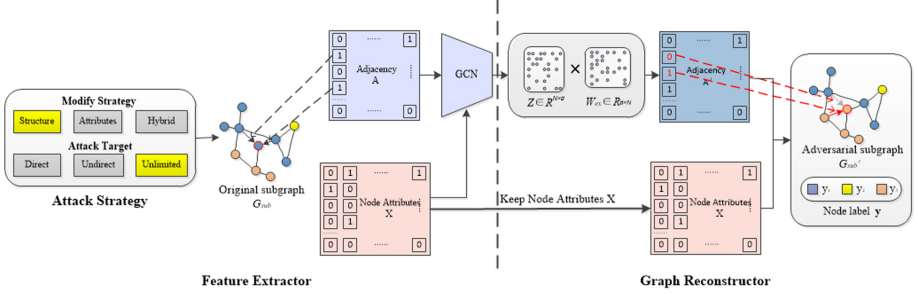
**Multiple Training Strategies.** In order to satisfy the AGA-GAN's requirements for different graph analysis attack strategies, as shown in Fig. 2, we determine how MAG generates graph structure  $A$  or node attributes  $X$  based on different attack strategies.

*Modify Strategy.* In MAG, we implement various attack strategies by modifying graph structure  $A$ , node attributes  $X$  or a combination of both.

**Graph Structure Attack:** In a general social network, the structure of the social network represents the interaction relationship between users. Modifying the links between nodes in the graph can effectively hide these relationships.

**Node Attributes Attack:** In community detection tasks, similar users usually have similar node attributes. By modifying the attributes of the target node itself or its connected nodes, the node information can also be hidden.

**Hybrid Attacks:** Modify both graph structure  $A$  and node attributes  $X$  to avoid too much perturbation in the graph structure or node attributes, and ensure that the attack is unnoticeable while performing an effective attack.



**Fig. 2.** MAG in node classification attack. The red solid line indicates the added links, and the grey dotted line indicates the deleted links. Here we choose an unlimited attack on the graph structure. We obtain the low-dimensional features of the graph through the feature extractor, and then obtain our adversarial graph structure through the graph reconstructor. (Color figure online)

**Attack Scale.** In order to efficiently implement the adversarial attack on the graph and reduce the attack cost, we use  $K$ -hop graph instead of the original graph in the attack process to achieve an efficient attack. Specifically, we select the target node and its  $K$ -hop neighbors from the original network  $G$  to form a  $K$ -hop subgraph  $G_{K-sub}(A_{sub}, X_{sub})$ . To prevent the node categories in the subgraph from being too concentrated when  $K$  is too small, which leads to poor attack effects, we randomly add nodes with other categories that are 20% of the number of subgraph nodes to the  $G_{K-sub}$ , then adaptively generate the adversarial subgraph  $G'_{K-sub}(A'_{sub}, X'_{sub})$  corresponding to the size of  $G_{K-sub}$  through MAG, when the attack on subgraph is successful, replace the subgraph  $G_{K-sub}$  with  $G'_{K-sub}$  in the original graph  $G$  to obtain the adversarial graph  $G' = (A', X')$ . We consider the following three different attack scale  $k \in N_+$ :

**Direct Attack ( $k = 1$ ):** Only delete the existing links of the target node or add a new one, or modify the target node's attributes.

**Indirect Attack ( $k \leq K, k \neq 1$ ):** Delete or add links in the 2-to- $K$  hop node pairs except the target node in subgraph  $G_{K-sub}$ , or modify these nodes' attributes.

**Unlimited Attack ( $k \leq K$ ):** Combining the above two attack scales, delete or add links between any pair of nodes in  $G_{K-sub}$ , or modify the attributes of any node.

## 4.2 Similarity Discriminator

SD aims to learn the difference between  $G_{K-sub}$  and  $G'_{K-sub}$ , and distinguish the two as much as possible. SD also provides feedback to the MAG and guides it to generate the adversarial subgraph that is more similar to the original one.



**Structure of SD.** We use a classical Multi-layer Perceptron (MLP) with a hidden layer as our SD, where the output layer is set to a one-dimensional sigmoid function. The hidden layer and the output layer in SD can be generally expressed as:

$$h^{(l+1)} = \text{sigmoid}(W_{SD}^{(l)}h^{(l)} + b^{(l)}) \quad (4)$$

where  $W_{SD}^{(l)}$  and  $b^{(l)}$  are the trainable weight matrix and bias term of  $l$  layer. We use the subgraph structure  $A_{sub}$  or the node attributes  $X_{sub}$  as the input  $h^{(0)}$  of the SD according to the modification strategy, and calculate the hidden layer's output  $h^{(1)}$  as the input of the output layer, then get a one-dimensional value  $h^{(out)} \in [0, 1]$  from the sigmoid function.

**Training Steps.** During the training process, MAG tries to generate a more realistic subgraph to fool the SD, and the SD needs to maximize the difference between  $G_{K-sub}$  and  $G'_{K-sub}$ . The optimization objective of alternating training of the SD and the MAG can be defined by:

$$\min_{MAG} \max_{SD} \mathbb{E}_{G \sim P_{real}} [\log SD(G_{K-sub})] + \mathbb{E}_{G' \sim P_{MAG}} [\log(1 - SD(G'_{K-sub}))] \quad (5)$$

where  $G_{K-sub} \sim P_{real}$  and  $G'_{K-sub} \sim P_{MAG}$  denotes original subgraph and adversarial subgraph generated by MAG.

### 4.3 Attack Discriminator

In node classification attack, we use the GCN model with the same structure as the MAG as the AD. Through the alternating training of MAG and SD, we obtain the adversarial subgraph that is similar to the real one. AD then provides feedback to MAG and guides it to generate adversarial subgraph which can fail the target model.

**Traning Steps.** For an effective attack on the target model, AD performs the following two steps in each iteration:

*Step1.* Freezing the weights of MAG and SD, train the weights of the AD using the real subgraph  $G_{K-sub}$ , and then we optimize the AD by minimizing the cross-entropy loss function to improve the accuracy of AD in classifying nodes in real subgraph:

$$\arg \min L_{AD} = - \sum_{l=1}^{|T_s|} \sum_{k=1}^{|F|} Y_{lk} \ln(Z_{lk}(A_{sub}, X_{sub})) \quad (6)$$

where  $T_s$  is the set of labeled nodes,  $F = [\tau_1, \dots, \tau_{|F|}]$  denotes the category set of nodes,  $Y_{lk} = 1$  if node  $v_l$  belongs to category  $\tau_k$  and  $Y_{lk} = 0$  otherwise,  $Z_{lk}(A, X)$  is the category prediction confidence output calculated by Eq (2) when  $d = |F|$ .

*Step2.* Freezing the weights of SD and AD, and using the AD obtained in *Step1* to fine-tune train the MAG. We get the predicted category confidence of the adversarial subgraph through the two-layer GCN trained in *Step1*, and define the attack loss function:

$$\arg \min L_{MAG} = - \sum_{l=1}^{|T_{tar}|} \sum_{k=1}^{|F|} Y_{lk} \ln(1 - Z'_{lk}(A'_{sub}, X'_{sub})) \quad (7)$$

where  $T_{tar}$  is the set of attack nodes,  $Y_{lk} = 1$  if node  $v_l$  belongs to category  $\tau_k$  and  $Y_{lk} = 0$  otherwise.

The optimization objective of alternating training of the AD and the MAG can be defined by:

$$\min_{MAG} \max_{AD} \mathbb{E}_{G_{K-sub} \sim P_{real}} [\log AD^F(G_{K-sub})] + \mathbb{E}_{G'_{K-sub} \sim P_{MAG}} [\log(1 - AD^F(G'_{K-sub}))] \quad (8)$$

where  $AD^F(\cdot)$  denotes the AD with  $F$  as the node category set.

## 5 Experiments

In order to testify the effectiveness of our AGA-GAN, we attack the graph embedding models with node classification as the downstream tasks, and compare the results with some baseline attack methods. In each attack, we set the ratio of training times of MAG, SD and AD is 1: 1: 1. For each attacked node, we generate 20 adversarial graphs. Once a confrontation graph can successfully attack the node classification, we consider that the attack was successful our experimental environment consists of i7-7700 K 3.5 GHzx8 (CPU), TITAN Xp 12 GiB (GPU), 16 GB  $\times$  4 memory (DDR4) and Ubuntu 16.04 (OS).

### 5.1 Dataset and Baseline Methods

**Dataset.** In the node classification, each node in the graph has a category. We evaluated our method on three real-world datasets: Pol. Blogs [34], Cora [35] and Citeseer [36]. The nodes denote blogs/documents, and the links are blog links/citations. Their basic statistics are shown in Table 1.

**Table 1.** The basic statistics of the three graph datasets.

Dataset	#Nodes	#Links	#Classes
Pol.Blogs	1490	19090	2
Cora	2708	5427	7
Citeseer	3312	4732	6

**Baseline Methods.** We compare our AGA-GAN with three graph embedding attack methods:

Dice [36]: DICE randomly disconnect  $b$  links of target node, then randomly connect the target node to  $M - b$  nodes of different categories.

Nettack [18]: generates adversarial disturbances for graph structure and node attributes, and according to the degree distribution and attributes co-occurrence probability to remain the perturbations are unnoticeable.

GF-Attack [20]: GF-Attack attacks graph embedding models by constructing corresponding graph filters and attacking it in a black box background.

### 5.2 Attack Performance

For each graph, we randomly select 20 nodes in each category as the target nodes. We give the attack effect of the proposed AGA-GAN method under different attack strategy settings, and compare it with several baseline methods. We use the following three metrics to measure the attack effectiveness.

**Attack Success Rate(ASR).** ASR [19] is the ratio of targets which will be successfully attacked within a given fixed budget, the ASR is defined as

$$ASR = \frac{\text{Number of successful attack nodes}}{\text{Number of attack nodes}} \tag{9}$$

**Average Modified Links (AML).** AML [19] is designed for the structure attack, which indicates the average links perturbation size leading to a successful attack.

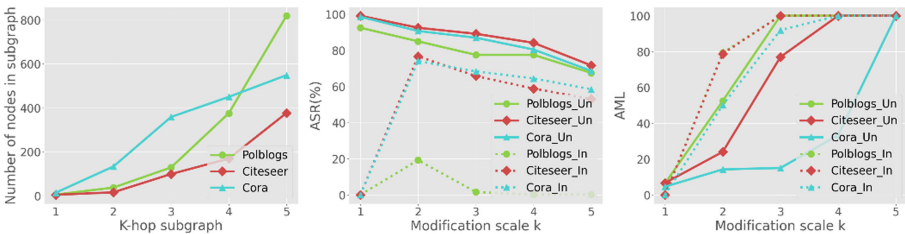
$$AML = \frac{\text{Number of modified links}}{\text{Number of attack nodes}} \tag{10}$$

**Average Modified Attributes (AMA).** AMA is designed for the attributes attack, which indicates the average attributes perturbation size leading to a successful attack.

$$AMA = \frac{\text{Number of modified attributes}}{\text{Number of attack nodes}} \tag{11}$$

**Selection of K and Attack Scale k.** In this part, considering that the useful information of the target node mostly exists in its neighborhood, we select  $K \in N_+, K \leq 5$  to get proper  $G_{K-sub}$ . We compare the average subgraph sizes under different K. We can see from Fig. 3 that when K is less than 3, the subgraph of the Cora and Citeseer dataset only contains less than 2% of the original graph, we consider the information contained in the subgraph is not enough for the subgraph attack to replace the original graph attack.

We further combine different modification scales k to observe the attack effect. We choose  $K = 3$  as the initial setting of the three attack scales. When K increases, we will also increase k to perform the corresponding three-scale attacks. Figure 3 also shows the ASR under different attack scales. The solid line represents our unrestricted attack, and the dashed line is an indirect attack. Since the initial value of K is 3, when k = 1,



**Fig. 3.** The subgraph size under different K and the ASR/AML under different modification scales k. Here we do not limit the number of modifications to get ASR, when the number of modifications is greater than 100, AML is set to 100.

**Table 2.** The ASR and AML obtained by different attack methods on various network embedding methods and multiple datasets. Here, ASR is obtained by changing 20 links.

Dataset	Model	ASR(%)				AML			
		Ours		Baseline		Ours		Baseline	
		AGA-GAN	DICE	NETTACK	GF-Attack	AGA-GAN	DICE	NETTACK	GF-Attack
Pol. Blogs	GCN	<b>92.50</b>	50.27	82.97	19.89	<b>6.47</b>	11.85	11.89	20
	Deepwalk	<b>85.50</b>	64.52	75.41	12.82	<b>7.21</b>	12.35	10.06	20
	LINE	<b>85.00</b>	66.74	76.35	23.48	<b>7.98</b>	12.82	10.26	20
	Average	<b>87.67</b>	60.51	78.25	18.73	<b>7.22</b>	12.34	10.74	20
Cora	GCN	<b>98.57</b>	54.95	92.87	82.55	6.62	9.13	<b>6.09</b>	20
	Deepwalk	<b>96.43</b>	93.52	94.06	63.47	<b>6.71</b>	7.20	7.24	20
	LINE	95.71	88.99	<b>96.34</b>	83.19	<b>6.64</b>	7.66	7.02	20
	Average	<b>96.90</b>	79.15	94.42	76.40	6.66	7.99	6.78	20
Citeseer	GCN	<b>99.17</b>	70.37	87.50	61.78	<b>4.53</b>	9.87	6.88	20
	Deepwalk	<b>98.33</b>	93.44	96.96	50.87	<b>6.18</b>	7.08	7.06	20
	LINE	<b>99.17</b>	96.72	95.82	60.41	6.42	7.21	<b>6.02</b>	20
	Average	<b>98.89</b>	86.84	93.42	57.69	<b>5.71</b>	8.05	6.65	20

we are actually conducting a direct attack, at this time we set the ASR and AML of the indirect attack to 0.

We can see that our unlimited attack has better results than indirect attack. When  $k = 1$ , i.e., when performing a direct attack, the highest ASR and the lowest AML are obtained. Interestingly, we found that as the scale of the attack increases, while AML increased, ASR decreased significantly in Fig. 3. We consider this because in  $G_{K-sub}$ , the effect of indirect links on the target node is different from that in  $G$ . Modifying the indirect links in the  $G_{K-sub}$  can misclassify the target node, but the impact of these modifications may not be that effective in  $G$ .

**Compared with the Baseline Attack Methods.** According to the experimental results above, we set  $K = 3$  in AGA-GAN and use direct attack, i.e.,  $k = 1$ . We compare with several other baseline attack methods using the most effective direct attack AGA-GAN, and the attack results are shown in Table 2, we can see that AGA-GAN outperforms all the other attack methods in all the cases, in terms of higher ASR and lower AML. However, when attacking other graph embedding methods, the ASR of AGA-GAN may slightly lower than that of attacking GCN, which is different from other baseline attack methods. This may be because the subgraph is missing part of the original graph information, which is further expanded when attacking other graph embedding methods.

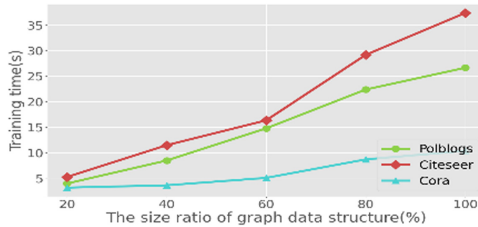
**Different Modification Strategies.** We also have node attributes attack and hybrid attack and compare the AGA-GAN-ori without adding random nodes to the k-hop subgraph. Here we also set  $K = 3$  and use direct attack. When attacking node attributes, we limit the modification of up to 100 attributes to obtain ASR. In each epoch of the hybrid attack, the MAG generates the adversarial subgraph structure  $A'_{sub}$ , and then generates the adversarial node attributes  $X'_{sub}$  based on  $A'_{sub}$ . From Table 3, we can find that the attack on node attributes can only obtain less than 50% ASR. This may be that the node attributes of the citation datasets are relatively sparse, and the node categories are more

determined by the graph structure. When we use a hybrid attack, we can get the highest ASR while reducing the AML by modifying the node attributes. Similar results can be observed in AGA-GAN-ori, however, the ASR of AGA-GAN-ori is much lower than that of AGA-GAN. This is because the category distribution of the nodes is relatively concentrated in the original neighbor subgraph of the target node, which makes the attack very difficult. This proves that our strategy of randomly adding nodes to the subgraph is effective.

**Table 3.** The ASR, AML and AMA obtained by AGA-GAN and AGA-GAN-Orisub attacking GCN model with different attack strategies. Here, ASR is obtained by changing 20 links or 100 attributes.

Attack method	Dataset	ASR(%)			AML		AMA	
		A	X	Hybrid	A	Hybrid	X	Hybrid
AGA-GAN	Pol. Blogs	92.50	67.50	<b>95.00</b>	3.34	<b>2.21</b>	5.63	<b>4.30</b>
	Cora	98.57	49.29	<b>99.29</b>	6.62	<b>5.74</b>	<b>12.88</b>	13.93
	Citeseer	99.17	61.67	<b>100</b>	4.53	<b>2.99</b>	<b>33.48</b>	34.97
AGA-GAN-ori	Pol. Blogs	67.50	47.50	<b>57.50</b>	18.23	<b>17.12</b>	<b>9.14</b>	10.53
	Cora	66.43	27.86	<b>35.00</b>	<b>3.58</b>	3.72	7.48	<b>7.04</b>
	Citeseer	69.17	45.83	<b>47.50</b>	5.52	<b>4.97</b>	21.97	<b>19.47</b>

**Time Efficiency of Attack.** Most existing methods iteratively generate adversarial perturbations on the original graph one by one to obtain the adversarial graph, which is essentially different from AGA-GAN directly generating a complete adversarial graph. In Fig. 4, we compare the training time spent by each iteration when the AGA-GAN attacks graph data of different sizes. As the graph data size increases, the training time also increases greatly. Since our  $K$ -hop neighbor subgraph size is smaller than 20% of the original graph, which means that our adaptive strategy of attacking  $K$ -hop neighbor subgraph can effectively reduce the attack cost.



**Fig. 4.** The time spend in each iteration when attacking different graph sizes.

## 6 Conclusion

In this paper, we propose an adaptive graph adversarial attack framework based on generative adversarial network(AGA-GAN). We designed MAG, SD and AD. According to different modification strategies and attack scales, MAG trains alternately with SD and AD respectively, and generates the adversarial graph similar to the real graph, which can successfully attack graph embedding models. In order to reduce the attack cost of GAN on graph data, we consider the adaptively selected K-hop neighbor subgraph as the attack target instead of the original graph. We compared the attack effects of AGA-GAN under different attack strategies, and attacked several graph embedding methods with node classification as downstream task. The experimental results show that AGA-GAN can achieve state-of-the-art attack performance in node classification when only using a small part of the original graph's structure and node attribute information.

## References

1. Cai, H., Zheng, V.W., Chang, K.: A comprehensive survey of graph embedding: problems, techniques and applications. *IEEE Trans. Knowl. Data Eng.* **30**(9), 1616–1637 (2018)
2. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **29**(12), 2724–2743 (2017)
3. Choi, E., Bahadori, M.T., Song, L., Stewart, W.F., Sun, J.: Gram: graph-based attention model for healthcare representation learning. In: *The ACM SIGKDD International Conference*, pp. 787–795 (2017)
4. Tang, J., Qu, M., Mei, Q.: Pte: predictive text embedding through large-scale heterogeneous text networks, pp. 1165–1174 (2015)
5. Wang, S., Tang, J., Aggarwal, C., Liu, H.: Linked document embedding for classification. In: *Proceedings of the 25th ACM international on conference on information and knowledge management* pp. 115–124 (2016)
6. Ying R., et al. Hierarchical graph representation learning with differentiable pooling. In: *Advances in Neural Information Processing Systems*, pp. 4800–4810 (2018)
7. Gibert, D., Mateu, C., Planes, J.: An end-to-end deep learning architecture for classification of malware's binary content. In: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, Ilias (eds.) *ICANN 2018*. LNCS, vol. 11141, pp. 383–391. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01424-7\\_38](https://doi.org/10.1007/978-3-030-01424-7_38)
8. Wang, S., Tang, J., Aggarwal, C., Chang, Y., Liu, H.: Signed network embedding in social media. In: *SDM*, pp. 327–335 (2017)
9. Tian, F., Gao, B., Cui, Q., Chen, E., Liu, T.Y.: Learning deep representations for graph clustering. In: *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1293–1299 (2014)
10. Allab, K., Labiod, L., Nadif, M.: A semi-nmf-pca unified framework for data clustering. *IEEE Trans. Knowl. Data Eng.* **29**(1), 2–16 (2017)
11. Liu, L., Cheung, W.K., Li, X., Liao, L.: Aligning users across social networks using network embedding. In *IJCAI*, pp. 1774–1780 (2016)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016)
13. Masashi, T., Kentaro, T., Jun, S.: Compound-protein Interaction Prediction with End-to-end learning of neural networks for graphs and sequences. *Bioinformatics* **35**(2), 309–318 (2018)

14. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: 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: SIGKDD, pp. 855–864 (2016)
16. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top* **69**(6), 066133 (2004)
17. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech., Theory Exp.* **2008**(10), 10008 (2008)
18. Zügner, D., Akbarnejad, A., Günnemann, S.: Adversarial attacks on neural networks for graph data. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, 19–23 August 2018, pp. 2847–2856 (2018)
19. Chen J, Wu Y, Xu X, et al. Fast Gradient Attack on Network Embedding (2018)
20. Chang H., et al. A restricted black-box adversarial framework towards attacking graph embedding models. (2019)
21. Zügner, D., Günnemann, S.: Adversarial attacks on graph neural networks via meta learning (2019)
22. Chen, J., et al.: GA based Q-attack on community detection. *IEEE Transactions on Computational Social Systems* **6**(3), 491–503 (2018)
23. Chen J, Chen Y, Chen L, et al.: Multiscale evolutionary perturbation attack on community detection 2019
24. Milani Fard, A., Wang, K.: Neighborhood randomization for link privacy in social network analysis. *World Wide Web* **18**(1), 9–32 (2013). <https://doi.org/10.1007/s11280-013-0240-6>
25. Chen J, Shi Z, Wu Y, et al. Link Prediction Adversarial Attack (2018)
26. Goodfellow, I.J., Pouget-Aabadie, J., Mirza, M., et al.: Generative adversarial networks. In: Proceedings of International Conference on Neural Information Processing Systems Kuching:, pp. 2672–2680 (2014)
27. Mangla, P., Jandial, S., Varshney, S., et al. AdvGAN ++: Harnessing latent layers for adversary generation. In: Proceedings of the IEEE International Conference on Computer Vision Workshops 2019
28. Zhu, Z.A., Lu Y.Z., Chiang C.K.: Generating adversarial examples by makeup attacks on face recognition. In: 2019 IEEE International Conference on Image Processing (ICIP). IEEE (2019)
29. Wang, H., Wang, J., Wang, J., et al. GraphGAN: Graph representation learning with generative adversarial nets. *IEEE Trans. Knowl. Data Eng.* (2017)
30. Bojchevski, A., Shchur, O., Zügner, D., et al.: NetGAN: generating graphs via random walks (2018)
31. Pan, S., Hu, R., Long, G., et al.: Adversarially regularized graph autoencoder for graph embedding (2018)
32. Razavi, M.N., Iverson, L.: Improving personal privacy in social systems with people-tagging. In: Proceedings of the 2009 International ACM SIGGROUP Conference on Supporting Group Work, GROUP 2009, Sanibel Island, Florida, USA, May 10–13, 2009. ACM (2009)
33. Garcia, J.O., Ashourvan, A., Muldoon, S.F., et al.: Applications of community detection techniques to brain graphs: algorithmic considerations and implications for neural function. *Proc. IEEE* **106**, 1–22 (2018)
34. Nagaraja, S.: The Impact of Unlinkability on Adversarial Community Detection: Effects and Countermeasures Privacy Enhancing Technologies. Springer, Berlin Heidelberg (2010)
35. Lucic, M., Kurach, K., Michalski, M., Gelly, S., Bousquet, O.: Are gans created equal? a large-scale study. In: Advances in Neural Information Processing Systems, pp. 700–709 (2018)
36. Waniek, M., Michalak, T., Rahwan, T.: et al. Hiding Individuals and Communities in a Social Network. *Nat. Hum. Behav.* (2016)



# Research on LDoS Attack Detection and Defense Mechanism in Software Defined Networks

Shengxu Xie<sup>1</sup>, Changyou Xing<sup>1</sup>(✉), Guomin Zhang<sup>1</sup>, Xianglin Wei<sup>2</sup>, and Guyu Hu<sup>1</sup>

<sup>1</sup> Command and Control Engineering College, Army Engineering University of PLA,  
Nanjing 210007, China  
xsx1727@qq.com, changyouxing@126.com, zhang\_gmwn@163.com,  
huguyu@189.cn

<sup>2</sup> The 63rd Research Institute, National University of Defense Technology,  
Nanjing 210007, China  
wei\_xianglin@163.com

**Abstract.** The LDoS (Low rate Denial of Service) attack that aims at exhausting the limited SDN switch buffer resource is hard to detect and degrade network performance seriously. To solve such a problem, this paper proposes an SDN LDoS detection and defense mechanism ADAR (Attack-flow Detection and Attack-port Recognition), which can detect the attack flows based on the collected statistical data, and identify and suppress these attack flows. The experimental results show that ADAR can effectively detect the SDN switch buffer overflow LDoS attacks, and mitigate their impact by using the attack port suppression method. Meanwhile, it can also effectively alleviate the problem of switch buffer overflow caused by the normal traffic burst in the network.

**Keywords:** Software-Defined networking · SDN switch · Low-rate denial of service attack · Switch buffer overflow

## 1 Introduction

Due to the serious harmfulness and high concealment of the Low rate Denial of Service (LDoS) attacks, it has always been a key research topic in the network security field [1]. In traditional networks, LDoS attacks against TCP are widely studied [2]. The principle is that by sending periodic high-rate pulse traffic to a router node in the link at the same time, TCP packets will be lost due to the buffer saturation in the router, and the packets that are retransmitted over time also hit the pulse traffic [3–5]. As a result, packets will be lost frequently and eventually the TCP connection is broken, degrading the performances of network communication.

In order to realize the flexibility of network management, SDN has been proposed [6]. Since SDN has the characteristics of centralized management, separation of the network forwarding and control functions, it has been widely used and studied [7]. Due to the limitation of the buffer size of the SDN hardware switch, the SDN network is also prone to packet loss caused by heavy traffic, so it is vulnerable to the same kinds of LDoS attack.



The LDoS attack flow has the characteristics of high concealment and can be hidden in the normal user flow in the network. However, the LDoS attack flow against the switch buffer space has obvious periodic characteristics after aggregated at the target switch. Based on this, without changing the network environment, this paper proposes a detection and defense scheme ADAR (Attack-flow Detection and Attack-port Recognition) that uses the data collection of the controller to detect the attack flow and identify the attack port. At the same time, ADAR can effectively alleviate the problem of switch buffer overflow caused by normal sudden large flows in the network.

The rest of the paper is organized as follows. Section 2 introduces related research works. Section 3 studies the attack model according to the characteristics of the LDoS attack. Section 4 designs and implements a detection and defense system for LDoS attacks against the switch buffer space. Section 5 evaluates the detection and defense systems through experiments. Finally, Sect. 6 concludes the paper.

## 2 Related Work

The research of LDoS attack detection and defense mechanism based on SDN environment is currently mainly focused on the types of attacks on edge servers. This type of attack generally targets resources such as server caches and queues [8, 9].

At present, there is little research on LDoS attacks against the architecture of the SDN network itself. Cao et al. proposed an LDoS attack on a control channel shared with a data channel in an SDN network, which severely affected applications such as ARP proxy, passive routing, and load balancing in SDN [10]. At the same time, the number of flow entries in SDN switches is also vulnerable to LDoS attacks [11]. None of the above studies have proposed a suitable attack detection and defense solution.

In addition, the Link Flood Attack (LFA) is a saturation attack on the core link of the SDN network, resulting in denial of service to normal network user traffic [12]. It uses the traceroute command to detect the network topology and then sends high-density traffic to keep the target link in a congested state for a long time [13], resulting in all the traffic passing through the link being affected by congestion. Unlike LFA, LDoS attacks use low-rate attack traffics, which are highly concealed and have a periodic feature. The attack aims at some specific traffics, and has little impact on the other traffics.

## 3 Definition of Attack Model

Combining the characteristics of LDoS attacks initiated by using router cache space in traditional networks, this section defines the model of LDoS attacks against SDN switch cache space based on SDN network environment. First, the attacker needs to have the following conditions to launch an attack:

- 1) The attacker has obtained the control authority of some hosts in the SDN network and can rely on these hosts to send flows to the network;
- 2) The attacker has obtained the topology information of the SDN network by means of network topology analysis, etc.;

- 3) The attacker has obtained the link selection strategy of the controller to the network flow through routing strategy analysis, etc.;
- 4) The attacker can use the synchronized host to initiate a periodic attack stream through time synchronization and other means.

Since the above four conditions have corresponding means in the network [14–16], the attacker can effectively initiate an attack flow with the following characteristics against the SDN switch cache space:

- 1) The attack traffic from a single controlled host is small, and the SDN switch can forward the attack flow of a single controlled host at line speed;
- 2) All attack flows sent by the controlled host can be converged on a port of a switch in the SDN network;
- 3) When all attack flows are converged on a certain port of a switch in the SDN network, a periodic pulse attack flow can be formed, and each pulse can cause the port on the switch to lose packets due to the buffer space overflow.

Based on the attack model, this paper will evaluate the effect of the attack by setting up a network environment in Sect. 5, and then evaluate the performance of the ADAR detection and defense mechanism proposed in Sect. 4.

## 4 Detection and Defense System Design

Based on the attack flow characteristics of the switch buffer overflow LDoS attack discussed in the previous section, in order to effectively defend against this type of LDoS attack and alleviate normal network burst flow, this section designs the corresponding ADAR detection and defense system. The ADAR system consists of four parts: Traffic Data Collection Module (TDCM), Traffic Balancing Module (TBM), Attack Period Prediction Module (APPM) and Attack Mitigation Module (AMM). The system structure is shown in Fig. 1, where each arrow indicates the corresponding message passing direction.

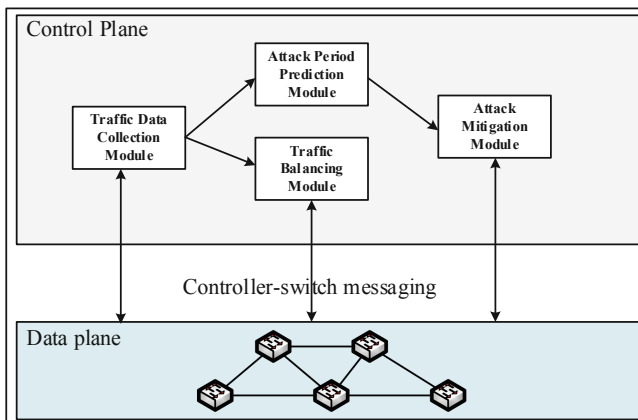


Fig. 1. Structure diagram of detection and defense system.

The general process of the system is as follows: the data collection module collects statistics field data from each switch and calculates the real-time rate of each port of the network. When it is found that the forwarding rate of a certain port of the switch is greater than a certain threshold, it sends an over-threshold warning to the attack period prediction module and the traffic balancing module. After receiving the over-threshold warning, the traffic balancing module collects the corresponding flow information on the corresponding switch and redirects some of these flows to the link with lighter load. After receiving the over-threshold warning, the attack period prediction module records the time of each warning occurrence, and analyzes and determines whether the recorded time has periodic characteristics. If it does, it sends an LDoS attack warning to the attack mitigation module. After receiving the attack suppression warning, the attack mitigation module performs attack end identification on each network access port to suppress subsequent attack flow. In this section, the specific implementation of the four modules will be introduced in detail.

#### 4.1 Flow Data Collection Module

The data collection module is mainly responsible for periodically collecting real-time load information of network links to obtain the real-time rate of each port of all switches in the SDN network. Because OpenFlow requires the switch to support the statistics of the number of sent (received) packets and bytes sent (received) by each port, this provides good support for obtaining the real-time rate of each port of the switch. Under this condition, the controller only needs to send a port statistics field request to each switch, whether it is the controller load or the secure channel load.

The specific method is that the data collection module periodically sends a port statistics field request (*port\_stats\_request*) message to the connected switch, by recording the time of each port statistics field reply (*port\_stats\_reply*) message of the switch and the amount of forwarded data of each port. The average traffic rate is calculated by the difference between the amount of data forwarded twice by each port and the difference in reply time. When the data collection period is small enough, the average flow rate can be regarded as the real-time rate, and the implementation algorithm is shown in Algorithm 1.

**ALGORITHM 1:** Information acquisition

---

```

get sWS of all switches in SDN network
when sWS
  start a thread for period request
  while time elapsed T
    for s in sWS
      send port_stats_request to s
    end for
  end while
end thread
end when
when receive a port_stat_reply msg of switch s
  record the time t
  for p in msg.port
    if first_bytes[s][p] is not none
       $rate = (msg.p.tx\_bytes - first\_bytes[s][p][2]) / (t - first\_bytes[s][p][1])$ 
      if rate > threshold
        send the warning msg of the switch s and the port p
      end if
    end if
     $first\_bytes[s][port] = [t, msg.p.tx\_bytes]$ 
  end for
end when

```

---

In Algorithm 1, lines 3–9 use a single thread to send a port statistics field value request to all switches periodically, while lines 11–22 of the code implement the *port\_stat\_reply* message event-driven method to record the forwarding bytes of each switch port count information, and calculate the average rate of the port in the period through the previous record, that is, the real-time rate.

## 4.2 Attack Period Prediction Module

Since the LDoS attack against the switch buffer space has a certain periodicity, it is feasible to determine whether the LDoS attack has been made by inferring the attack cycle and fitting the cycle verification. The specific method is to record the port rate exceeding the threshold warning time triggered by the traffic data collection module, based on the last trigger time, based on the difference between the recorded time and all the time in the recorded time as the hypothetical attack period, and continue detect the hypothetical attack period. That is, based on the last trigger time, the hypothetical attack period is sequentially subtracted. If the time value after each subtraction can find a recorded value with a small difference in the recorded time, it is reasonable to believe that the port is suffering. LDoS attack. The algorithm is shown in Algorithm 2.

**Algorithm 2:** SYSTEM PROTECTION

---

```

//record the over shreshold time in record_time
tl = record_time[1] // get the time of the last one
for t in record_time of port p of swtich s
  count = 1 // for record the hit times
  count, last_time, start_time = find_position(count, tl, t)
   $T = (\textit{last\_time} - \textit{start\_time}) / \textit{count}$ 
  record_T = [count, T]
end for
sort the record_T by count in descending order
if the record_T[1][1] is larger than 2:
  send the warning msg of the port p of the switch s and the period record_T[1][2]
end if
find_position(count, last_time, start_time)
   $T = (\textit{last\_time} - \textit{start\_time}) / \textit{count}$ 
  if  $\textit{start\_time} - T < \textit{record\_time}[\textit{len}(\textit{record\_time})]$ 
    return count, last_time, start_time, T
  end if
for t in the left times in record_time
  if  $\textit{start\_time} - t - T < 0.2$ 
    count += 1
    start_time = t
    return count, last_time, start_time
  end if
end for
return count, last_time, start_time

```

---

In Algorithm 2, lines 2–7 call the `find_position` function to count the number of hits in the record with  $T$  as the period. In the `find_position` function, line 13 calculates the average period value through statistical times to reduce statistical errors. At the same time, line 18 uses 0.2 s as the error to determine whether the time in the next record falls on the cycle. Lines 8 to 11 of the algorithm determine the cycle with the most hits. If the number is greater than 2, a warning message is sent to the attack mitigation module. After receiving the warning message, the attack mitigation module will verify at the next attack prediction point.

### 4.3 Flow Balancing Module

Because the detection of the LDoS attack flow is based on the periodicity of the attack flow, and the attack period prediction module algorithm cannot judge whether the LDoS attack has been encountered through one or two buffer overflows, there are uncertain factors in the network for a long time. Therefore, in order to detect the LDoS attack and suppress the attack, the switch affected by the buffer overflow needs to be appropriately mitigated to reduce the impact of high-speed traffic on the network performance. To this end, the traffic balancing module needs to redirect part of the traffic on the heavily loaded port to the lightly loaded port.

Because the traffic balancing module works on the controller, through the controller’s network-wide topology information, an alternate path is found when a buffer overflow occurs on the switch port, and the forwarding port of the  $n$  flows entries on the corresponding switch is modified to make the  $n$  flows redirected to the alternative path. Each overflow here redirects fewer  $n$  flows, which can effectively alleviate the switch buffer overflow under normal conditions at a high sampling rate, and the data collection module can still accurately capture the periodic pulse of LDoS attack flow. As shown in Fig. 2, when the buffer space overflow occurs on *port 1* of switch *S1*, which causes a large number of packets from switch *S1* to switch *S2* to be lost, it need to find an alternative path (*S1-S4-S2* in the Fig. 2) to redirect part of the flows. To this end, an alternative path search algorithm is designed.

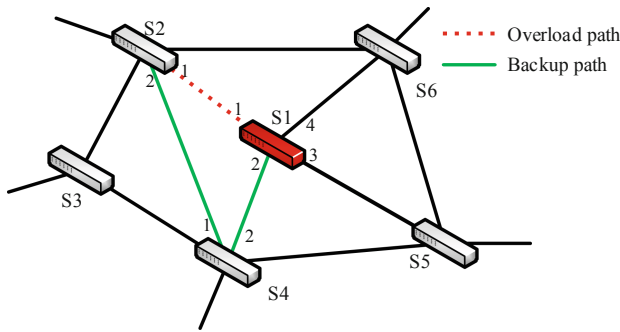


Fig. 2. Load balancing

Since the link overload problem caused by the full output queue of the switch is directional, we designed an alternative path search algorithm based on the tree search method. As shown in Fig. 3, when the *port 1* egress buffer of switch *S1* in the link  $\langle S1: 1, S2: 1 \rangle$  is full, use *S1* as the root node, add directly connected switches except the port with buffer overflow as child nodes. If the child node does not have an S2 switch, then use the child node as the parent node to search in the same way until the switch *S2*

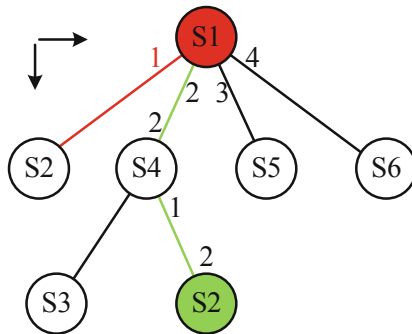


Fig. 3. Alternative path search

is found. After the switch  $S_2$  is searched, the shortest alternative link can be determined, and the link load is not overloaded.

#### 4.4 Attack Mitigation Module

The main function of the attack mitigation module is to find the attacking end and suppress the subsequent attack flow of the attacking end node accordingly. After obtaining the attack warning message sent by the attack period prediction module, extract information such as the switch  $S$ , port  $P$ , the last attack time  $t$ , and the attack period  $T$ , and use the traceability method to analyze the rate change of the corresponding flow of each network port in the period to detect the attack side. The specific implementation steps are as follows:

①Through the *aggregate\_stats\_request* message, the controller queries the  $S_1$  switch for all flow entries whose action is *out\_port* = 1, that is, the flow forwarded from the buffer overflow port;

②Extract the matching domain information of all the found flow entries, and use the matching domain as the object, and modify the *in\_port* matching domain to be the inbound port of each inbound switch;

③At the time of  $T/2$ ,  $T - \delta$ ,  $T + \delta$ ,  $T + T/2$ , send a *flow\_stats\_request* message corresponding to the modified matching domain flow entry to each network switch;

④Calculate the ratio of the difference between the number of bytes forwarded at two times of  $T - \delta$  and  $T + \delta$  on each network access port and the difference of the number of bytes forwarded at two times of  $T/2$  and  $T + T/2$ . If it exceeds a certain threshold (that is, there is a pulse stream in the next predicted attack cycle), the host connected to the port is considered to be the attacker;

⑤For the detected attack port, deliver the flow entry whose matching field is *in\_port* = *attckt\_port*, action is drop action (*apply\_actions* is empty), *priority* = 99, and *hard\_timeout* is set to 600. The matching domain setting is used to match the attacker's flow, and the action domain setting is used to directly discard the matched attack flow. Set the hard timeout time of the flow entry to 10 min, the purpose is to lift the limit after 10 min to avoid the normal flow forwarding caused by misjudgment or the attacker returning to normal user.

It can be seen from the above process that the advantage of the source tracing method is that only a small number of request response messages between the controller and the switch, which can extract the corresponding flow information from the corresponding flow table information when needed, thereby achieving the purpose of detection of the attack side.

## 5 Experimental Evaluation

In order to evaluate the LDoS attack against the switch cache space and the designed ADAR detection and defense system, the network simulation environment shown in Fig. 4 was designed based on the RYU controller and mininet. The delay of each link is set to 10 ms, where the bandwidth between links  $\langle S_1, S_3 \rangle$  is set to 1 Mbps, and the bandwidth of other links is set to 10 Mbps. Under normal circumstances, the controller

uses the Dijkstra algorithm to select the network traffic link. Therefore, it can be seen from the Fig. 4 that the links passed by the communication flow between each host and the server are  $\langle S2, S1 \rangle$ ,  $\langle S1, S3 \rangle$ .

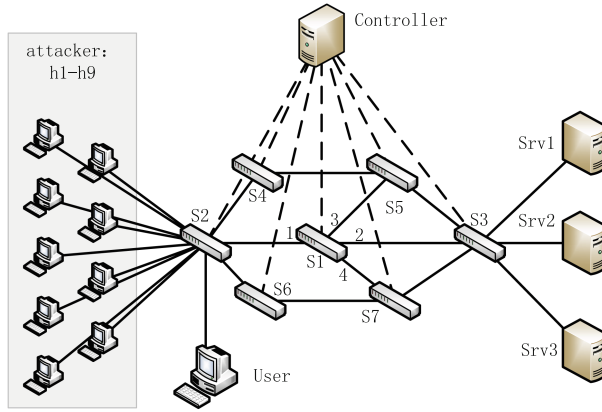


Fig. 4. Experimental topology of the LDoS attack mitigation mechanism

### 5.1 Attack Effect Evaluation

Without using the detection and defense system modules, this section firstly evaluates the effectiveness of the switch attack model. Among them, the attacker sends low-speed attack traffic pulses with a period of 10 s to the three servers through the hosts  $h1-h9$ , so that the attack traffic can reach 2 Mbps when converging. Since the bandwidth between the links  $\langle S1, S2 \rangle$  in the experimental topology is 1 Mbps, some packets will be lost at *port 2* of switch  $S1$  periodically due to the buffer overflow. Simultaneously, the normal user  $User$  connected to switch  $S2$  sends packets to the server  $Srv1$  synchronously. The result is shown in Fig. 5.

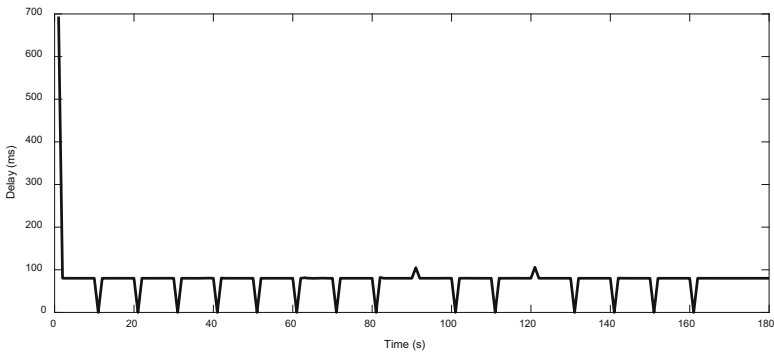


Fig. 5. Demonstration of the packet loss



As can be seen from Fig. 5, since there is no matched flow entry, the first packet in the packet sequence has a large delay. The delay of the remaining packets is either about 80 ms, or packet loss occurs (The delay in the Fig. 5 is set to be 0), and the packet loss event has a periodic characteristic with a period of 10. It can be seen from this that a carefully designed LDoS attack can periodically cause packet loss in network traffic to a certain extent.

### 5.2 Evaluating the Effectiveness of the Detection Defense System

**Effect Evaluation of Flow Balancing Module.** In order to effectively evaluate the detection and defense system, the performance of the traffic balancing module is first evaluated. The system redirects 1 flow each time the switch port buffer overflow event happens. In the case that the controller only runs the flow data collection module and the flow balancing module, the attack is launched in the experimental topology in the same attack mode, and the network speed measurement is performed on each output port of the *S1* switch. The result is shown in Fig. 6. It can be seen from the figure that the network speed of port *S1:4* will increase to a certain extent after each buffer overflow of port *S1:2*. This is because the flow through port *S1: 2* will cause the controller to pass the chain after an overflow event. The alternative path search algorithm redirects one of the flows to the switch *S1-S7-S2* link. In Fig. 6, at some attack moments, the rate of port *S1: 4* has decreased to a certain extent. It is found that this is due to the expiration of the flow entry of some flows. Since a large flow in the network generally exists for a while, and the pulse period of the LDoS attack flow is small, and the sampling period of the data collection module is 0.1 s, which can ensure that the normal network large flow can be detected, and can ensure the accuracy of the data collection information for periodic attack flows.

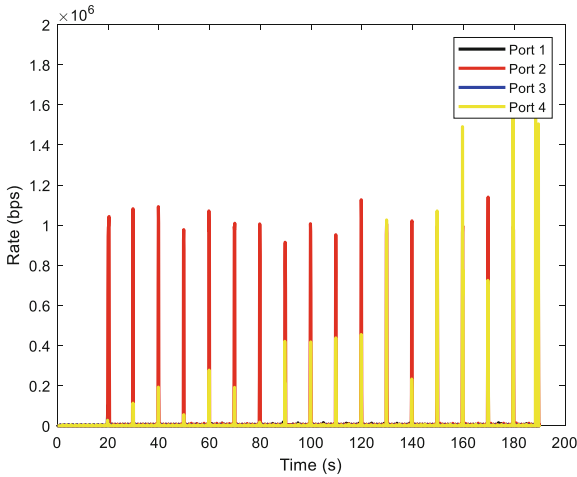
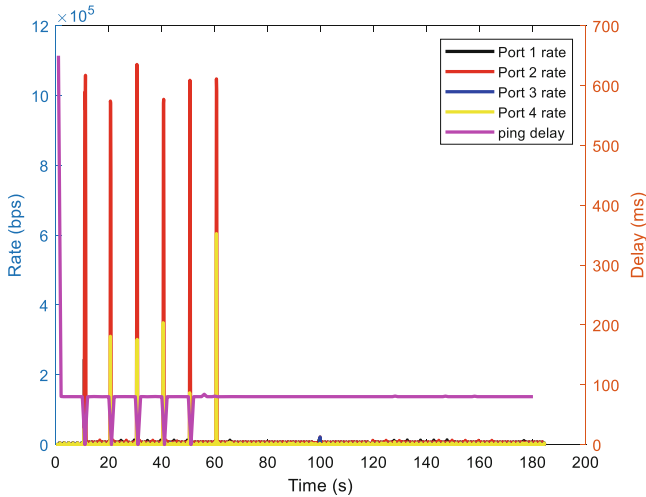


Fig. 6. Variation of the transmission rate of each port of *S1* switch under attack

**Evaluate the Effectiveness of the Detection Defense System.** Under the same attack environment, the controller runs all four modules of the ADAR to evaluate the detection and defense system, and the ratio threshold of the attack mitigation module is set to 50%. The experimental results are shown in Fig. 7. As can be seen from the Fig. 7 after *port 2* of switch *S1* has suffered six LDoS attack pulses, there will be no buffer overflow, and no packet loss will occur in ping packets.



**Fig. 7.** Packet loss analysis of detection and defense module

According to the installation of flow entries in switch *S2*, after the detection and defense module detects the attack flow, the flow entry with the action of drop is delivered to the nine attack end ports of switch *S2*. Therefore, it can be concluded that the detection and defense system designed in this paper can effectively detect the LDoS attack flow against the switch buffer overflow, and find the corresponding attack port to suppress the subsequent attack flow of the attack port by issuing drop flow entries.

## 6 Conclusion

In this paper, taking the LDoS attack that aims at exhausting the limited SDN switch buffer resources into consideration, we proposed an SDN switch buffer LDoS attack mitigation mechanism ADAR, which includes four modules named traffic data collection module, attack period prediction module, traffic balancing module and attack mitigation module. By using the centralized control plane, ADAR can predict the potential LDoS attacks based on the information collected from the data plane, and then mitigate the attacks through the SDN switch port load balancing and transmission rate limitation method. The experiment results show that the detection and defense mechanism can effectively detect the attack flow and its source position, and then mitigate its influence

through the traffic suppressing method. Meanwhile, the traffic balancing module can select the alternative path when the switch port buffer overflows, and redirect some traffic to the lighter alternative link, which alleviates the impact of the traffic bursts on the detection accuracy. In the future work, we will optimize the detection process and evaluate its performance in a larger production network.

**Acknowledgments.** This research was supported by a research grant from the National Basic Research Program of China (973 Program) under Grant No. 2012CB315806, the China Post-doctoral Science Foundation under Grant No. 2017M610286, and the National Natural Science Foundation of China under Grant No. 61103225, 61379149, 61772271.

## References

1. Richmond, R.: Firms join forces against hackers. *Wall Street J.* 28March (2005)
2. Wu, Z., Pan, Q., Yue, M., et al.: Sequence alignment detection of TCP-targeted synchronous low-rate DoS attacks. *Comput. Netw.* **152**, 64–77 (2019)
3. Luo, J., Yang, X., Wang, J., et al.: On a mathematical model for low-rate shrew DDoS[J]. *IEEE Trans. Inf. Forensics Secur.* **9**(7), 1069–1083 (2014)
4. Rabie, R., Drissi, M.: Applying sigmoid filter for detecting the low-rate denial of service attacks. In: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), pp. 450–456. IEEE (2018)
5. Cotae, P., Rabie, R.: On a game theoretic approach to detect the low-rate denial of service attacks. In: 2018 International Conference on Communications (COMM), pp. 19–26. IEEE (2018)
6. McKeown, N.: INFOCOM keynote talk. *Softw.-Defined Netw.* **17**(2), 30–32 (2009)
7. Jain, S., Kumar, A., Mandal, S., et al.: B4: Experience with a globally-deployed software defined WAN. *ACM SIGCOMM Comput. Commun. Rev. ACM* **43**(4), 3–14 (2013)
8. Hong, K., Kim, Y., Choi, H., et al.: SDN-assisted slow HTTP DDoS attack defense method. *IEEE Commun. Lett.* **22**(4), 688–691 (2017)
9. Lukaseder, T., Maile, L., Erb, B., Kargl, F.: SDN-assisted network-based mitigation of slow DDoS attacks. In: Beyah, R., Chang, B., Li, Y., Zhu, S. (eds.) *SecureComm 2018*. LNICST, vol. 255, pp. 102–121. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01704-0\\_6](https://doi.org/10.1007/978-3-030-01704-0_6)
10. Cao, J., Li, Q., Xie, R., et al.: The crosspath attack: disrupting the {SDN} control channel via shared links. In: *Usenix security symposium*, pp. 19–36 (2019)
11. Cao, J., Xu, M., Li, Q., Sun, N., Yang, Y., Zheng, J.: Disrupting SDN via the data plane: a low-rate flow table overflow attack. In: Lin, X., Ghorbani, A., Ren, K., Zhu, S., Zhang, A. (eds.) *SecureComm 2017*. LNICST, vol. 238, pp. 356–376. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-78813-5\\_18](https://doi.org/10.1007/978-3-319-78813-5_18)
12. Kang, M.S., Gligor, V.D., Sekar, V.: SPIFFY: inducing cost-detectability tradeoffs for persistent link-flooding attacks. In: *NDSS* (2016)
13. Kang, M.S., Lee, S.B., Gligor, V.D.: The crossfire attack. In: 2013 IEEE Symposium on Security and Privacy. IEEE, pp. 127–141 (2013)
14. Marin, E., Buccioli, N., Conti, M., et al.: An in-depth look into SDN topology discovery mechanisms: novel attacks and practical countermeasures. In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1101–1114 (2019)
15. Magoni, D.: Nem: a software for network topology analysis and modeling. In: *Proceedings. 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, pp. 364–371 (2002)
16. Nguyen, T.H., Yoo, M.: Analysis of link discovery service attacks in SDN controller. In: 2017 International Conference on Information Networking (ICOIN). IEEE, pp. 259–261 (2017)



# Unsupervised Analysis of Encrypted Video Traffic Based on Levenshtein Distance

Luming Yang<sup>1</sup>, Yingming Zeng<sup>2</sup>, Shaojing Fu<sup>1,3</sup>(✉), and Yuchuan Luo<sup>1</sup>

<sup>1</sup> College of Computer, National University of Defense Technology, Changsha, China  
fushaojing@nudt.edu.cn

<sup>2</sup> Beijing Institute of Computer Technology and Applications, Beijing, China

<sup>3</sup> State Key Laboratory of Cryptology, Beijing, China

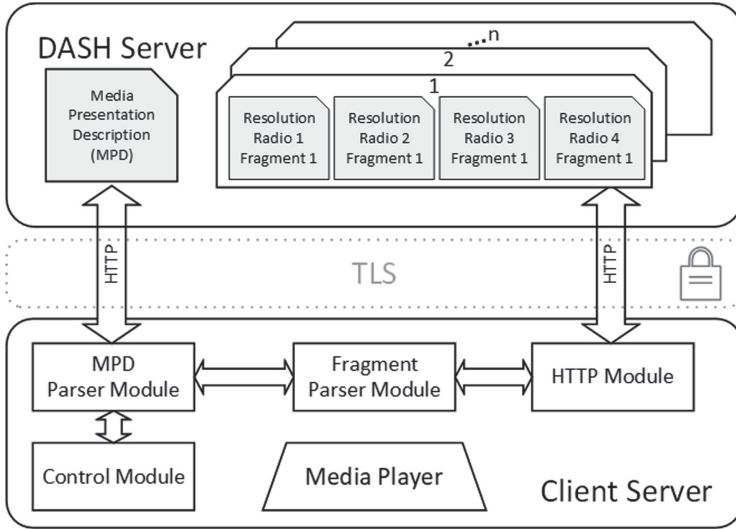
**Abstract.** It is effective for supervisors to monitor the network by analyzing traffic from devices. In this way, illegal video can be detected when it is played on the network. Most Internet traffic is encrypted, which brings difficulties to traffic analysis. However, many researches suggest that even if the video traffic is encrypted, the information of video segmentation leaked by DASH (Dynamic Adaptive Streaming over HTTP) can also be used to identify the content of encrypted video traffic without decryption. Moreover, each encrypted video stream can be represented by a fragment sequence. This paper presents two methods based on Levenshtein distance for encrypted video traffic analysis. Using the distance distribution fitted by gamma distribution functions, we calculated a threshold to determine whether two encrypted video traffic belonging to the same video. The accuracy of the judgment using the threshold reached 89%, stably. As far as I am concerned, it is the first work to apply unsupervised methods for content analysis of encrypted video traffic.

**Keywords:** Encrypted traffic · Levenshtein distance · Threshold · DASH.

## 1 Introduction

As the continuous development of network technology, there are millions of Internet video viewers online every day. More than half of Internet traffic will be video traffic nowadays. According to the survey, the proportion of video traffic has grown to 80% in 2019. In general, the video traffic is expected to increase 135 exabytes per month, approximately [7]. It is expected that more than 82% of Internet traffic will come from videos by 2022.

Dynamic Adaptive Streaming over HTTP (DASH) is used by most of the video streaming web sites, such as YouTube. DASH is a streaming method, designed to improve the quality of experience [10]. It uses HTTP for video transmission. DASH server divides each video into several short segments (typically a few seconds long), and encodes each segment with a different quality



**Fig. 1.** Dynamic Adaptive Streaming over HTTP

representation level. A media presentation description (MPD) describes segment information, and will be transmitted firstly when transmission of a video is started. According to the network condition and client preferences, adaptive video segments will be transmitted by the DASH server later.

Transport Layer Security (TLS) is widely applied to protect content confidentiality, and adopted by almost all famous video sites. Consequently, traditional Deep Packet Inspection (DPI) [6] methods over plain network traffic do not work here. However, it does not mean that it is not possible to analyze the content of encrypted video traffic. DASH video is always streamed in segment-sized chunks, and it is typically segmented at the application layer [16]. Even though the stream is encrypted between the transport layer and the application layer (e.g., using TLS), the sizes of segments are visible for network monitor. In a steady encrypted video stream, fragment sizes are correlated with the original segment sizes due to the variable-rate encoding.

Due to the difficulty of decryption, how to analyze encrypted traffic without decryption is a worthy studying direction. Deep learning has been used to analyze encrypted traffic in many works [1, 18]. Background traffic and unencrypted part of the encrypted traffic mentioned in [3–5], which are useful for normal encrypted traffic, are insignificant for content identification of encrypted video traffic. Video traffic is usually long-session with a large amount of information transmitted. What is more, most existing encryption traffic analysis methods are based on supervised learning but are not functionally faced with unlabeled traffic data. In real life, the traffic data is basically unlabeled, which makes these existing methods impractical. How to analyze encrypted video traffic without prior knowledge is a problem that should be solved quickly.

In this paper, we proposed a new method based on sequence similarity for encrypted video traffic analysis. A similarity threshold was selected to determine whether two unknown encrypted video streams belong to the same video title. As far as we knew, it is the first work to apply unsupervised methods for content analysis of encrypted video traffic.

The paper’s main contributions are:

1. This is the first work that used unsupervised methods to analyze encrypted video traffic. We proposed to measure the similarity of encrypted video streams using Levenshtein distance, innovatively. On this basis, we present an unsupervised methods (threshold) that are applicable to analyze the content of encrypted video traffic.
2. A threshold was computed using a Gamma distribution fitting to determine whether two unknown video streams belong to the same video title, and have achieved an acceptable probability of correct judgment.
3. We run through a set of experiments to prove the possibility and robustness of the threshold we computed.

The remainder of this paper is organized as follows. In Sect. 2 we review related work. In Sect. 3, we introduction the preliminaries - TLS protocol and Levenshtein distance. In Sect. 4 we introduced the generation of fragment sequence and two analysis methods. In Sect. 5 we introduced the dataset used in this paper. In Sect. 6 we computed a threshold to determine whether two unknown video streams belong to the same video title. Finally, we conclude in Sect. 7.

## 2 Related Work

Many works have suggested methods for encrypted traffic identification. Several works have examined different features.

Liu et al. [12] presented a method for video title classification of RTP/UDP traffic. Liu et al. [13] used the wavelet transform for constructing unique and robust video signatures with different compactnesses. Ashwin Rao et al. [15] showed that the streaming strategies vary with the type of the application, and the type of container used for video streaming by studying the network characteristics of Netflix and YouTube. Pablo Ameigeiras et al. [2] presented a characterization of the traffic generated by YouTube when accessed from a regular PC, and proposed a YouTube server traffic generation model. However, there are several changes in video traffic over the Internet. They do not fit modern streaming traffic as previous solutions operated on a time series with the granularity of single video frames [10].

In recent years, some work used machine learning algorithms for the identification of encrypted video traffic. Algorithms using custom KNN and SVM were presented by Ran Dubin [10] for encrypted HTTP adaptive video streaming title classification. Roei Schuster [16] showed that many video streams are uniquely characterized by their fragment patterns, and classifiers based on convolutional neural networks can accurately identify these patterns given very coarse network

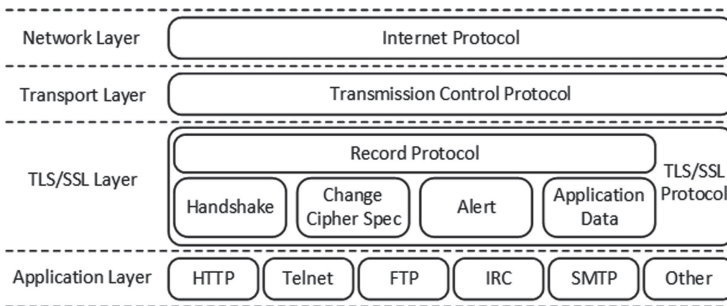
measurements. Yan Shi et al. [17] proposed a key idea to examine encrypted and tunneled video streaming traffic at a Soft-Margin Firewall (SMFW), which was located near the streaming client in order to identify undesirable traffic sources and to block or throttle traffic from such sources. These works showed that the content classification of encrypted video stream is possible although the content is not visible.

Even if some encouraging progress has been made, the timing characteristics of encrypted video stream have been ignored in these works, which contains valuable information. Moreover, the methods based on supervised learning are powerless when faced with unlabeled encrypted video traffic on the Internet. In view of this, we tried to use unsupervised learning to analyze encrypted video traffic, which requires no labels. As far as I am concerned, it is the first work using unsupervised methods to analyze the content of encrypted video traffic.

### 3 Preliminaries

#### 3.1 TLS Protocol

Transport Layer Security (TLS) is cryptographic protocol that provides secure communication between two parties over the Internet by encapsulating and encrypting application layer data. It is used by most of the video sites in order to encrypt the network traffic. The TLS protocol is between application layer and transport layer, and it is application protocol independent [8].



**Fig. 2.** Protocol layers

The TLS includes two protocol layers (as Fig. 2). The Record Protocol takes messages to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, and transmits the result [8]. Received data is decrypted, verified, decompressed, reassembled, and then delivered to higher-level clients. There are four protocols that use the record protocol, including the application data protocol. Application data message is carried by the record layer and are fragmented, compressed, and encrypted based on the current connection state.

### 3.2 Levenshtein Distance

The Levenshtein distance is a string metric for measuring the difference between two sequences. It is named after the Soviet mathematician Vladimir Levenshtein, who considered this distance in 1965 [11]. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutiond) required to change one word into other. It may also be rederred to as edit distance [14]. The Levenshtein distance between two string  $a$ ,  $b$  (of length  $|a|$  and  $|b|$  respectively) is given by  $lev_{a,b}(|a|, |b|)$  as follow:

$$lev_{a,b}(i, j) = \begin{cases} \max(i, j) & , if \min(i, j) = 0 \\ \min \begin{cases} lev_{a,b}(i-1, j) + 1 \\ lev_{a,b}(i, j-1) + 1 \\ lev_{a,b}(i-1, j-1) + 1_{a_i \neq b_j} \end{cases} & , otherwise \end{cases} \quad (1)$$

where  $1_{a_i \neq b_j}$  is the indicator function equal to 1 when  $a_i \neq b_j$  and equal to 0 otherwise, and  $lev_{a,b}(i, j)$  is the distance between the first  $i$  characters of  $a$  and the first  $j$  characters of  $b$ .  $i$  and  $j$  are 1-based indices.

For example, the Levenshtein distance between “kitten” and “sitting” is 3 because of the following edits change:

1. kitten → sitten
2. sitten → sittin
3. sittin → sitting

The upper bounds of the Levenshtein distance is the length of the longer string. The Levenshtein distance is zero only if the strings are equal.

## 4 Methodology

Video traffic has some unique characteristics. Video sessions are usually long sessions with a large amount of information transmitted, while the amount of information transmission of non-video traffic is small relatively. Regarding the information leakage in terms of timing, the timing information is leaked due to the long duration of video traffic, relatively. Therefore, we focus on timing information of encrypted video traffic in order to analyze.

Levenshtein distance can compare the similarity of sequences with different length, which is suitable for video streams. Besides, Levenshtein distance is simple and effective. Consequently, it is chosen to measure the similarity of video streams. We have developed two methods based on Levenshtein distance for identifying encrypted video traffic. Before this, packet reorganization technique was applied to generate the fragment sequence of encrypted video traffic. We used the normalized Levenshtein distance to the content similarity of unknown video streams, and calculated a threshold to determine whether two streams belong to the same video by Gamma distribution fitting.



#### 4.1 Fragment Sequence of Encrypted Video Traffic

Application data traffic accounts for most of the total encrypted traffic, especially for video traffic. Video compression and encoding algorithms cause that different video scenes contain different amounts of perceptually meaningful information. The meaningful information refers to the size of video fragments. Because of application protocol independence of TLS, this meaningful information is retained although the content of the message is encrypted.

In order to parse a TLS stream, first of all, we should reassemble the video traffic packets to TCP flow according to the TCP protocol. After that, we parse the TCP flow according to the TLS protocol. When packets reassembly and TLS parsing is completed, the TLS session exhibits a request-response pattern similar to HTTP interactive. A transaction between client and server in the TLS session, the payload sent by the server contains more than one application data. Consequently, fragment refers to the number of application data in encrypted video traffic.

After parsing the TLS stream, we can get a sequence representing, which we called Fragment Sequence, the number of the application data sent by the server per HTTP interactive. Because the encode in DASH is variable bitrate (VBR), the size of video fragments is related to the content complexity of video fragment. If the content of a video fragment is complex, the fragment size is large. Otherwise, the size of fragment is small when the content of a video fragment is simple. For example, fragment sequences of three video titles are listed as follows. We find it that videos of the same title have similar fragment sequences.

##### Cheerleader

seq1:

30-23-73-22-124-25-124-130-23-123-103-23-92-130-23-118-130-24-130-91-12

seq2:

30-23-73-22-124-25-124-130-23-116-130-23-123-103-23-92-130-23-118-130-24-130-91

seq3:

30-23-73-113-124-124-23-130-123-23-103-92-23-130-118-24-130-130-23-91

##### Fast\_and\_Furious\_six

seq1: 14-9-44-90-16-145-2-221-21-21-253-20-21-7-214-20-258-27-39

seq2: 14-9-44-21-145-2-221-21-21-253-20-9-12-7-214-20-258-27-39

seq3: 14-9-44-21-145-2-221-21-21-253-20-21-221-20-258-27-39

##### Wo\_sind\_die\_Clowns

seq1: 5-53-11-25-28-2-244-241-26-77-17

seq2: 5-53-25-27-2-244-241-26-77-17

seq3: 5-53-25-28-2-244-241-26-77-17

## 4.2 Threshold Selection

Because Levenshtein distance is affected by the length of sequences, it is necessary to normalize. The definition of normalized Levenshtein distance is as follows:

$$\text{Normalized\_LD}(a, b) = \frac{LD(a, b)}{\max\{|a|, |b|\}} \quad (2)$$

We used normalized Levenshtein distance of Fragment sequence to measure the similarity of two unknown encrypted video streams and determined whether the two streams belonging to the same video. To determine whether two unknown video streams belong to the same video, we need to set a threshold for normalized Levenshtein distance. If the distance is greater than the threshold, it is determined that two video streams belong to the different video. On the contrary, two video streams belong to the same video if the distance is less than the threshold.

The normalized Levenshtein distance of video streams is a random variable. The random variable  $X_1$  and  $X_2$  indicate the normalized Levenshtein distance of two video streams belonging to the same video and different video, respectively. After the analysis of samples, the results demonstrate that  $X_1$  and  $X_2$  conform to Gamma distribution. The parameters  $\alpha$  and  $\beta$  of the gamma distribution can be calculated from the mean  $\mu$  and variance  $\sigma^2$  of the data.

$$X_1 \sim Ga(\alpha_1, \beta_1^2), \quad X_2 \sim Ga(\alpha_2, \beta_2^2) \quad \alpha = \frac{\mu^2}{\sigma^2}, \quad \beta = \frac{\mu}{\sigma^2} \quad (3)$$

The probability density function were as follow:

$$f_1(x) = \frac{\beta_1^{\alpha_1}}{\Gamma(\alpha_1)} x^{\alpha_1-1} e^{-\beta_1 x}, \quad f_2(x) = \frac{\beta_2^{\alpha_2}}{\Gamma(\alpha_2)} x^{\alpha_2-1} e^{-\beta_2 x} \quad (4)$$

We set their distribution functions as  $F_1(x) = \int_0^x f_1(x)dx$  and  $F_2(x) = \int_0^x f_2(x)dx$ . Let the sum of the correct judgment probabilities be  $P$ , which is defined as follow:

$$P = \int_0^x f_1(x)dx + \int_x^1 f_2(x)d = F_1(x) + F_2(1) - F_2(x) \quad (5)$$

One way to get the minimum of  $P$  is to take a derivative with respect to  $x$ , and look for the derivative being zero. The derivation process is as follow:

$$\frac{dP}{dx} = F_1'(x) - F_2'(x) = f_1(x) - f_2(x) \quad (6)$$

When  $f_1(x) = f_2(x)$ , the sum of the correct judgment probabilities is the smallest. Simplify the equation  $f_1(x) = f_2(x)$  are as follows:

$$e^{(\beta_2 - \beta_1)x} = \frac{\beta_2^{\alpha_2} \Gamma(\alpha_1)}{\beta_1^{\alpha_1} \Gamma(\alpha_2)} x^{\alpha_2 - \alpha_1} \quad (7)$$

The equation is transcendental, so it does not have an analytical solution. We can use numerical analysis methods, like bisection method and Newton's method, to find the numerical solutions of this transcendental equation.

The threshold is one of the solutions.

## 5 Data

In this paper, we use the public dataset in [9]. It contains 10,000 YouTube streams of 100 video titles (100 streams per title). The video streams were collected via a real-world Internet connection over different real-world network conditions. They were collected by crawler using the Selenium web automation tool with ChromeDriver. The video titles in dataset are popular YouTube video from different categories such as sports, news, nature, etc. The traffic of the dataset were collected using Chrome browser because of its popularity.

## 6 Threshold Experimental Evaluation

In this section, we calculated the threshold for judging the homology of video traffic, and used accuracy to evaluate its performance.

### 6.1 Metrics

Before the experiment, we should define two metrics of experiments, including Theoretical Accuracy (TA) and Real Accuracy (RA).

- **Theoretical Accuracy:** The normalized Levenshtein distance of Encrypted video traffic between the same video and different video subject to be Gamma distribution. Therefore, we are able to calculate the theoretical accuracy using the gamma distribution and the threshold  $x_0$ . Theoretical accuracy is calculated as follow:

$$TA = \frac{\int_{x_0}^1 f_1(x)dx + \int_0^{x_0} f_2(x)dx}{\int_0^1 f_1(x)dx + \int_0^1 f_2(x)dx} \quad (8)$$

- **Real Accuracy:** Real accuracy is calculated using the statistical result of the experiment. The threshold  $x_0$  is used to judge positive samples and negative samples. The real accuracy is as follow:

$$RA = \frac{TP + TN}{TP + FP + FN + TN} \quad (9)$$

### 6.2 Threshold Experimental

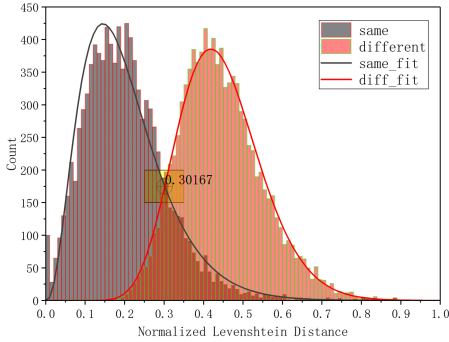
We extracted two encrypted video streams with the same title from the dataset, and calculated their similarity. We also extracted two encrypted video streams with different titles and calculated their similarity. Both operations were performed 10'000 times, and we get two sets of data about the video streams similarity (of the same title and of different titles).

As can be seen from the distributions in Fig. 3 and Table 1, there are two normal distributions with different mean and nearly the same variance ( $X_1 \sim Ga(3.53, 17.48)$  and  $X_2 \sim Ga(17.50, 39.46)$ ). It showed that the assumptions are reasonable. According to the Eq. 7, we computed the threshold to be 0.30167,

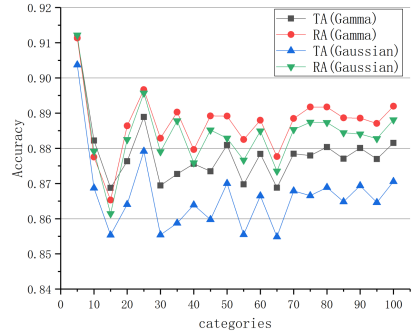
and used it to test the accuracy. The experimental result indicated that the threshold we computed can distinguish whether the two streams belonging to the same video title with 89.00% probability. The theoretical accuracy is 88.03%. The error of theoretical accuracy is less than 1%. The experimental result is acceptable.

**Table 1.** Parameter of the gamma distribution

Variables	$\mu$	$\sigma$	$\alpha$	$\beta$
same ( $X_1$ )	0.2019	0.1075	3.5289	17.4815
diff ( $X_2$ )	0.4434	0.1060	17.4970	39.4571



**Fig. 3.** Data statistics and gamma distribution fitting effect.

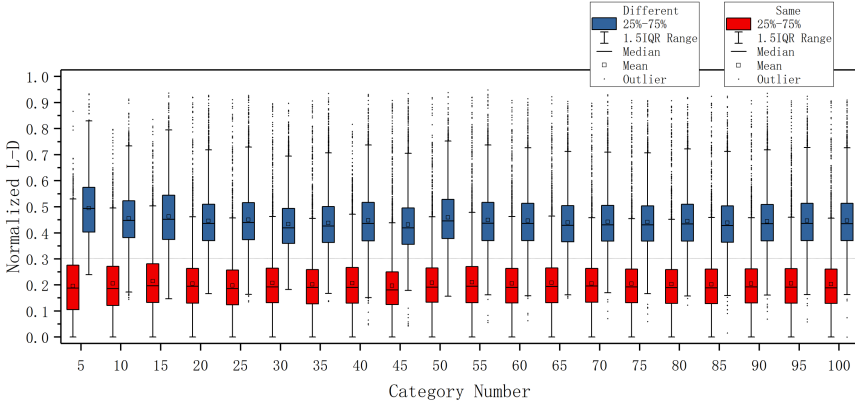


**Fig. 4.** The accuracy comparison of the fitting effect of gaussian distribution and gamma distribution.

### 6.3 Stability Test and Comparison of Different Fitting Methods

In order to test the stability of threshold and accuracy, we performed experiments on datasets with different number of categories. In each dataset, we selected 10'000 distance data for the same video and 10'000 distance data for different videos. The box-plot of datasets distribution of with different category number is depicted in Fig. 5. On the whole, it showed that the distance distribution of encrypted video stream does not vary with the category number of datasets. More importantly, Fig. 5 illustrates that the threshold we calculated has high generalization performance.

On this basis, we also compared the effect of two fitting methods, Gaussian distribution and Gamma distribution. The theoretical accuracy and real accuracy are shown in Fig. 4 and Table 2. As evident from figure, generally



**Fig. 5.** Box-plot of data distribution for different datasets.

**Table 2.** The comparison of the fitting effect of gaussian distribution and gamma distribution on different datasets.

Categories	Gamma distribution				Gaussian distribution			
	Threshold	TA(%)	RA(%)	Error(%)	Threshold	TA(%)	RA(%)	Error(%)
5	0.33068	91.158	91.135	0.023	0.34833	90.371	91.218	0.847
10	0.31046	88.221	87.752	0.470	0.33210	86.871	87.916	1.045
15	0.31326	86.876	86.532	0.344	0.33863	85.534	86.142	0.608
20	0.30310	87.633	88.639	1.006	0.32538	86.406	88.247	1.840
25	0.30353	88.891	89.668	0.777	0.32379	87.918	89.570	1.652
30	0.29889	86.943	88.288	1.345	0.32072	85.540	87.905	2.365
35	0.29682	87.269	89.030	1.762	0.31988	85.875	88.781	2.906
40	0.30441	87.561	87.965	0.404	0.32675	86.394	87.588	1.195
45	0.29119	87.348	88.922	1.574	0.31433	85.973	88.521	2.548
50	0.31144	88.094	88.914	0.820	0.33334	87.001	88.290	1.289
55	0.30567	86.975	88.249	1.274	0.32960	85.550	87.666	2.115
60	0.30517	87.838	88.800	0.962	0.32663	86.652	88.491	1.840
65	0.30097	86.882	87.764	0.882	0.32381	85.488	87.354	1.865
70	0.30325	87.846	88.847	1.002	0.32396	86.790	88.534	1.744
75	0.30246	87.794	89.173	1.378	0.32335	86.652	88.743	2.091
80	0.30269	88.033	89.173	1.140	0.32397	86.890	88.733	1.843
85	0.29901	87.707	88.864	1.157	0.32062	86.484	88.444	1.960
90	0.30382	88.007	88.857	0.850	0.32450	86.940	88.407	1.467
95	0.30444	87.697	88.706	1.008	0.32638	86.462	88.276	1.814
100	0.30370	88.152	89.196	1.044	0.32472	87.056	88.811	1.756
Average	0.30475	87.846	88.724	0.961	0.32654	86.642	88.382	1.739

speaking, the Gamma distribution fitting performs better than Gaussian distribution fitting (about 1.0% performance improvement). The theoretical accuracy (TA) error of the Gamma distribution fitting is 0.35% approximately, which is

better than the Gaussian distribution fitting (the error is about 1.7%). Besides, Fig. 4 also illustrates the performance of our method is steady, and the average accuracy is at about 89%.

## 7 Conclusion

Although many well-known video sites such as YouTube uses HTTPS, which is considered to protect user privacy, it still leaks content information of videos. In this paper, we showed that the Levenshtein distance of fragment sequences are able to assess the content similarity of encrypted video streams. We demonstrated it possible to analyse the content of encrypted video traffic with Unsupervised method. It is effective to analyze the encrypted video traffic when there is lacking the apriority knowledge.

First, a threshold was calculated by fitting with the Gamma distribution function. Our statistical analysis concluded that the threshold can determine whether two unknown video streams belong to the same video title with a probability of over 89%. Moreover, we illustrated the stability of the threshold and its judgment accuracy through further experiments. In another work, we also implemented the clustering of encrypted video streams using spectral clustering based on Levenshtein distance and achieved a good result.

Regardless, we can foresee a bright future for encrypted video stream analysis based on sequence similarity. It is necessary for unsupervised learning of encrypted video traffic. Future researchers should consider a new definition of sequence distance, which reflects the similarity of video content better.

**Acknowledgments.** This work is supported by the National Key Research and Development Program of China (No. 2018YFB0204301), Open Foundation of State Key Laboratory of Cryptology (No: MMKFKT201617), and the NUDT Research Grants (No. ZK19-38).

## References

1. Aceto, G., Ciunozzo, D., Montieri, A., Pescapé, A.: Mobile encrypted traffic classification using deep learning: experimental evaluation, lessons learned, and challenges. *IEEE Trans. Netw. Service Manag.* **16**(2), 445–458 (2019)
2. Ameigeiras, P., Ramos-Munoz, J.J., Navarro-Ortiz, J., Lopez-Soler, J.M.: Analysis and modelling of youtube traffic. *Trans. Emerg. Telecommun. Technol.* **23**(4), 360–377 (2012)
3. Anderson, B., McGrew, D.: Identifying encrypted malware traffic with contextual flow data. In: *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*. pp. 35–46 (2016)
4. Anderson, B., Paul, S., McGrew, D.: Deciphering malware’s use of tls (without decryption). *J. Comput. Virol. Hacking Tech.* **14**(3), 195–211 (2018)
5. Bagaria, S., Balaji, R., Bindhumadhava, B.: Detecting malignant tls servers using machine learning techniques. *arXiv preprint [arXiv:1705.09044](https://arxiv.org/abs/1705.09044)* (2017)
6. Bujlow, T., Carela-Español, V., Barlet-Ros, P.: Independent comparison of popular dpi tools for traffic classification. *Comput. Netw.* **76**, 75–89 (2015)

7. Cisco, C.V.N.I.: The zettabyte era-trends and analysis, 2015–2020. white paper (2016)
8. Dierks, T., Rescorla, E.: Rfc 5246-the transport layer security (tls) protocol version 1.2. Internet Engineering Task Force (2008)
9. Dubin, R., Dvir, A., Pele, O., Hadar, O.: The video streams pcap files dataset. [http://www.cse.bug.ac.il/title\\_fingerprinting/](http://www.cse.bug.ac.il/title_fingerprinting/) (2017)
10. Dubin, R., Dvir, A., Pele, O., Hadar, O.: I know what you saw last minute-encrypted http adaptive video streaming title classification. *IEEE Trans. Inf. Forensics Secur.* **12**(12), 3039–3049 (2017)
11. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: *Soviet physics doklady*. vol. 10, pp. 707–710 (1966)
12. Liu, Y., Ou, C., Li, Z., Corbett, C., Mukherjee, B., Ghosal, D.: Wavelet-based traffic analysis for identifying video streams over broadband networks. In: *IEEE GLOBECOM 2008–2008 IEEE Global Telecommunications Conference*. pp. 1–6. IEEE (2008)
13. Liu, Y., Sadeghi, A.-R., Ghosal, D., Mukherjee, B.: Video streaming forensic – content identification with traffic snooping. In: Burmester, M., Tsudik, G., Magliveras, S., Ilić, I. (eds.) *ISC 2010. LNCS*, vol. 6531, pp. 129–135. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-18178-8\\_11](https://doi.org/10.1007/978-3-642-18178-8_11)
14. Navarro, G.: A guided tour to approximate string matching. *ACM Comput. Surv.* **33**(1), 31–88 (2001)
15. Rao, A., Legout, A., Lim, Y.s., Towsley, D., Barakat, C., Dabbous, W.: Network characteristics of video streaming traffic. In: *Proceedings of the Seventh Conference on Emerging Networking Experiments and Technologies*. pp. 1–12 (2011)
16. Schuster, R., Shmatikov, V., Tromer, E.: Beauty and the burst: remote identification of encrypted video streams. In: *26th {USENIX} Security Symposium ({USENIX} Security 17)*. pp. 1357–1374 (2017)
17. Shi, Y., Ross, A., Biswas, S.: Source identification of encrypted video traffic in the presence of heterogeneous network traffic. *Comput. Commun.* **129**, 101–110 (2018)
18. 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* (2019)



# An Efficient and Revocable Auditing Scheme for the Internet of Things

Tian Jun-Feng<sup>1,2</sup> and Guo Rui-Fang<sup>1,2</sup>(✉)

<sup>1</sup> School of Cyberspace Security and Computer Institute, Hebei University,  
Baoding 071000, China  
781135676@qq.com

<sup>2</sup> Hebei Key Laboratory of High Confidence Information Systems, Hebei University,  
Baoding 071000, China

**Abstract.** With the rapid development and popularization of Internet of Things (IoT) technology, the problem of limited data storage and computing power of smart devices is becoming more and more obvious, and cloud computing can provide computing and storage services. IoT data is outsourced in cloud storage, and how to ensure the integrity of the data is worth studying. A Certificateless Provable Data Possession (CL-PDP) scheme for the Internet of Things environment is proposed. To solve the problem of weak user computing power and malicious third-party problems, a KTC fog alliance structure model is designed. The user revocation by the cloud service provider reduces the computational burden of the KGC. Finally, through theoretical analysis and experimental verification, it shows that the scheme has less calculation, revocation overhead and higher credibility than other schemes.

**Keywords:** IoT · Data integrity · CL-PDP · KTC fog alliance · User revocation · Auditing scheme

## 1 Introduction

Cloud computing has the advantages of convenience, scalability, and resource sharing. It can provide users with strong computing and storage capabilities, so it has been widely developed and applied.

In the era of rapid development of the Internet of Things, due to the limited storage capacity and computing power of IoT devices, the combination of Internet of Things and cloud computing has become an inevitable choice [1]. However, the data in Internet of Things is uploaded to cloud storage servers and the security of cloud storage will directly threaten the security of data in the Internet of Things. Therefore, the data integrity in the Internet of Things storage is worth of studying.

The Provable data possession (PDP) scheme can help the Internet of Things to remotely check the integrity of data stored in the cloud. Aiming at the existing PDP scheme, a new PDP scheme is proposed for the Internet of Things.



### 1.1 Related Work

In 2007, Ateniese et al. [2] proposed a PDP scheme. This scheme allows users to verify the integrity of data without having to retrieve and download the entire outsourced file. However, it only supports user static operations on files. In 2008, Ateniese et al. [3] proposed a PDP scheme that can support users' dynamic operations. In order to eliminate complex key and certificate management, in 2013, Zhao et al. [4] proposed the first identity-based PDP scheme (ID-PDP). In order to solve the key escrow problem, Wang et al. [5] proposed a certificateless PDP scheme (CL-PDP) in 2013. In the CL-PDP scheme, part of the user's key is generated by the user himself, and the other part is generated by the key generator (KGC). However, the revocation of illegal users by the revocable PDP scheme in the past requires group managers or third parties to participate in the calculation, which has a certain computational overhead.

In summary, the CL-PDP scheme can avoid complex certificate maintenance and key escrow problems. Considering that the user's revocation in the existing PDP solution requires group managers or third parties to participate in the calculation, how to effectively revoke the illegal users is worth studying. An efficient and revocable certificateless PDP (RE-CL-PDP) scheme for the Internet of Things is proposed, whose main contributions are as follows:

- (1) According to the definition and security model of the CL-PDP scheme, a definition and security model of the RE-CL-PDP scheme is constructed.
- (2) A KTC fog alliance structure model is designed to solve the problem of weak computing power and malicious third-party problems.
- (3) The RE-CL-PDP scheme transfers the revocation function to the CSP, thereby reducing the burden on the KGC.

## 2 The Design of the RE-CL-PDP Scheme

This section gives the system model and the definition of the RE-CL-PDP scheme, the KTC fog alliance structure, and the security model of the RE-CL-PDP scheme.

### 2.1 System Model of the RE-CL-PDP Scheme

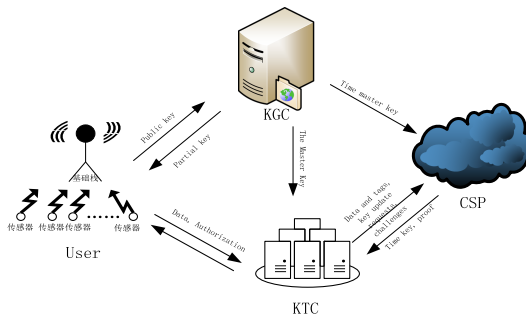


Fig. 1. System model of the RE-CL-PDP scheme

The system model of the RE-CL-PDP scheme is shown in Fig. 1. The RE-CL-PDP scheme model mainly includes four entities: KGC, User, KTC, CSP. The KGC function is to generate a public parameter for the system, a partial key and a time master key for the user. User is composed of the basic stack, which is equivalent to the intelligent device in the Internet of Things, collecting, storing or using the data collected by the sensors in the Internet of Things. KTC is a structure built using the trusted cloud platform technology of the TPM Alliance. Due to the weak computing power and storage capacity of users, KTC can help users generate data tags, send challenge information to cloud service providers, and verify the correctness of evidence. The CSP can provide storage functions for user data, generate time keys, revocation users, and generate evidence for the challenge to prove the integrity of the challenge block.

## 2.2 The Definition of the RE-CL-PDP Scheme

The RE-CL-PDP scheme includes the following basic algorithms:

**Setup** : This algorithm is executed by KGC, by inputting the security parameter  $k$ , to generate the common parameters, system master key and time master key.

**UserKeyGen** : User executes this algorithm and generates a public-private key by entering system parameters and identity ID.

**UserParKey** : KGC executes this algorithm to generate a partial key for User by inputting the system parameters, User ID and its public key.

**TimeKeyUp** : This algorithm is executed by the CSP. When the CSP receives the user's time key update request, it checks whether the User ID is in the non-revocation user list. If it not exists, it rejects the update. Otherwise, it runs this algorithm.

**TagGen** : KTC runs this algorithm. For file  $F$ , it generates one used signature key and verification key, and generates a file label for file  $F$ . Finally, it sends the file and related parameters to the CSP.

**ChalGen** : This algorithm is executed by the KTC, which generates challenge information and sends it to the CSP.

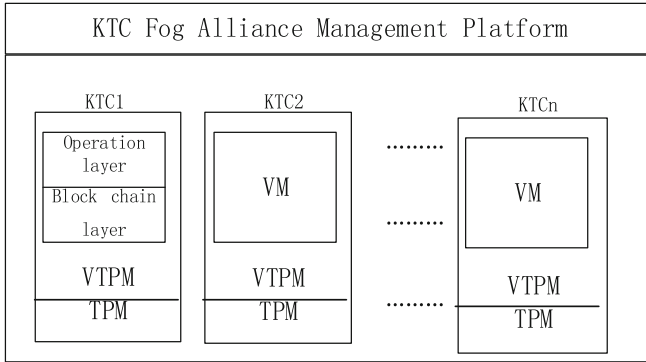
**ProofGen**: The CSP runs this algorithm. When it receives the challenge information from the KTC, which generates the corresponding evidence based on the challenge information.

**ProofVer** : When the KTC receives evidence from the CSP, the KTC runs this algorithm to verify the correctness of the evidence.

## 2.3 KTC Fog Alliance Structure

Considering the weak computing power of users and the situation of malicious third parties, the structure of KTC (FAKTC, Fog Alliance of KTC) was designed. The FAKTC is a distributed system structure model designed to implement trusted KTC based on the TPM Alliance-based Trusted Cloud Platform [6] technology and TPA Cloud Alliance [7]

technology. KTC can help users with calculations and audits to ensure that third parties are trusted and prevent third parties from colluding. The KTC Fog Alliance guarantees KTC’s trusted operating environment and honest verification process. Each member of the fog alliance interacts through P2P and is managed through the KTC fog alliance management platform. Its logical structure is shown in Fig. 2:



**Fig. 2.** KTC fog alliance structure

The KTC Fog Alliance consists of multiple KTC fog nodes. Each KTC consists of TPM (Trust Platform Module) and VTPM (Virtual Trust Platform Module) to form a trusted computing base, which provides storage, verification, calculation and other functions for trusted metrics. The TPM is a physical trusted platform module, VTPM is a virtual trusted platform module, and the TPM module is responsible for measuring VTPM to ensure its authenticity. VTPM is an extension of the TPM application that addresses the performance pitfalls of TPM. VTPM measures the static metrics in the KTC runtime environment and the baseline database. If the metrics passes, the KTC is in a trusted state. Instead, the KTC’s fog resource is suspended and the administrator is notified to process it. KTC performs corresponding operations such as data integrity auditing in the virtual machine of VTPM measurement. The KTC logical structure is divided into an operation layer and a block chain layer. The operational layer performs an integrity verification workflow. The blockchain layer receives information such as operation layer operation log records, trusted metric values, and non-revocation lists to record unchangeable blocks.

The KTC Fog Alliance Management Platform performs comprehensive management and scheduling of KTC, including trusted execution environment metrics, authentication between components, allocation of auditing schemes, and using or cancellation of KTC fog resources. Its structure is shown in Fig. 3:

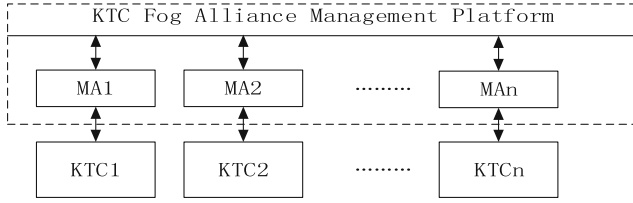


Fig. 3. KTC fog alliance management platform

From Fig. 3, we can see that the KTC fog alliance management platform contains many MA (Manager Agent) management agents. The management of the KTC fog alliance is realized by managing the corresponding KTCs by each MA. The MA structure is shown in Fig. 4:

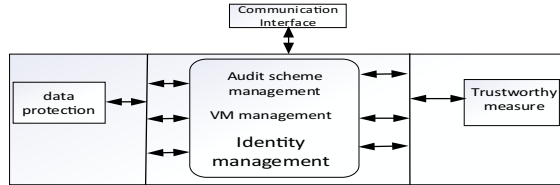


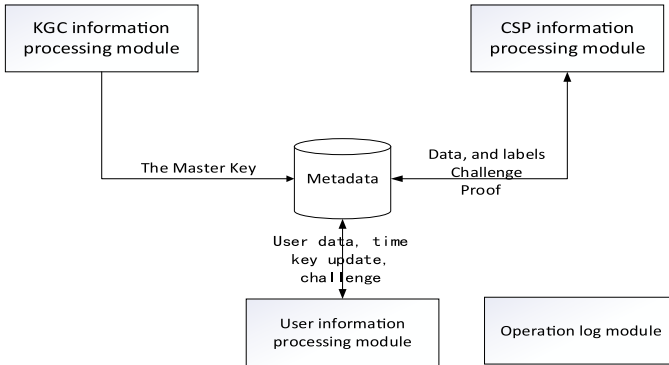
Fig. 4. MA Management Agent

The MA mainly includes an identity management component, a VM (Virtual Machine) management component, an auditing scheme component, a data protection component, and a trusted metric component. Their main functions are described below:

- (1) Identity management component: This component includes a series of authentication features. Specifically, the identity registration and logout functions of KTC members, users, and federation administrators, the identity and normal functions of each component in the platform, the system interaction of the KTC fog alliance and the management platform, the user task request, and the interaction between the user and the KTC through the data protection component and platform components interact with the blockchain through cryptographic tokens.
- (2) The VM management component: In the case that the identity authentication is passed, the VM resources in the KTC fog coalition are opened, closed, and suspended according to the state of the node. The alliance administrator performs an undo command on the VM with abnormal behavior and deducts the corresponding KTC score (the KTC integral indicates the metric value of the KTC member in the KTC fog alliance, determined by the KTC’s work efficiency, integrity measurement, and audit task metric). If the user submits an integrity verification command to the KTC, the component will match the appropriate VM resource to the user in the KTC fog coalition based on the parameter information.
- (3) Auditing scheme management component: It is responsible for storing efficient auditing programs and providing encryption functions for auditing programs.

- (4) Data protection component: It is responsible for protecting the security of user data.
- (5) Trusted metrics component: It is responsible for managing and monitoring the collection and measurement of VTPM evidence in each KTC. The collected evidence is used to check with the data of the benchmark database to implement the static integrity metrics for the KTC startup and the dynamic integrity metrics of the system runtime.

The logical structure of the operation layer is shown in Fig. 5:



**Fig. 5.** Operating layer structure

As can be seen from Fig. 5, the operation layer includes four modules: KGC information processing module, CSP information processing module, User information processing module, operation log module. The KGC information processing module mainly stores the time master key in the KTC metadata; the User information processing module mainly stores the user's data block information, time key update and challenge block information request in the metadata, and then calculates The data block label, and the data block and the label, the time key update request, and the challenge information are sent to the CSP; the CSP information processing module stores the CSP proof information, the key update information in the metadata, and returns the information to the user. The operation log module stores operation information in the operation layer.

Blockchain is a public distributed ledger with opening, unmodifiable, traceable features. The introduction of the blockchain can record information such as operational log information, trusted metrics, and non-revocation lists, forming unchangeable evidence to prevent malicious third parties. Therefore, a blockchain layer is established in the KTC logical structure.

The blockchain layer consists of multiple blockchain nodes. Each blockchain node is based on a trusted VTPM. All KTC Cloud Alliance members form a blockchain network. The consensus mechanism in the blockchain can guarantee the consistency of blockchain nodes.

The blockchain consists of a block header and a block body. The information included in the block header includes: the hash value of the previous block, and the previous block is calculated by using SHA256, and then the value is saved in the block to prevent the

modification of the previous information; The hash value of the metric information is used to represent the metric value information; the block information is recorded in the block in which the information of the selected node is stored; the timestamp is used to record the block write time, so that the block is time-series. The block includes information: metrics that record system, component, and software operational status, non-revocation lists, and operational logs.

The blockchain node performs monitoring, receiving, calculating, recording, and broadcasting in the blockchain network. According to the designed KTC structure and the characteristics of data integrity authentication, a consensus algorithm for blockchain layer is proposed. The specific steps are as follows:

- 1) The operational logging module collects trusted evidence of the working layer and broadcasts data validation information ( $T_U, T_{CSP}, T_{KTC}, T_{int}, OP$ ) in the blockchain network.  $T_U$  indicates the metric value for the cloud user, which is determined by the KTC registration duration and user behavior.  $T_{CSP}$  indicates the metric value for the cloud service provider, which is determined by the CSP for the user response and data integrity verification.  $T_{KTC}$  indicates the KTC score.  $T_{int}$  indicates the time interval at which this operation is recorded.  $OP$  indicates the operation performed by the KTC operating layer during the time interval.
- 2) All nodes independently listen to and record information broadcast in the blockchain network.
- 3) After the time  $t$ , each node sends its own monitored information and its own signature ( $T_U, T_{CSP}, T_{KTC}, T_{int}, OP, W$ ) to the blockchain network. Where  $W$  is the sum of the scores of each feature and weight.
- 4) Each node calculates the weight of each node and broadcasts the largest node information to the blockchain network.
- 5) After the node receives more than  $n$  identical response messages, a consensus can be reached. The signer in this same message packs information such as trusted metrics and operational records into new blocks. As a bonus, the node own metrics will also increase. Where  $n$  is determined by the number of nodes and the fault tolerance of the system.
- 6) After the new block is completed, each node deletes the previous information and starts the next round of consensus.

### 3 The Proposed RE-CL-PDP Scheme

This section mainly describes the implementation of this scheme.

- Setup: Given a security parameter  $k$ , KGC selects an addition cycle group  $G$  with a prime number  $q$ ,  $P \in G$  is the generator of  $G$ . KGC selects  $x, t \in Z_q^*$ , where  $x$  is the system master key and  $t$  is the time master key. KGC keeps  $x$  private and sends  $t$  to the CSP and KTC. It calculates  $X = x.p, T = t.p$  and selects hash function:  $H_i : \{0, 1\} \rightarrow Z_q^*$ ,  $i = 1, 2, 3, 4, 5$ . Finally,  $C$  sets params =  $\{k, q, P, G, X, T, H_i\}$  as system parameters and exposes them.
- UserKeyGen :The user selects  $u \in Z_q^*$  as the private key and calculates  $U = u.P$  as the public key.

– UserParKey :KGC selects  $y \in z_q^*$  and calculates  $Y = y.p, h_1 = H_1(\text{ID}||Y||U||X||T)$

$Z = y + x.h_1(\text{mod } q)$ , then  $(Y, Z)$  is sent to the user as part of the user's key.

– TimeKeyUp :When receiving the user's key update request  $(\text{ID}, t_i)$ , the CSP will views the non-revocation user list. Only when the user is a non-revocation user, it will selects  $y_T \in z_q^*$ , calculates  $Y_T = y_T.p, h_2 = H_2(\text{ID}||Y_T||t_i), Z_T = y_T + t.h_2(\text{mod } q)$ , then  $(Y_T, Z_T)$  as the user's time Key, and sends it to the user. The user can judge the correctness of the time key by calculating Eq. (1):

$$Z_T.p = Y_T + h_2.T \quad (1)$$

– TagGen :The KTC encrypts the file  $F = \{m_i\}$  and divides it into  $n$  blocks, then  $F' = \{m_i'\}$ . Select  $a_F \in z_q^*$  and calculate  $A_F = a_F.p$  as the signature and verification key for one use. Calculate  $h_2 = H_2(\text{ID}||Y_T||t_i)$ ,  $h_3 = H_3(\text{ID}||Y||U||Y_T||A_F)$ ,  $V = a_F.(h_3 + Z) + Z_T \text{mod } q$ ,

$h_4 = H_4(\text{NI}||i)$ ,  $\sigma_i = (m_i' + a_F.h_4).p$ ,  $D = Z.p, V_F = a_F.D$ . Where NI is the file name,  $\sigma_i$  is the label. Finally, the user sends the relevant parameters  $\theta = \left\{ \left\{ m_i' \right\}, \left\{ \sigma_i \right\}_{i=1}^n, V, A_F, V_F \right\}$  such as files and tags to the CSP.

– ChalGen :The KTC selects a set  $I \in \{1, 2, 3 \dots, n\}$ ,  $w_i \in z_q^*$ , where  $i \in I$ . Then the KTC sends the challenge information  $\{i, w_i\}_{i \in I}$  to the CSP.

– ProofGen :After the CSP receives the challenge information, select  $x_c \in z_q^*$  and calculate

$$\sigma_c = \sum_{i \in I} w_i \cdot \sigma_i, X_c = x_c \cdot V_F, \alpha_c = H_5(\text{ID}||Y||U||Y_T||A_F||X||T||X_c||\sigma_c), \varepsilon = \alpha_c \cdot x_c,$$

$\varphi = \sum_{i \in I} w_i \cdot m_i'$ , Then the CSP will prove that  $\rho = \{A_F, V_F, X_c, \sigma_c, \varepsilon, \varphi, V\}$  are sent to the verifier.

– ProofVer :When the KTC receives the certificate from the CSP, it calculates  $\alpha_c$ ,  $h_3$ ,  $h_2$ ,  $h_4$ , and finally judges whether the following Eq. (2) holds, to judge the correctness of the proof:

$$\varepsilon(V.p - h_3.A_F - T.h_2 - Y_T) + T_c = A_F \cdot \sum_{i \in I} w_i.h_4 + \alpha_c.X_c + \varphi.p \quad (2)$$

## 4 Performance Analysis

In this section we compare and analyze the RE-CL-PDP program with other schemes. The hardware platform is windows xp operating system, the server uses 3 GHZ PIV processor and 512 MB RAM. In order to achieve a safe level, the security level of the 1024-bit RSA algorithm is implemented, and the super-singular curve E on the finite field  $F_p$  is used in the evaluation, wherein the length of P is 512 bits, and the order q of E is 160 bits.

## 4.1 Characteristic Analysis

First, the RE-CL-PDP scheme and other schemes are analyzed in terms of characteristics. The results are shown in Table 1:

**Table 1.** Characteristic analysis

	Public verification	Revocable	Certificateless	Non-linear linear pair operation	Trust level
Scheme [8]	✓	✓	X	X	I
Scheme [9]	✓	✓	✓	X	II
Scheme [10]	✓	X	✓	X	II
RE-CL-PDP	✓	✓	✓	✓	III

## 4.2 Computational Overhead

We assume that the user file is divided into  $n$  blocks and the number of challenge blocks is  $c$  blocks. In this part, the computational cost of the three stages of label generation phase, evidence generation and verifier verification phase in the proposed scheme is compared with the other schemes. We define the symbol for the correlation operation, using  $T_p$  to represent a bilinear pairing operation time, using  $T_{exp}$  to represent an exponential operation time, using  $T_{mul}$  to represent the multiplication time on  $G$ , and using  $T_h$  to represent the operation time of a hash to the point. Other operations such as point addition on  $G$ ,  $z_q^*$  multiplication, and Original hash operations have less computational time, so we ignore them when evaluating.

In scheme [8], the user's computational overhead in the label generation phase is  $2nT_{exp} + nT_{mul} + nT_h$ , the computational overhead required for CSP to generate evidence is  $(c - 1)T_{mul} + cT_{exp}$ , and the computational cost required for the verifier to verify the correctness of the evidence is  $cT_h + (c + 3)T_{mul} + (c + 3)T_{exp} + 2T_p$ . The total computational overhead is represented by  $T_{16}$ , then

$T_{16} = (n + c)T_h + (n + 2c + 2)T_{mul} + (2n + 2c + 3)T_{exp} + 2T_p$ . The three-stage computational cost of scheme [9] is  $2nT_{exp} + nT_{mul}$ ,  $(c - 1)T_{mul} + cT_{exp}$ ,  $(c + 3)T_{mul} + (c + 1)T_{exp} + 3T_p$ . The total computational overhead is represented by  $T_{24}$ , then  $T_{24} = (n + 2c + 2)T_{mul} + (2n + 2c + 1)T_{exp} + 3T_p$ . The three-stage computational cost of scheme [10] is  $2nT_h + 2nT_{mul}$ ,  $(c + 1)T_{mul}$ ,  $(c + 1)T_h + (c + 5)T_{mul} + 2T_p$ . The total computational overhead is represented by  $T_{21}$ , then  $T_{21} = (2n + 2c + 6)T_{mul} + (2n + c + 1)T_h + 2T_p$ . The three-stage computational cost of RE-CL-PDP scheme is  $(n + 4)T_{mul}$ ,  $(c + 1)T_{mul}$ ,  $7T_{mul}$ . The total computational overhead is represented by  $T_{Ours}$ , then  $T_{Ours} = (n + c + 12)T_{mul}$ . The calculation overhead of each scheme in these three phases is shown in Table 2:



**Table 2.** Computational overhead

	Label generation	Evidence generation	Evidence verification	Total computational overhead
Scheme [8]	$2nT_{exp} + nT_{mul} + nT_h$	$(c - 1)T_{mul} + cT_{exp}$	$cT_h + (c + 3)T_{mul} + (c + 3)T_{exp} + 2T_p$	$T_{16} = (n + c)T_h + (n + 2c + 2)T_{mul} + (2n + 2c + 3)T_{exp} + 2T_p$
Scheme [9]	$2nT_{exp} + nT_{mul}$	$(c - 1)T_{mul} + cT_{exp}$	$(c + 3)T_{mul} + (c + 1)T_{exp} + 3T_p$	$T_{24} = (n + 2c + 2)T_{mul} + (2n + 2c + 1)T_{exp} + 3T_p$
Scheme [10]	$2nT_h + 2nT_{mul}$	$(c + 1)T_{mul}$	$(c + 1)T_h + (c + 5)T_{mul} + 2T_p$	$T_{21} = (2n + 2c + 6)T_{mul} + (2n + c + 1)T_h + 2T_p$
RE-CL-PDP	$(n + 4)T_{mul}$	$(c + 1)T_{mul}$	$7T_{mul}$	$T_{RE-CL-PDP} = (n + c + 12)T_{mul}$

We calculate the difference between the total computational cost of the scheme [8] and the total computational cost of the RE-CL-PDP scheme:

$$\Delta T_8 = (c - 10)T_{mul} + (2n + 2c + 3)T_{exp} + 2T_p \approx 22.40n + 28.78c + 9.82$$

Similarly, the difference calculation is calculated with the calculation cost of the scheme [9]:

$$\Delta T_9 = (c - 10)T_{mul} + (2n + 2c + 1)T_{exp} + 3T_p \approx 22.40n + 28.78c + 8.43$$

Similarly, the difference calculation is calculated with the calculation cost of the scheme [10]:

$$\Delta T_{10} = (n + c - 6)T_{mul} + (2n + c + 1)T_h + 2T_p \approx 12.46n + 9.42c + 8.77$$

Since both  $n$  and  $c$  are positive, the above results are positive. Therefore, the RE-CL-PDP scheme has less computational overhead than the other three schemes.

### 4.3 User Revocation

This section focuses on the overhead of scheme and proposed scheme [9] in terms of user revocation. The number of users revocation in the scheme is denoted by  $N$ . The revocation of the illegal user in the scheme [9] mainly comes from the two parameter overheads generated by the non-revocation user, the two parameter overheads generated by the illegal user, and the overhead of the cloud service to convert the user's file label. They are  $T_{exp}, T_{exp} + T_{mul}, 2nT_{exp} + nT_{hash} + nT_{mul}$  respectively. The total cancellation cost is  $T_{RE[9]} = N \cdot \{(2n + 2)T_{exp} + nT_{hash} + (n + 1)T_{mul}\}$ . It can be seen that it is related to the number of user revocation and the number of user data blocks. The user revocation of the RE-CL-PDP scheme is mainly to refuse to revoke the user's time key update, and there is almost no computational overhead. When an illegal user requests to

update the time key, the trusted cloud platform queries the non-revocation user list, and if the user is in the table, updates the time key of the user; otherwise, the time key update of the user is rejected. Therefore, the revocation cost of this scheme is more efficient than the scheme [9].

## 5 Conclusion

This paper proposes a new highly efficient and revocable auditing scheme for the Internet of Things. The scheme solves the problem of limited data storage and computing power of IoT devices, and realizes the integrity verification of IoT data. The system model and security model of the scheme are given, the trust level of the third party is improved, the efficiency of the scheme is improved, and the credibility of the scheme is guaranteed. Finally, experiment shows that the scheme is more efficient.

## References

1. Zhu, C., Sheng, Z., Leung, V.C.M., Shu, L., Yang, L.T.: Toward offering more useful data reliably to mobile cloud from wireless sensor network. *IEEE Trans. Emerg. Topics Comput.* **3**, 84–94 (2015)
2. Ateniese, G., Burns, R., Curtmola, R., et al.: Provable data possession at untrusted stores. In *Proceedings of the 14th ACM Conference Computer Communication Security (CCS 2007)*, Alexandria, Virginia, USA, pp. 598–609 (2007)
3. Ateniese G., Pietro R.D., Mancini, L.V, et al.: Scalable and efficient provable data possession. In: *Proceeding of the 4th International Conference on Security Privacy Communication Networks. (SECURECOMM)*, NY, USA, pp. 1–10 (2018)
4. Zhao, J., Xu, C., Li, F., et al.: Identity-based public verification with privacy-preserving for data storage security in cloud computing. *IEICE Trans. Fund. Elect. Commun. Comput. Sci.* **96**(12), 2709–2716 (2013)
5. Wang, B., Li, B., Li, H., Li, F.: Certificateless public auditing for data integrity in the cloud. In: *Proceedings 2013 IEEE Conference Commun. Networks Security*, pp. 136–144,(2013)
6. Tian, J., Chang, F.: Trusted Cloud Platform management model based on Tpm alliance. *Trans. Commun.* **37**(2), 1–10 (2016)
7. Tian, J., Li, T.: Data integrity verification model based on TPA cloud alliance. *Trans. Commun.* **39**(8), 113–124 (2018)
8. Zhang ,Y., Yu, J., Hao, R., et al.: Enabling efficient user revocation in identity-based cloud storage auditing for shared big data. *IEEE Trans. Dependable Secur. Comput.* (99), 1–1,(2018)
9. Li, J., Yan, H., Zhang, Y.: Certificateless public integrity checking of group shared data on cloud storage. *IEEE Trans. Services Comput.* 1–1 (2018)<https://doi.org/10.1109/tsc.2018.2789893>
10. He, D.: Certificateless provable data possession scheme for cloud-based smart grid data management systems. *IEEE Trans. Ind. Informat.* **14**(3), 1232–1241 (2018)



# A Configurable off-Policy Evaluation with Key State-Based Bias Constraints in AI Reinforcement Learning

Shuoru Wang<sup>1</sup>, Jiqiang Liu<sup>1</sup>, Tong Chen<sup>1</sup>, He Li<sup>1</sup>, Wenjia Niu<sup>1</sup>(✉), Endong Tong<sup>1</sup>(✉), Long Li<sup>2</sup>, and Minglu Song<sup>1</sup>

<sup>1</sup> Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, 3 Shangyuan Village, Haidian District, Beijing 100044, China

{Niuwj, Tonged}@bjtu.edu.cn

<sup>2</sup> Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

**Abstract.** In reinforcement learning field, off-policy evaluation(OPE), a core task to learn a new policy from existing trajectory data of real policy to evaluate, is highly important for real policy deployment before policy running, avoiding unexpected dangerous or expensive agent actions. Among existing methods, the return value of a trajectory is calculated through Markov decision process (MDP)-based rewards summation of sequential states' actions, and the aim of a new policy is to achieve the minimum variances compared with return values from existing trajectory data. However, such methods ignore to guide the influence of key states in OPE, which are critical to success and should be set with more preference as well as the return value bias. In this paper, we develop a configurable OPE with key state-based bias constraints. We first adopt FP-Growth to mine the key states and get corresponding reward expectations of key states. Through further configuring every reward expectation scope as bias constraint, we then construct new goal function with the combination of bias and variance and realize a guided importance sampling-based OPE. Taking the GridWorld game as our experiment platform, we evaluate our method with performance analysis and case studies, as well as make comparisons with mainstream methods to show the effectiveness.

**Keywords:** OPE · Reinforcement Learning · Trajectory · Bias expectation · Key state

## 1 Introduction

As a highly important branch of machine learning, Reinforcement Learning (RL) is a framework that allows agents to learn policy and make best decisions based on massive real experiences and feedbacks, when facing the environment unable to dynamic

---

This work was supported by the National Natural Science Foundation of China (61972025, 61802389, 61672092, U1811264, 61966009), the Fundamental Research Funds for the Central Universities of China (2018JBZ103, 2019RC008), Science and Technology on Information Assurance Laboratory, Guangxi Key Laboratory of Trusted Software (KX201902).

© Springer Nature Singapore Pte Ltd. 2020

Y. Xiang et al. (Eds.): SocialSec 2020, CCIS 1298, pp. 120–131, 2020.

[https://doi.org/10.1007/978-981-15-9031-3\\_11](https://doi.org/10.1007/978-981-15-9031-3_11)

planning. Similar to supervised learning, RL also has a learning ability on off policies from historical trajectory data, called off-policy reinforcement learning. For large-scale environment, such learning is useful due to allowing utilize former experience. to iterate without the need for an entity environment. With off-policy reinforcement learning, we can train multiple models using the same historical data collected by previous agents, and then select the best model.

For off-policy reinforcement learning, the quality learned from new agents. It is necessary to do OPE before any RL usage, since a bad policy might be too infeasible, expensive, or even hazardous, especially in some fields such as medicine, finance, advertising, and education, to name a few [1–4].

For RL off-policy evaluation, there are two predominant methods: the direct method (DM) and importance sampling (IS) [5]. DM first fitted the dynamics model of the system and then estimated the mean reward of the target policy to evaluate by estimator. Such estimators generated show a low variance most time. However, when the sample size is relatively small compared to the function complexity, a large deviation will occur [6]. Lately, the IS, also known as the inverse propensity score method in statistical causal inference was proposed [7]. The IS aims to correct the mismatch between the distributions generated by the behavior and the target policies [8, 9]. IS estimators are unbiased and strongly consistent, but the variance tends to be high if the behavior and evaluation policies differ significantly [10], or the evaluation policy is deterministic, growing exponentially with the horizon. Currently, Doubly Robust (DR) estimator is the state-of-art OPE, that combines a DM estimator with an IS estimator [5, 10, 11], having less bias than the estimator of DM and less variance than the estimator of IS.

Among existing methods, the return value of a trajectory is calculated through the Markov decision process (MDP)-based rewards summation of sequential states' actions, and the aim of a new policy is to achieve the minimum variances compared with return values from existing trajectory data. However, such methods ignore to guide the influence of key states in OPE, which are critical to success and should be set with more preference as well as the return value bias. More specifically, when all trajectories have the same initial state, we have to evaluate the performance of an OPE estimate  $\widehat{V}^{\pi^e}(S_0)$ ,  $S_0$  is an initial state. By calculating mean square error (MSE) from  $S_0$ :  $\left[ \mathbb{E}_{S_0} \widehat{V}^{\pi^e}(S_0) - \mathbb{E}_{S_0} V^{\pi^e}(S_0) \right]^2$ . But the initial state does not completely represent the real value of all states, especially the key states. especially ignoring the influence from the key states.

Aiming on this problem, we try to develop a configurable OPE with key state-based bias constraints. FP-Growth from data mining field is adopted to mine the key states and get corresponding reward expectations of key states. The big challenge is how to construct new goal function with the combination of bias and variance and realize a guided importance sampling-based OPE. In this paper, we define, KCBC.

Taking the GridWorld game as our experiment platform, we evaluate our method with performance analysis and case studies. We summarize our contributions as follows:

1. We are the first to consider guiding the bias impact of key states on offline strategy evaluation, which help OPE meet more real RL scenario and reduce the influence from trajectory noises.

2. We find that FP-Growth is effective for key state mining among trajectories and experiments.

The outline of the rest of this paper is as follows. We present symbol definitions related to off-policy evaluation and Markov, and FP-Growth in Sect. 2. Section 3 section describes in detail configurable OPE including discovering critical state, defining objective function constraints and model training. Section 4 conducts extensive experiments on GridWorld to validate the effectiveness of our method. Section 5 summarizes related work, while Sect. 6 outlines our conclusions and plans for future work.

## 2 Preliminary

We adopt notational standard MDPNv1 for Markov decision processes [12], with state space  $S$ , transition probabilities  $P$ , action space  $A$ , reward function  $R$ , discount factor  $\gamma \in (0, 1]$ , horizon  $L$ , and initial state distribution  $P_0$ [13]. A trajectory  $H$  is a series of states  $S_t$ , actions  $A_t$ , and rewards  $R_t$ :  $H = (S_0, A_0, R_0, \dots, S_{L-1}, A_{L-1}, R_{L-1})$ . The return of trajectory  $H$  is  $g(H) := \sum_{t=0}^{L-1} \gamma^t R_t$ . we use the average cumulative discounted reward:  $V(\pi) := E_{P,\pi}[g(H)|H \sim \pi]$ , where  $H \sim \pi$  is the policy  $\pi$  generated trajectory  $H$ . The state value function is:  $V^\pi(s) := E_{P,\pi}[\sum_{t=0}^{\infty} \gamma^t R_t | S_0 = s, \pi]$ . The state-action value function is:  $Q^\pi(s, a) := E_{P,\pi}[\sum_{t=0}^{\infty} \gamma^t R_t | S_0 = s, A_0 = a, \pi]$ .

In off-policy evaluation, we are given an evaluation policy  $\pi_e$ , and the known policies (behavior policies) were used to generated historical data,  $D$ , be a set of  $m$  trajectories:  $D := \{H_i, \pi_i^b\}_{i=1}^m$  where  $H_i \sim \pi_i^b$ .

Notice the  $V(\pi)$  and  $V^\pi$  are related:  $V(\pi) = \sum_{s \in S} Pr(S_0 = s) V^\pi(s)$ . From this function, we can get the value function of the initial state of a trajectory from off-policy evaluation. When the OPE is used, the average cumulative discounted reward of the two policies  $\pi_1^e$  and  $\pi_2^e$ , is the same according to the same historical trajectory:  $V(\pi_1^e) = V(\pi_2^e)$ , but it does not completely mean that the two policies are the same. It is possible that the policy  $\pi_1^e$  has a large bias at the key state and a small bias at other points and the second policy  $\pi_2^e$  has a small bias at the key state and a large bias at other points. Most of the key states are not in the initial state, so the value function evaluated according to the OPE cannot indicate the value of the key states.

## 3 Key State-Based OPE

### 3.1 Key States Mining

The goal of OPE is to estimate the average cumulative discounted reward:  $V(\pi)$ , which is a state value. The more times a particular trajectory appears, the more weight it gets [14]. Then the states in these particular trajectories are critical to OPE. To select the key states, we introduce the algorithm of FP-Growth.

FP-Growth algorithm adopts the frequent pattern tree (FP-tree). FP-tree is an improved trie structure so that each set of items is stored as a string in the trie along with its frequency. On each node of trie, store the item, count, and next fields. The items in

the path from the root of the trie to the node constitute the item set stored on the node, and the count is the frequency of the item set. The next node link is a pointer to the next node in the FP-tree that has the same item. The field parent contains a pointer to the parent node and the root node is empty [15].

FP-tree carries the complete information required for frequency mining in a compact manner; The height of the tree is bounded by the maximum number of frequent items in the transaction [10]. For example, suppose  $p$  is the end of state or catastrophic state. If the support degree is 4, and the number of frequent items is 1. we consider the path  $\langle f : 4, c : 3, a : 3, m : 2, p : 2 \rangle$  in FP-Tree. The frequent item is  $\langle f : 4 \rangle$ . Then the key state is  $f$ .

### 3.2 Goal Function Construction

How to get its reward expectation after obtaining the key state through FP is a problem. The previous method is to estimate the reward expectation of a trajectory, which can also be said to be the reward expectation of the initial state of the trajectory. The first time it was proposed to use the minimum mean square error as the constraint is MAGIC, which is an estimator combines DM and WDR. The estimator corresponds to use WDR with index  $t < j$ , DM with index  $t \geq j$  for some  $0 \leq j \leq L$ . The definition of off-policy  $j$ -step return is:  $g_j := \sum_{i=1}^m g_i^{(j)}$ ,  $g_i^{(j)}$  for each trajectory  $i$  is defined as:

$$g_i^{(j)} := \sum_{t=0}^j \gamma^t w_t(H_i) R_t^{H_i} + \gamma^{j+1} w_j(H_i) \widehat{V}^{\pi_e}(S_{j+1}^{H_i}) - \sum_{t=0}^j \gamma^t (w_t(H_i) \widehat{q}^{\pi_e}(S_t^{H_i}, A_t^{H_i}) - w_{t-1}(H_i) \widehat{V}^{\pi_e}(S_t^{H_i})) \quad (1)$$

Note that  $g_0$  is the DM estimator,  $g_L$  is equal to the WDR estimator. The last component is the combined control variate for the importance of sampling and model-based term. Hence, when  $j$  increases, the bias will decrease, at the cost of an increase in variance. The estimator is a convex combination of the partial importance sampling estimators  $g_i$ [11]. The convex combination minimizes MSE, that is we want to use as estimator  $(x^*)^T g$ , with  $g = (g_0, \dots, g_L)$ , where:

$$\begin{aligned} x^* &= \arg \min_{\substack{0 \leq x \leq 1 \\ \sum_{j=0}^T x_j = 1}} MSE(x^T g, V^{\pi_e}) \\ &= \arg \min_{\substack{0 \leq x \leq 1 \\ \sum_{j=0}^T x_j = 1}} \left\{ Bias^2(x^T g) + Var(x^T g) \right\} \\ &= \arg \min_{\substack{0 \leq x \leq 1 \\ \sum_{j=0}^T x_j = 1}} x^T \left[ \Omega_m + b_m b_m^T \right] x \end{aligned} \quad (2)$$

where  $m$  is the number of trajectories in  $D$ .

$\Omega_m$  is the covariance matrix:

$$\Omega_m(i, j) = Cov(g^{(i)}(D), g^{(j)}(D)) \quad (3)$$

$\Omega_m(i, j)$  can be estimated for  $g^{(i)}$  the sample covariance matrix  $\widehat{\Omega}_m$ , and  $b_m$  is the vector with:

$$b_n(j) = E\left[g^{(j)}(D) - V^{\pi_e}\right] \quad (4)$$

for all  $j = 0, \dots, L$ . The bias  $b_m$  can be estimated by the distance from  $g^{(j)}(D)$  to a  $CI(g^L(D), \delta)$ , a  $1 - \delta$  confidence interval on the  $g_L$ .

A configurable OPE with key state-based bias estimator which we call KSBC. Getting reward expectations of key states is to estimate the off-policy return for an arbitrary state  $s$ . Let  $n$  be the number of episodes containing state  $s$ , and let  $t_m$  is the first time when  $s_{t_m} = s$  in the  $n$ th of these episodes. Then the definition of off-policy state  $s$  return is:

$$g(s) := \frac{1}{n} \sum_{i=1}^n \left[ \sum_{t=t_m}^j \gamma^t w_t(H_i) R_t^{H_i} + \gamma^{j+1} w_j(H_i) \widehat{V}^{\pi_e}(S_{j+1}^{H_i}) - \sum_{t=t_m}^j \gamma^t (w_t(H_i) \widehat{q}^{\pi_e}(S_t^{H_i}, A_t^{H_i}) - w_{t-1}(H_i) \widehat{V}^{\pi_e}(S_t^{H_i})) \right] \quad (5)$$

Estimating the bias vector is challenging because it has a strong dependence on the value,  $V^{\pi_e}$  [11]. The bias of reward expectations of key states,  $g(s)$ , by its distance to a  $\delta$ -confidence interval for  $g_T$  obtained by bootstrapping it, for some  $\delta \in (0, 1)$ , like MAGIC to estimate bias [11]. We propose a new weighting scheme: a combination of bias in the key states and the mean squared error. This new weighting scheme is:

$$x^* = \arg \min_{\substack{0 \leq x \leq 1 \\ \sum_{j=0}^T x_j = 1}} \left( MSE(x^T g, V^{\pi_e}) + \text{Bias}(x^T g(s)) \right) \quad (6)$$

### 3.3 Model Training

The integration step follows the MAGIC procedure. we also use as estimator  $(x^*)^T g$ , with  $g = (g_0, \dots, g_L)$ , but in solving for  $x^*$  we add new constraints. We present the pseudo-code of the procedure as Algorithm 1.

The covariance matrix  $\Omega_m$ , using the sample covariance matrix of base estimators  $g_k^{(i)}$  and  $g_l^{(i)}$ . The covariance matrix is given [16], as follow:

$$\widehat{\Omega}_{k,l} := \frac{m}{m-1} \sum_{i=1}^m \left( g_k^{(i)} - m^{-1} \sum_{i=1}^m g_k^{(i)} \right) \times \left( g_l^{(i)} - m^{-1} \sum_{i=1}^m g_l^{(i)} \right) \quad (7)$$

We propose a variant of it, in the algorithms to minimize the mean square error, configuring every reward expectation scope as bias constraint, which we call KSBC.

In summary, we first adopt FP-Growth to mine the key states and get corresponding reward expectations of key states. Through further configuring every reward expectation scope as bias constraint, we then construct new goal function with the combination of bias and variance and realize guided importance sampling-based OPE.

---

**Algorithm 1** KSBC

---

1: Input:

- D: Historical data
- $\pi_e$ : Evaluation policy
- B: The number of bootstrap samples
- K: The set of return lengths to consider
- d: the support degree
- n: the number of frequent

2: Allocate  $D_i, b \in \{1, \dots, B\}$ ,  $D_b$  can hold m trajectories

3:  $S \leftarrow \text{FP-Growth}(d, k)$

4: for  $k = 1$  to K do

5: compute  $g_k^{(b)}, g_k^{(b)}(S)$

6: for  $l=1$  to K do

7: compute  $\hat{\Omega}_{k,l}$

8: end for

9:  $CI(\alpha) \leftarrow [\text{percentile}(\{g_k^{(b)}: b\}, \alpha), \text{percentile}(\{g_k^{(b)}: b\}, 1 - \alpha)]$

11:  $\hat{b}_{m,k} \leftarrow \text{distance}(g_k, CI(\alpha))$

12:  $\hat{b}_{n,k} \leftarrow \text{distance}(g_k(S), CI(\alpha))$

12: end for

13:  $x \leftarrow \arg \min_{0 \leq x \leq 1} x^T [\Omega_m + b_m b_m^T + b_n] x$   
 $\sum_{j=0}^T x_j = 1$

14: return  $x^T g$

---

## 4 Experiments

Throughout this section, we demonstrate the effectiveness of a configurable OPE with key state-based bias constraints by comparing it with other methods of OPE in various RL environments.

### 4.1 GridWorld Setting up

GridWorld: this grid is a  $4 \times 4$  GridWorld used in prior off-policy evaluation research [11, 16], when a final state is reached or an episode ending at L. The reward of state  $S_8$  is +1,  $S_{12}$  with +10,  $S_6$  where the agent is punished with -10 reward, and the rest of reward is -1.

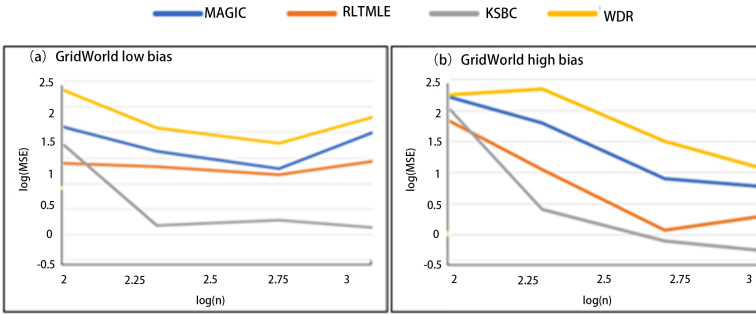
In evaluating our estimator, we explore how different degrees of model misspecification, sample size, and horizon. We start with a low level of model misspecification,  $b_0 = 0.005$ . Then, we increase model misspecification to  $b_0 = 0.05$ . We test the performance of estimators to the number of episodes in D with  $m = \{100, 200, 500, 1000\}$ . In addition, we test sensitivity to the horizon with  $L = \{10, 100\}$  for GridWorld.

We compare our estimator to WDR, MAGIC, as Thomas and Brunskill [11] demonstrate improved performance over all simulations (IS step-IS, WIS, step-WIS, and DR) in RL environments considered. In evaluation of our estimator, we test a different number of episodes in D. We implement the same various degree of model misspecification as in previous work [16].



### 4.2 Bias and Variance Evaluation

The purpose of these experiments is to demonstrate the performance improvement of our estimator. The GridWorld MSE by varying sample size and model misspecification can be found in Fig. 1. We use Table 1 and Table 2 to show the mean square error, as well as the bias. For the same number of trajectories, our method is superior to other estimators in both deviation and mean square error.



**Fig. 1.** Empirical results in GridWorld environments and varying level of model misspecification. (a) bias equivalent to  $b_0 = 0.005$ ,  $L = 100$ ; (b) bias equivalent to  $b_0 = 0.05$ ,  $L = 100$

**Table 1.** Empirical results in GridWorld environment low bias.

N	Estimator	MSE	Bias
100	WDR	99.30658	4.89658
	MAGIC	69.26160	5.00235
	RLTMLE	65.78726	4.70567
	KSBC	39.29339	3.34366
200	WDR	78.09318	6.32840
	MAGIC	26.21697	2.98726
	RLTMLE	20.01185	2.72565
	KSBC	11.97674	1.99773
500	WDR	73.52007	0.82319
	MAGIC	69.26160	5.00235
	RLTMLE	65.78726	4.70567
	KSBC	8.17901	-0.41855
1000	WDR	32.93048	2.90256
	MAGIC	27.86813	-1.19897
	RLTMLE	4.92661	1.28668
	KSBC	4.04298	1.08952

**Table 2.** Empirical results in GridWorld environment high bias.

N	Estimator	MSE	Bias
100	WDR	224.87392	0.74625
	MAGIC	97.90734	-1.12258
	RLTMLE	43.48329	0.63197
	KSBC	61.04025	2.81988
200	WDR	84.69244	8.67259
	MAGIC	10.76192	3.20873
	RLTMLE	16.35650	3.85293
	KSBC	10.02197	3.09867
500	WDR	81.66024	-1.19897
	MAGIC	19.99907	-0.98246
	RLTMLE	17.46223	-0.51929
	KSBC	8.17901	-0.41855
1000	WDR	157.84966	-2.60290
	MAGIC	27.86813	-1.19897
	RLTMLE	2.64773	0.68033
	KSBC	0.95131	0.18459

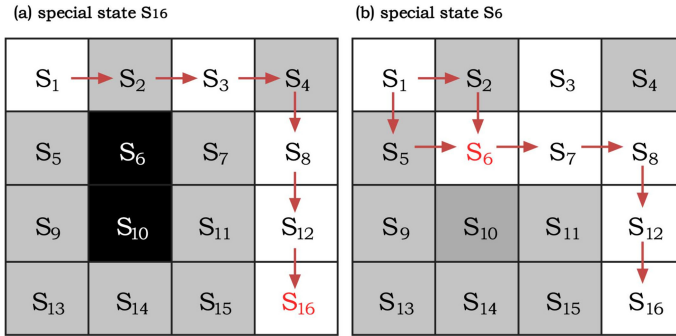
### 4.3 Efficiency Evaluation

We adopt FP-Growth to mine the key states. the experimental results (Fig. 2), as we have observed in the search for the final state  $S_{16}$  frequent item sets, the  $S_8$  have a positive reward will appear many times in the track, so there is a higher weight, and  $s_6$  has a negative incentive, agent when choosing action, will reduce the probability. In addition, through FP-Growth, we can get other frequent items, such as  $S_3$  and  $S_{12}$ . We can also find frequent item sets with negative reward states.

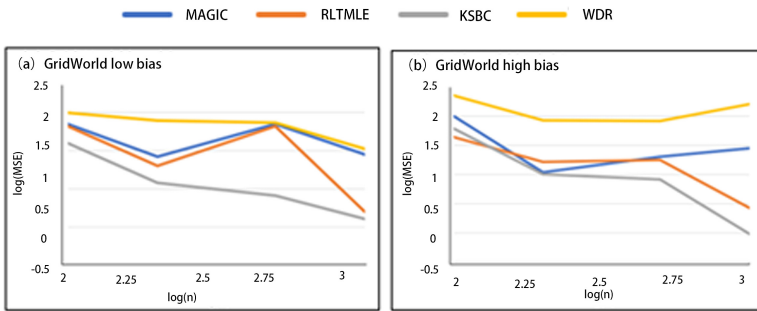
In general, we think the states with a positive reward are crucial, not frequent state. Then, we consider the performance of our estimator in the case where the agent transitions to the state, it gets a positive reward, is infrequent. The state of  $s_8$  is a positive reward, we implement the behavior policy that the agent rarely transitions to the state  $s_8$  (Fig. 3).

### 4.4 Parameter Discussion

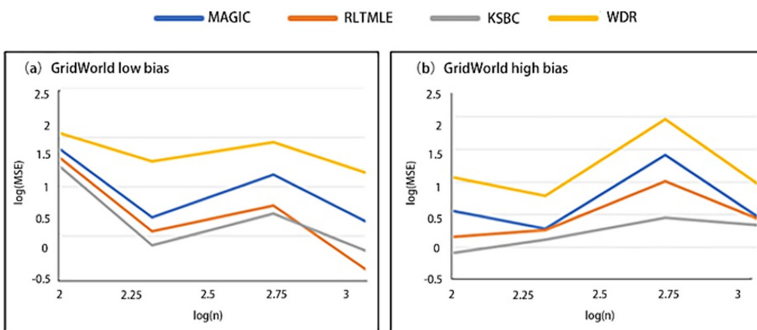
In some previous experimental results, experiments with different track lengths were rarely carried out. In order to verify the effectiveness of our method, we changed the length of the track level to conduct experiments. The experimental results are shown in Fig. 5. We can see that our estimator performs better in the segment path, which may be due to the fact that the key states appear less frequently in the shorter path. By increasing its weight through our method, we can obtain a smaller mean square error (Fig. 4).



**Fig. 2.** Key states in GridWorld environments by different states. The first one is  $S_{16}$  and the second one is  $S_6$ . The color at the bottom of the grid represents the support degree of the grid, and the color ranges from dark to light, indicating the support degree from low to high. The states indicated by the arrows constitute frequent trajectories. (Color figure online)



**Fig. 3.** Empirical results in positive reward states are rare in GridWorld environments. (a) bias equivalent to  $b_0 = 0.005$ ,  $L = 100$ .; (b) bias equivalent to  $b_0 = 0.05$ ,  $L = 10$ .



**Fig. 4.** Empirical results in positive reward states are rare in GridWorld environments. (a) bias equivalent to  $b_0 = 0.005$ ,  $L = 10$ .; (b) bias equivalent to  $b_0 = 0.05$ ,  $L = 10$ .

## 5 Related Work

### 5.1 Importance Sampling

Importance Sampling is an unbiased estimate for reweighting returns from behavior policies by averaging the following function of each trajectory  $H_i$  in the data  $D$ : define the per-step importance sampling weight as  $\rho_t(H) := \pi_e(a_t|s_t)/\pi_b(a_t|s_t)$ . The IS estimator, and a step-wise version is thus defined as:

$$V_{IS}(H_i) := \prod_{t=0}^{L-1} \rho_t(H_i) \cdot g(H_i) \quad (8)$$

$$V_{step-IS}(H_i) := \sum_{t=0}^{L-1} \gamma^t \prod_{t'=0}^{L-1} \rho_{t'}(H_i) \cdot R_t \quad (9)$$

Given a dataset  $D$ , a set of  $m$  trajectories, the IS estimator is the average estimate of the trajectories, as  $\frac{1}{m} \sum_{i=0} V_{IS}(H_i)$ . In general, IS (even in the step-wise version) suffers from very high variances and can easily grow exponentially in horizon.

Weighted importance sampling (WIS) is a variant of IS that has lower variance but is a biased estimator of the value. the average cumulative importance sampling weight at horizon  $L$  in a dataset  $D$ , is defined as  $w_t(H_i) = \sum_{i=1}^m \prod_{t'=0}^{L-1} \rho_{t'}(H_i)/m$ . The trajectory-wise and step-wise WIS are given as follows:

$$V_{WIS}(H_i) := \prod_{t=0}^{L-1} \rho_t(H_i)/w_t(H_i) \cdot g(H_i) \quad (10)$$

$$V_{step-WIS}(H_i) := \sum_{t=0}^{L-1} \gamma^t \prod_{t'=0}^{L-1} \rho_{t'}(H_i)/w_t(H_i) \cdot R_t \quad (11)$$

### 5.2 Doubly Robust

In the MDP setting, Jiang and Li [5] the first propose a doubly robust estimator. The doubly robust estimator suppose we have an estimated reward function by using the MDP approximation model. The DR estimator is thus defined as:

$$\begin{aligned} DR(D) := & \sum_{i=1}^m \sum_{t=0}^{L-1} \gamma^t w_t(H_i) R_t^{H_i} \\ & - \sum_{i=1}^m \sum_{t=0}^{L-1} \gamma^t \left( w_t(H_i) \hat{q}^{\pi_e}(S_t^{H_i}, A_t^{H_i}) - w_{t-1}(H_i) \hat{V}^{\pi_e}(S_t^{H_i}) \right) \end{aligned} \quad (12)$$

Weighted Doubly Robust (WDR) is a variant of DR, which is obtained by stabilized importance sampling weights. The WDR estimator is given as follows:

$$\begin{aligned} WDR(D) := & \frac{1}{m} \sum_{i=1}^m \hat{V}^{\pi_e}(S_0^{H_i}) \\ & + \sum_{i=1}^m \sum_{t=0}^{L-1} \gamma^t w_t(H_i) \left[ R_t^{H_i} - \hat{q}^{\pi_e}(S_t^{H_i}, A_t^{H_i}) + \gamma \hat{V}^{\pi_e}(S_{t+1}^{H_i}) \right] \end{aligned} \quad (13)$$

### 5.3 RLtmle

The RLMLE [16] estimator makes two splits of the trajectories  $D$ :  $D^{(0)} = (H_1, \dots, H_{(1-p)m})$ , and  $D^{(1)} = (H_{(1-p)m+1}, \dots, H_m)$ , for some  $0 < p < 1$ . Use  $D^{(0)}$  to fit estimators the action value functions, called the initial estimators. It defines a parameter model based on the initial estimator fitting, which is called the second phase parameter model  $\widehat{Q}_t^{\pi_e}$ , by fitting this parametric model by maximum likelihood on the split  $D^{(1)}$ , achieves bias reduction.

## 6 Conclusion

In this paper, we consider the existing methods ignore to guide the influence on key states in OPE. When all trajectories have the same initial state, the value of the state value function obtained by OPE is the value of the initial state. But the initial state does not completely represent the real value of all states, especially the key states. we develop a configurable OPE with key state-based bias constraints. FP-Growth from data mining field is adopted to mine the key states and get corresponding reward expectations of key states. We implementation the KSBC estimator construct new goal function with the combination of bias and variance and realize a guided importance sampling-based OPE. Taking the GridWorld game as our experiment platform, we evaluate our method with performance analysis and case studies. Which gets the best of both worlds: it can have a much lower MSE than the popular off-policy evaluation estimator. it can select the one that performs well in key states. In various degrees of model misspecification and sample size, we evaluate our method, as well as make comparisons with mainstream methods to show the effectiveness.

## References

1. Murphy, S.A., van der Laan, M.J., Robins, J.M.: Marginal mean models for dynamic regimes. *J. Am. Stat. Assoc.* **96**(456), 1410–1423 (2001)
2. Petersen, M., Schwan, J., Gruber, S., Blaser, N., Schomaker, M., van der Lan, M.: Targeted maximum likelihood estimation for dynamic and static longitudinal marginal structural working models. *J. Causal Inference* **2**(2), 147–185 (2014)
3. Theocharous, G., Thomas, P.S., Ghavamzadeh, M.: Personalized ad recommendation systems for life-time value optimization with guarantees. In: *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI 2015*, pp. 1806–1812. AAAI Press (2015)
4. Hoiles, W., Van Der Schaar, M.: Bounded off-policy evaluation with missing data for course recommendation and curriculum design. In: *Proceedings of the 33<sup>rd</sup> International Conference on International Conference on Machine Learning – Vol. 48, ICML2016*, pp. 1596–1604. JMLR.org (2016)
5. Jiang, N., Li, L.: Doubly Robust Off-policy Value Evaluation for Reinforcement Learning. arXiv e-prints, art. [arXiv:1511.03722](https://arxiv.org/abs/1511.03722), (2015)
6. Mannor, S., Simester, D., Sun, P., Tsitsiklis, J.N.: Bias and variance approximation in value function estimates. *Manage. Sci.* **53**(2), 308–322 (2007). <https://doi.org/10.1287/mnsc.1060.0614>

7. Rosenbaum, P.R., Rubin, D.B.: The central role of the propensity score in observational studies for causal effects. *Biometrika* **70**(1), 41–55 (1983). <https://doi.org/10.1093/biomet/70.1.41>
8. Precup, D.: Temporal abstraction in reinforcement learning. PhD thesis, University of Massachusetts Amherst, 2000. <https://scholarworks.umass.edu/dissertations/AAI9978540>
9. Precup, D., Sutton, R.S., Singh, S.P.: Eligibility traces for off-policy policy evaluation. In: Proceedings of the Seventeenth International Conference on Machine Learning, ICML 2000, pp. 759–766, San Francisco, CA, USA (2000). ISBN 1-55860-707-2
10. Farajtabar, M., Chow, Y., Ghavamzadeh, M.: More robust doubly robust off-policy evaluation. CoRR, abs/1802.03493, 2018
11. Thomas, P., Brunskill, E.: Data-efficient off-policy policy evaluation for reinforcement learning. In: Balcan, M.F., Weinberger, K.Q., (eds.), Proceedings of The 33rd International Conference on Machine Learning, vol. 48 of Proceedings of Machine Learning Research, pp. 2139–2148, New York, USA (2016.)
12. Thomas, Philip S., Theodoropoulos, G., Ghavamzadeh, M.: High confidence off-policy evaluation. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), (2015)
13. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New York (2014)
14. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Chen, W., Naughton, J., Bernstein, P.A., editors, 2000 ACM SIGMOD International Conference on Management of Data, pp. 1–12. ACM Press (2000)
15. Hanna, J.P., Niekum, S., Stone, P., et al.: Importance sampling policy evaluation with an estimated behavior policy[C]. In: International Conference on Machine Learning, pp. 2605–2613 (2019)
16. Bibaut, AF., Malenica, I., Vlassis, N., et al.: More efficient off-policy evaluation through regularized targeted learning.[C]. In: International Conference on Machine Learning, pp. 654–663 (2019)



# An Efficient Framework for Text Document Security and Privacy

Umair Khadam<sup>1</sup>, Muhammad Munwar Iqbal<sup>1</sup>, Leonardo Mostarda<sup>2</sup>,  
and Farhan Ullah<sup>3</sup>(✉)

<sup>1</sup> Department of Computer Science, University of Engineering and Technology,  
Taxila, Pakistan

umair\_khadim@live.com, munwariq@gmail.com

<sup>2</sup> Computer Science Department, Camerino University, 62032 Camerino, Italy  
leonardo.mostarda@unicam.it

<sup>3</sup> College of Computer Science, Sichuan University, Chengdu 610065, China  
farhankhan.cs@yahoo.com

**Abstract.** Nowadays, with the help of advanced technologies, an illegal copy of digital content can be shared easily. Which rise copyright and authentication problems. Digital text documents are generated and shared daily through different internet technologies such as the cloud, etc. The protection of these documents is a challenging task for researchers. In the past, steganography, cryptography, and watermarking techniques have been applied to resolve the copyright problem. However, most of the existing techniques are applicable for only plain text or protecting the document on the local paradigm. In the said perspective, we proposed a new technique to solve the problem of copyright and authentication on local and cloud paradigms. In this paper, we utilize some custom components of MS Word Document for concealing the watermark into a text document. These components are not referred to as the main document and will not modify the content and format. The experimental analysis and results prove that the proposed method improves the watermark capacity, imperceptible, and robust against formatting attacks.

**Keywords:** Steganography · Cryptography · Document security · Copyright protection · Digital watermarking

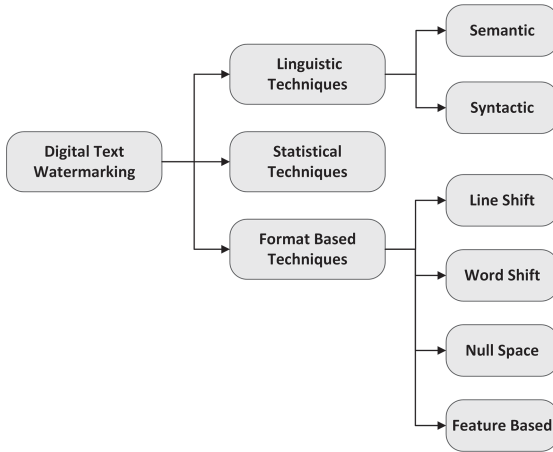
## 1 Introduction

In today's digital world, secure communications are required with rapidly evolving internet technology. Information security has gain importance in many areas like government applications, data storage, e-commerce, e-signature, banking, personal and corporate communication. The purpose of information security is to prevent third parties from accessing information for any purpose [1]. Data breaches are a significant challenge in the modern digital world because the critical data of the organization must be protected against unauthorized access. In the last five years, almost 10 billion records have been lost, exposed, or stolen,

with an average of five million records per day affected. Advanced digital technologies such as the cloud brought unlimited benefits to users, but they also cause problems for the original owner of the data against illegal copies [2]. In the past, steganography, cryptography, and watermarking techniques have been used to provide ownership verification. Digital watermarking plays an important in this field of research. Where, a secret message also called a watermark is embedded into the host content without compromising the data integrity [3,4]. When an illegal act occurs the same watermark, is used for ownership verification. Digital watermarking is classified into audio, video, text, and image, whereas, most of the watermarking research focuses on audio, video, and image [5]. The text watermark has now become very popular and become a hot area of research because text documents are almost part of all private and public sector organizations and need copyrights protection [6,7]. In recent years, cloud computing has been the most significant development in the field of information technology. It provides services to all organizations such as educational institutes, healthcare, banking, etc. via the internet by the pay-as-you-go model [8,9]. The security of data in cloud computing is the main issue for users. It is essential to ensure data security in many positions in data rest. Digital text watermarking is not considered yet in the context of cloud computing. None of the existing digital watermarking technique provides secrecy to a text document in the cloud computing paradigm. We proposed an efficient digital watermarking framework based on MS Word document custom components that protect the text document authentication and verification. Many researchers worked in the field of digital text watermarking, and numerous techniques have been proposed for text document security and privacy. Three major categories of digital text watermarking are statistical, linguistic, and format-based techniques, as shown in Fig. 1. Linguistic based techniques are divided into two significant types semantic and syntactic. In general, the semantic-based technique uses the synonym substitution method, where words synonym is used for embedding the watermark information. In the syntactic techniques the punctuation marks like full stop (.), comma (,), colon (:), and semicolon (;) etc. are placed to conceal the watermark in cover file [10]. In the word and line spacing techniques, the words or lines are shifted up down to some degree to hide secret data.

In the past, several techniques are designed for text documents. Khairullah et al. [11] presented a method based on invisible characters. The proposed technique sets the invisible characters to foreground colors such as the tab, space, or the carriage return characters, which can be obtained 24 bit per character. Similar English Font Types (SEFT) technique is proposed in [12], which utilizes the same English font for text watermarking. First of all, three different fonts are chosen which are identical, and then 26 characters and spaces are represented by a triple of capital letters. The proposed scheme is not considered as robust, because if the spaces between the text are removed then the watermark is ruined. Naqvi et al. [13] introduced a zero text steganography approach that is based on multilayer partially homomorphic. The proposed technique implements multilayer security on a secret message. Kumar et al. [14] suggested a





**Fig. 1.** Digital text watermarking techniques classification.

technique that is based on Huffman compression. The proposed technique uses email forwarding data to conceal watermark and not consider robust. Khosravi et al. [15] proposed an information hiding technique for PDF (Portable Document Format) based on justified text. First, the secret message is compressed by Huffman coding, then some unique lines of PDF files are chosen to conceal the information. The embedding operation takes place by replacing the added spaces with the regular spaces of the host rules. Alghamdi et al. [16] introduced a text steganography technique for the Arabic language. Markov Chain (MC) is implemented for encoder and decoder combined with Huffman Coding. The upper and lower bound are also computed for the stego-text. The proposed technique is format independent and less robust against attacks. Long et al. [17] suggest a coverless method based on web text, where a large number of web pages are used to conceal the secret message. The mature search engines are applied to obtain the secret information that is associated with web pages. Rizzo et al. [18] introduced a structural approach that protects digital content small portions. This approach is suitable for Latin symbols and white spaces which is based on homoglyph character substitution. Hence, the proposed system increases the hiding capacity of watermark but, not robust.

## 2 Proposed Method

A novel framework is proposed here for text document ownership verification and copyright protection based on Microsoft Word (MSW) document custom components, as shown in Fig. 2. Furthermore, this section covers the watermark (secret information) embedding and extraction process. Today, the MSW document is a critical part of all public and private organizations. The copyright protection and ownership verification of these documents are essential. We introduced a novel content free watermarking technique for text document security

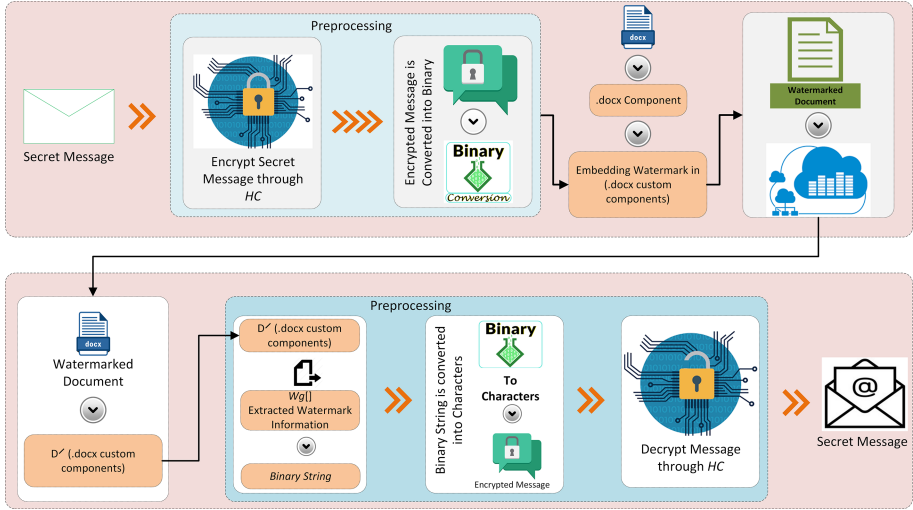


Fig. 2. The proposed model for digital watermarking.

and privacy on local and cloud paradigms. We use the custom components of MSW document for concealing the watermark information. The MSW custom components are appropriate for watermarking because these components are not part of the main document. The process of watermarking does not change the document original contents. The main reason for using these components is that they can mask enough watermark. In addition, the proposed technique is robust against format-based attacks, semantic-based attacks, and content-based attacks. Huffman Coding is used for compression. Secret Message (MS) and original document (DO) are given as input in the proposed model, MS is compressed through Huffman Coding. The compressed message ME is transformed into a binary string then divided into  $n$  groups. The custom components of text document DO are checked, and the groups of ME are inserted into these components. The watermark information does not affect the document's original content and does not interfere with imperceptibility. After hiding the watermark information, the watermarked document is generated and shared via different communication technologies.

## 2.1 Huffman Coding

Huffman Coding is one of the high data compression rate algorithms, which gives the variable-length code to input characters. On the bases of the character's frequencies Huffman tree is constructed, which determines the length of the code. Small codes are given to most frequent characters, and bulky code is assigned to less frequent characters. Huffman tree is responsible for the coding and decoding process from the sequence of characters to bitstream or vice versa. To avoid ambiguity a unique code is assigned to each character that should not be used

with other characters [16]. In Huffman Coding data compression is achieved through binary allocation codewords of different lengths. Let  $W$  belongs to the possible plain text, and  $M = M_1, M_2, M_3 \dots M_n$ , the plain text alphabet of  $P$ , and  $P$  belong  $W$  such that  $P = P_1, P_2$ . where  $P_i$  belongs to  $M$ . If  $W_i$  is the probability of  $P_i$  appearing in the plain text  $P$ , we have the Entropy of  $P$  defined by (1):

$$H(P) = - \sum_{m=1}^{i=1} W_i * \log W_i \quad (1)$$

As the average number of bits to represent each symbol  $M_i$  belongs  $M$ . Moreover,  $H(P)$  leads to zero redundancy, that is, has the exact number of significant bits to represents  $P$ . The encoding produced by Huffman Coding is prefix-free and satisfies through (2):

$$H(P) = 1(HC) < H(P) + 1 \quad (2)$$

Where 1 is the weighted average length.

## 2.2 Watermark Embedding

The MSW document is a common type of text document throughout the world. It comprises a lot of custom components. These components are suitable for watermarking and authorized users to manipulate with it through programming. Three main reasons are why these components are appropriate for watermarking. Firstly, the watermark information is stored in custom components, which cannot affect the contents of the document. Secondly, without affecting the imperceptibility a large amount of watermark information is stored. Thirdly, it is robust, any command of MSW will not interrupt or delete the watermark. The Microsoft Visual Basic (VB) is used to store and retrieve the watermark information from the MSW document. The ME is divided into groups before embedding using (3).

$$W_g = \left\{ \frac{i_w, \{w|w = 1, 2, \dots, n\}}{N_w} \right\} \quad (3)$$

Where  $W_g$  is total groups of watermark information,  $i_w$  is watermark information,  $N_w$  is the number of groups. The groups of watermark information  $W_g$  is dependent on  $Wobj(D)$ .  $W_g[i]$  in embedded into the value attribute of custom objects. When all  $W_g[n]$  is embedded into the DO then  $D_w$  is generated and shared on the cloud.

## 2.3 Watermark Extraction

The objective of extraction is to extract the MS and verify the document originality. In our system, the second phase of the proposed model describes the watermark extraction complete procedure, as shown in Fig. 2. The  $D_w$  document is given to the system as input, the list of interrupted components  $D$  are utilized for collecting the groups of watermark information  $W_g[n]$ . The groups of watermark information are concatenated and then converted into a binary string then characters, and finally, the secret message is recovered.

### 3 Experimental Classification Results and Analysis

In this section, the results of our proposed technique are analyzed on the bases of digital watermarking evaluation criteria. Which can be categorized into robustness, capacity, and imperceptibility.

#### 3.1 Robustness Analysis

Robustness is a critical factor in digital watermarks, and it indicates that after applying various attacks, either 100% watermark information is restored or not. Different types of brute force attacks are applied to the watermarked document to verify its robustness. These attacks include attacks based on content and format. Table 1 presents the comparison of the proposed method with [19–21] against content and format-based attacks. The proposed technique is based on MSW custom components and any mutual command cannot disturb the watermark. Table 1 shows that the proposed technique resistant against content and format-based attacks. The comparison demonstrates that the proposed algorithm presents improved results.

**Table 1.** The robustness comparison against content and format-based attacks

Techniques	Insert	Delete	Replace	Size	Copy	Color	Weight	Alignment	Spacing
Zhang et al.[19]	Yes	Yes	Yes	Yes	No	No	No	No	No
Castigli et al.[20]	Yes	Yes	Yes	Yes	No	Yes	Yes	No	No
Liu et al.[21]	No	No	No	Yes	No	Yes	Yes	Yes	No
Proposed Method	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

#### 3.2 Capacity Analysis

Hiding Capacity is one of the significant parameters that measure the watermarking algorithm’s strength. The capacity specifies the maximum number of bits called a secret message can be stored in the original text. The analysis of the existing technique summarized that an efficient system is required that maximize the hiding capacity, without affecting the original content of the text and conflicting other parameters. Equation (4) can be used to measure the hiding capacity of the proposed system.

$$HC = \frac{\text{Secret information (bits)}}{\text{Size of cover file (Kb)}} \quad (4)$$

In this experiment, we select 50 different text documents with different sizes. When we compared the proposed algorithm with [6] and [7], it improves the hiding capacity dramatically, as shown in Fig. 3.

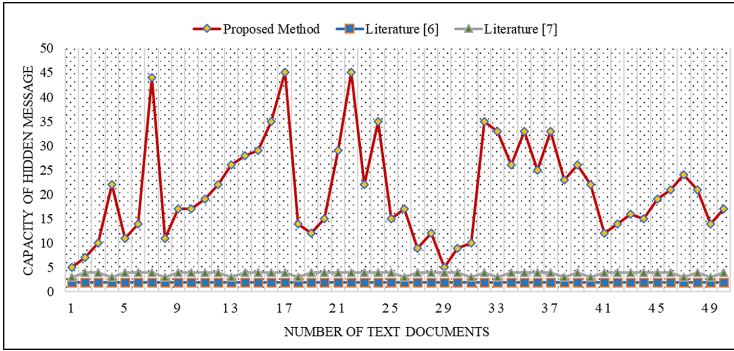


Fig. 3. The comparison of capacity analysis.

### 3.3 Imperceptibility Test

Imperceptibility defines as the watermark information that will not alter the original content and cannot be seen through human eyes. Only the authorized persons can extract the watermark through special processing or dedicated circuits. We use 15 different strings to measure the imperceptibility, the former technique differs from 0.83 to 0.97, but the average similarity of the proposed system is 1 as shown in Fig. 4. As mentioned above, we use the custom components to embed the watermark, so the watermark does not affect the original content that’s why our scheme has 100% results on imperceptibility.

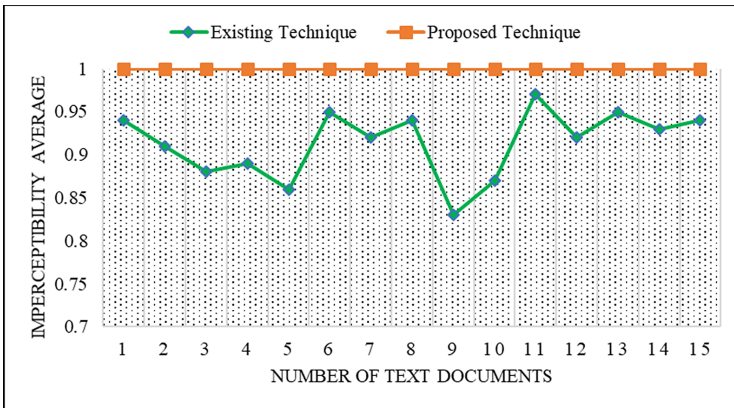


Fig. 4. The comparison of imperceptibility test with the previous technique [6].

As demonstrated in the experimental results, our proposed technique achieves excellent results against robustness, capacity, and imperceptibility. The proposed method is robust against all formatting attacks and more secure compared to

the previous techniques. The proposed technique uses the custom components of the MS Word document that cannot be referred to as the main document and will not modify the content and format. Our system can be applied for text documents authentication and copyright protection. It can also protect text documents from illegal use.

## 4 Conclusion

In this investigation, we proposed a content free watermarking technique that is based on Microsoft Word document custom components. These custom components are not referred to in the main document. Therefore, no changes were made to the content and format of the original document when we embed the watermark information. The experimental results and analysis prove that the proposed technique is robust against attacks based on content and formatting, imperceptible, and improves the capacity as compared to the previous techniques. The watermark information can be extracted with high probability after applying various formatting attacks. In the future, Microsoft Word and Excel documents, other properties will be examining for watermarking. Moreover, we also investigated the Portable Document Format (PDF) document that is the most popular document format in the world. The handwritten text, fingerprints, and manual signatures can also be taken as a watermark.

## References

1. Yesilyurt, M., Yalman, Y.: New approach for ensuring cloud computing security: using data hiding methods. *Sādhanā* **41**(11), 1289–1298 (2016). <https://doi.org/10.1007/s12046-016-0558-8>
2. Khadam, U., et al.: Text data security and privacy in the internet of things: threats, challenges, and future directions. *Wireless Commun. Mobile Comput.* (2020). <https://doi.org/10.1155/2020/7105625>
3. Khadam, U., et al.: Digital watermarking technique for text document protection using data mining analysis. *IEEE Access* **7**, 64955–64965 (2019). <https://doi.org/10.1109/ACCESS.2019.2916674>
4. Saba, T., et al.: Enhancing fragility of zero-based text watermarking utilizing effective characters list. *Multimed. Tools Appl.* **79**(1), 341–354 (2019). <https://doi.org/10.1007/s11042-019-08084-0>
5. Naz, F., et al.: Watermarking as a service (WaaS) with anonymity. *Multimed. Tools Appl.* **79**(23), 16051–16075 (2019). <https://doi.org/10.1007/s11042-018-7074-2>
6. Chen, X., Sun, H., Tobe, Y., Zhou, Z., Sun, X.: Coverless information hiding method based on the chinese mathematical expression. In: Huang, Z., Sun, X., Luo, J., Wang, J. (eds.) *ICCCS 2015*. LNCS, vol. 9483, pp. 133–143. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-27051-7\\_12](https://doi.org/10.1007/978-3-319-27051-7_12)
7. Zhou, Z., et al.: Coverless multi-keywords information hiding method based on text. *Int. J. Secur. Appl.* **10**(9), 309–320 (2016). <https://doi.org/10.14257/ijssia.2016.10.9.30>

8. AlKhamese, A.Y., Shabana, W.R., Hanafy, I.M.: Data security in cloud computing using steganography: a review. In: 2019 International Conference on Innovative Trends in Computer Engineering (ITCE). IEEE (2019). <https://doi.org/10.1109/ITCE.2019.8646434>
9. Wahsheh, H.A., Luccio, F.L.: Security and privacy of QR code applications: a comprehensive study, general guidelines and solutions. *Information* **11**(4), 217 (2020). <https://doi.org/10.3390/info11040217>
10. Mir, N., Khan, M.A.: Copyright protection for online text information: using watermarking and cryptography. In: 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS). IEEE (2020). <https://doi.org/10.1109/ICCAIS48893.2020.9096817>
11. Khairullah, M.: A novel text steganography system using font color of the invisible characters in microsoft word documents. In: 2009 Second International Conference on Computer and Electrical Engineering. IEEE (2009). <https://doi.org/10.1109/ICCEE.2009.127>
12. Bhaya, W., Rahma, A.M., Al-Nasrawi, D.: Text steganography based on font type in MS-Word documents. (2013). <https://doi.org/10.3844/jcssp.2013.898.904>
13. Naqvi, N., Abbasi, A.T., Hussain, R., Khan, M.A., Ahmad, B.: Multilayer Partially Homomorphic Encryption Text Steganography (MLPHE-TS): A Zero Steganography Approach. *Wireless Personal Commun.* **103**(2), 1563–1585 (2018). <https://doi.org/10.1007/s11277-018-5868-1>
14. Kumar, R., et al.: A high capacity email based text steganography scheme using Huffman compression. In: 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN). IEEE (2016). <https://doi.org/10.1109/SPIN.2016.7566661>
15. Khosravi, B., et al.: A new method for pdf steganography in justified texts. *J. Inf. Secur. Appl.* **45**, 61–70 (2019). <https://doi.org/10.1016/j.jisa.2019.01.003>
16. Alghamdi, N., Berriche, L.: Capacity investigation of markov chain-based statistical text steganography: arabic language case. In: Proceedings of the 2019 Asia Pacific Information Technology Conference. ACM (2019). <https://doi.org/10.1145/3314527.3314532>
17. Long, Y., et al.: Coverless information hiding method based on web text. *IEEE Access* **7**, 31926–31933 (2019). <https://doi.org/10.1109/ACCESS.2019.2901260>
18. Rizzo, S.G., Bertini, F., Montesi, D.: Fine-grain watermarking for intellectual property protection. *EURASIP J. Inf. Secur.* **2019**(1), 1–20 (2019). <https://doi.org/10.1186/s13635-019-0094-2>
19. Zhang, J., Xie, Y., Shen, J., Wang, L., Lin, H.: Text information hiding method using the custom components. In: Sun, X., Pan, Z., Bertino, E. (eds.) ICCCS 2018. LNCS, vol. 11066, pp. 473–484. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-00015-8\\_41](https://doi.org/10.1007/978-3-030-00015-8_41)
20. Castiglione, A., De Santis, A., Soriente, C.: Taking advantages of a disadvantage: digital forensics and steganography using document metadata. *J. Syst. Software* **80**(5), 750–764 (2007). <https://doi.org/10.1016/j.jss.2006.07.006>
21. Liu, T.-Y., Tsai, W.-H.: A new steganographic method for data hiding in microsoft word documents by a change tracking technique. *IEEE Trans. Inf. Forensics Secur.* **2**(1), 24–30 (2007). <https://doi.org/10.1109/TIFS.2006.890310>

# **Social Networks**





# A Real-Time Audio and Video Streaming Transmission Scheme for Social Media

Jianping Yu<sup>1,3</sup>, Gang Zhao<sup>2</sup>, Xiaohui Kuang<sup>2</sup>, and Ruyun Zhang<sup>4</sup>(✉)

<sup>1</sup> School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China  
ppzepp@163.com

<sup>2</sup> National Key Laboratory of Science and Technology on Information System Security, Beijing 100093, China  
zhao-gang20@126.com, xiaohui\_kuang@163.com

<sup>3</sup> Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou 510006, China

<sup>4</sup> Zhejiang Lab, Hangzhou, Zhejiang, China  
zhangry@zhejianglab.com

**Abstract.** Real-time video streaming transmission has been widely used in security monitoring field. However, audio function in the monitoring system has not attracted significant attention. As a well-known streaming media server in security monitoring field, Live555 projects have realized the real-time video capture and transmission with the secondary development, but they still lack the collection and transmission of real-time audio on IP camera. In this paper, we propose a streaming media transmission scheme to realize simultaneous transmission of real-time audio and video based on the Live555 project. In order to realize the above scheme, we add the real-time audio collection module, rewrite the related classes and methods in the project to transport real-time audio data with video. It is demonstrated by experiments that the real-time audio and video simultaneous transmission is stable, the delay is low and the quality is good. It can be played on mainstream players such as VLC and FFplay.

**Keywords:** Live555 real-time transmission · Audio and video · IP camera

## 1 Introduction

With the improvement of Internet of Things technology and people's concern about home security ecology, IP camera enters the field of home security from the professional field [1]. The new application scene brings new challenges. Most of IP cameras only support real-time video transmission, while in home, private and quiet environment requires IP camera can transport audio with video at the same time.

Live555 streaming media server plays an important role in security monitoring field [2]. However, the official source code only supports file streaming transmission rather than real-time audio and video transmission. Though there are already some secondary

development Live555 projects to implement real-time video collection and transmission, audio is still a file transmission framework [3, 4].

In our paper, we propose a real-time audio and video streaming media transmission scheme for social media. Live555 project is based on the secondary development version which is in the SDK of Ambarella S2Lm chip. This version has achieved real-time video capture and transmission without real-time audio. So we add the real-time audio collection module to get real-time audio data, rewrite classes and methods related to audio to realize the real-time audio transmission, then add audio subsession to `ServerMediaSession` to merge audio with video. After cross-compiling and transplanting it to IP camera, we achieve normal real-time audio and video forwarding play. To summarize, we make the following main contributions:

- Real-time audio collection module is introduced to get real-time audio data.
- Classes and methods related to audio in Live555 are rewritten to realize the real-time audio transmission.

## 2 Related Work

The IP camera used in this paper has S2Lm processing chip produced by Ambarella [5], and is based on Live555 streaming media server, collects and transports H264 video and PCM audio in time. Live555 has a streamlined architecture and good portability, so it is easy to be used on multiple platforms through cross compilation, especially embedded systems [6].

However, the official source code of Live555 only supports file streaming transmission rather than real-time audio and video transmission. At present, there are two main solutions for Live555 real-time transmission. One is to use named pipe, the other is inheriting related classes and rewriting related methods. For the first method, it has been realized the real-time video transmission with named pipe [4]: use the `mkfifo` command to create a named pipe, then run the program, so that the collected real-time stream is continuously written into this FIFO [7], the Live555 server can run directly to see the real-time video. However, when the bitrate is large, the real-time video playing will have obvious jams and mosaics.

Therefore, the second method is commonly used, which is also the method we reference. By rewriting the relevant methods, the reading of the audio and video data is changed from files to memory, which avoids the overhead of reading and writing files by FIFO, Lu Shaojun [3] and others have initially implemented a real-time H264 streaming media transmission system by adding classes to LiveMedia library. The problems of delay and unstable data transmission in the Live555-based video transmission system has been solved [8], and the video transmission is smooth and stable. But including the researches mentioned above, most of the secondary development is done only for the streaming of h264 video data, they lack the attention on audio. Therefore, the goal of this article is to add the audio function of Live555, and to achieve the integration and simultaneous playback of audio and video.

### 3 Design and Implementations

The Live555 in Ambarella SDK has realized real-time video data collection and transmission. The BasicUsageEnvironment and BasicTaskScheduler are recreated separately in setup\_streams. The real-time collection thread is added to store video data in a circular array, then enter doEventLoop to loop and wait for new client. When a client connects, RTSPClientSession class is created to process the client request [9]. A new subthread will be created in the process of interacting with client. In this subthread, enter doEventLoop to send real-time data with the independent BasicTaskScheduler object. The flow chart of real-time data is shown as Fig. 1:

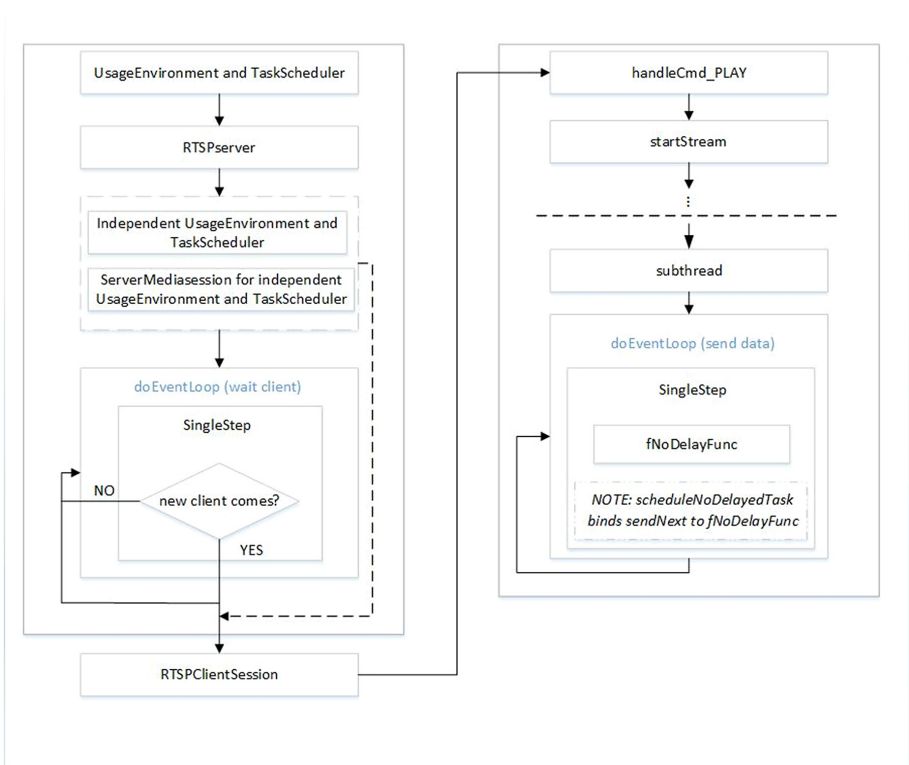


Fig. 1. Flow chart of real-time transmission in Live555.

#### 3.1 Real-Time Audio Collection and Preparation

We use ALSA framework to achieve the collection of PCM audio [10], rewrite the WAVAudioFileSource and WAVAudioFileServerMediaSubsession to achieve the preparation of real-time audio. Key classes for audio are as shown in Table 1:

**Table 1.** Key classes for audio in LiveMedia

Class	Main functions
RTSPServer	Build RTSP server, create RTSPClientSession to handle individual client sessions
RTSPClient	Handle RTSP requests and responses, create RTP sessions
WAVAudioFileSource	Get PCM format audio data from buffers
SimpleRTPSink	Save and sent audio data to client
WAVAudioFileServerMediaSubsession	Save information of the streaming media, connect WAVAudioFileSource and SimpleRTPSink

**Collection of Audio.** We set audio format as SND\_PCM\_FORMAT\_S16\_LE, sample rate as 16000, the number of sample channels as 1. Then follow the ALSA audio acquisition process [11]. Formula for calculating the size of audio frame is as follows:

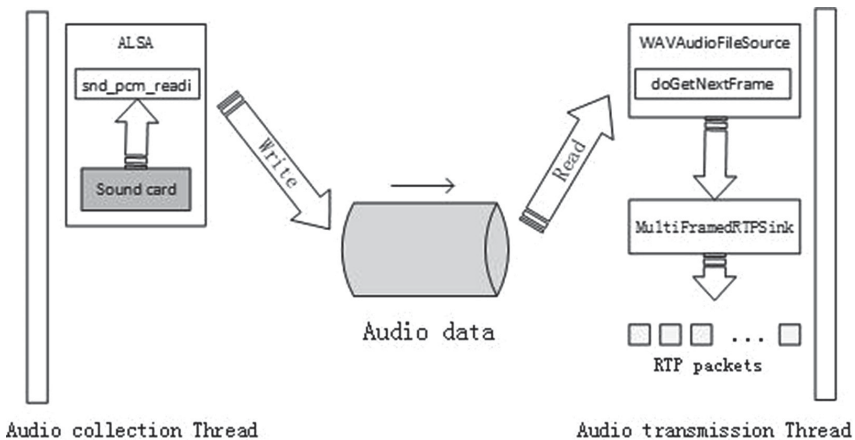
$$\text{FrameSize} = \text{sizeof}(\text{one sample}) * \text{nChannels} \tag{1}$$

So one frame occupies two bytes. We read `chunk_size` frames from the sound card one time, then store the audio data in buffer array `buf_in`, so we can read audio data from buffer other than sound card. The size of `buf_in` is calculated as follows:

$$\text{BufferSize} = \text{sizeof}(\text{FrameSize}) * \text{chunk\_size} \tag{2}$$

**Preparation of Audio.** PCM audio does not need to be encoded. So audio data stored in `buf_in` is read into `fTo` directly to wait to be consumed by Sink.

We assign empty data of the same length to `fTo` when the audio collection speed is lower than the reading speed. We also need to set the corresponding `fFrameSize` and calculate the corresponding playing time `fDurationInMicroseconds`. The relationship between audio capture thread and transport thread is as shown in Fig. 2:



**Fig. 2.** The relationship between audio capture thread and transport thread.

### 3.2 Real-Time Audio Consumption

Consumption of real-time data is in the multiFramedRTPSink class actually. The process of packaging and sending is as follows:

In continuePlaying method, scheduleNoDelayedTask ((TaskFunc \*) sendNext, this) use sendNext as the callback function to achieve the send task without delay. The live555 project on the S2Lm chip no longer uses the delay queue to send audio and video data. The scheduleDelayedTask method is no longer used, but the function ScheduleNoDelayedTask is recreated. The sendNext method calls BulidAndSendPacket to prepare the RTP header [12], and BulidAndSendPacket uses PackFrame to frame, PackFrame calls getNextFrame to continuously obtain data from Source, and sendPacketIfNecessary sends the data packet to the player. So far, the package is sent completely this time. Exit the singleStep function, enter doEventLoop and wait for the next packet sending process. This can effectively improve the efficiency of the server to send data, so as to obtain higher real-time and processing efficiency. This flow chart is as shown in Fig. 3:

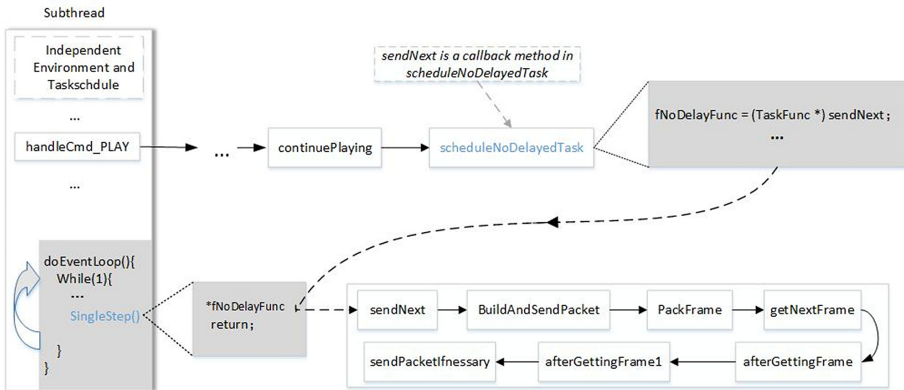


Fig. 3. Packaging and sending flowchart.

Create WAVAudioFileSource and SimpleRTPSink in the WAVAudioFileServerMediaSubsession class by implementing createNewStreamSource and createNewRTPSink, and set the corresponding audio parameters. Then add audio subsession into ServerMediaSession.

### 3.3 Redesign of Delayed Task Processing

Instead of using scheduleDelayedTask, ScheduleNoDelayedTask is called in continuePlaying method to deal with delayed tasks. ScheduleNoDelayedTask can deal task without delay. In order to have a deeper understanding of ScheduleNoDelayedTask, we compare the difference between ScheduleNoDelayedTask and delay queue [13].

The official live555 is a single-process, single-threaded server, but it can perfectly allow multiple clients to connect at the same time, delay queue is an important means [2]. ScheduleDelayedTask is called to add the transmitting work to the delay queue. In

singleStep, we need to actively go to the delay queue to check whether there is a timeout task, then execute the task of the timeout node, delete and synchronize the remaining time of the node in the queue [14].

This project improves to achieve multi-threaded concurrency. Each task thread creates independent TaskScheduler and UsageEnvironment classes, then executive doEventLoop independently to realize data processing. So the collected audio and video data can be continuously transmitted, and it is no longer necessary to call scheduleDelayedTask to add the sending work to the delay queue. Therefore, scheduleNoDelayedTask is introduced in the BasicTaskScheduler0 class to realize the transmission of real-time data immediately.

Because of the addition of audio stream, the data processing flow need to be refined. In the BasicUsageEnvironment0 class, there was originally only one pair of TaskFunc \* fNoDelayFunc and void \* fNoDelayClientData. This pair of member variables corresponds to the two parameters of scheduleNoDelayedTask, represents the nodelay task. If there is still only one pair of fNoDelayFunc and fNoDelayClientData, the latter caller will cover previous one. Therefore, add a new pair of member variables fNoDelayFunc2 and fNoDelayClientData2 to distinguish between audio and video; scheduleNoDelayedTask and singleStep in the BasicTaskScheduler class should handle the corresponding audio and video streaming tasks and data separately.

During the entire rtsp server operation process, multiple pairs of TaskScheduler and UsageEnvironment classes were created: one pair is mainly used to receive client requests and establish a connection with the client, the other pair is used to independently process audio and video data transmission. After one client connected, the main thread still calls select method to wait for the connection of new client, but it does not deal with the sending of the real-time data anymore—this task is processed in the subthreads created by audio and video subsessions, they send the real-time data in doEventLoop without any delay, and no longer handle the client's connection task. In fact, because we deal with the real-time data, unlike audio and video files, we no longer need to deal with operations such as pauses and fast forwards sent by the client, which can greatly improve the efficiency of the live555 server.

### 3.4 Thread Priority Setting

When creating the real-time transmission thread of audio, we noticed that the real-time transmission thread of video has a priority setting. In general, there are three kernel scheduling strategies in the Linux development environment [15]: SCHED\_OTHER, SCHED\_RR, SCHED\_FIFO. The default priority of thread is SCHED\_OTHER, which is allowed to be preempted by real-time tasks; SCHED\_RR level threads are based on time slice rotation. When the time slice of a thread is reduced to 0, it will actively give up the CPU; For SCHED\_FIFO level thread, once the CPU is occupied, it will continue to run until there is a higher priority task arrive or give up CPU on their own, so it will cause thread starvation [16].

The audio transmission thread we created here is a real-time thread, and it is not suitable for scheduling based on time slices—every time you send data, you must send all the data over in this time. You cannot give up data transmission just because the time slice is used up. Therefore, we set the audio transmission thread to the SCHED\_FIFO

level with the same priority as the video transmission thread [17]. This will cause a problem that if the video transmission thread is called first, the video transmission thread will always occupy CPU. Therefore, we need to make the video transmission thread voluntarily give up CPU. Here we add a mutex to SingleStep. If the video thread gets the right to use the CPU first, after it executes the SingleStep method to release the lock, when it waits for the mutex resource in the next round, it will voluntarily give up the CPU use right, the thread task is deleted from the ready queue, and join the wait Queue; the next ready thread—the audio transmission thread will get the right to use the CPU, thereby achieving the concurrency of audio and video threads.

## 4 Evaluation

Use arm-linux-gnueabi to cross compile the live555 project, transplant the executable file to the camera based on Ambarella platform, execute the script to start the service. The hardware parameters of IP camera are as shown in Table 2:

**Table 2.** hardware parameters of IP Camera

Nominal performance	Parameters
Hardware	Ambarella S2L
Operating system	Linux version 3.10.73
Crosstool	Ambarella Linaro Multilib GCC
CPU	ARMv7 Processor rev 1 (v7l)
RAM	103 MB

Enter the RTSP protocol playback address on the VLC client: rtsp://192.168.43.138/stream1, it can be played normally on player and audio codec information is as shown in Table 3.

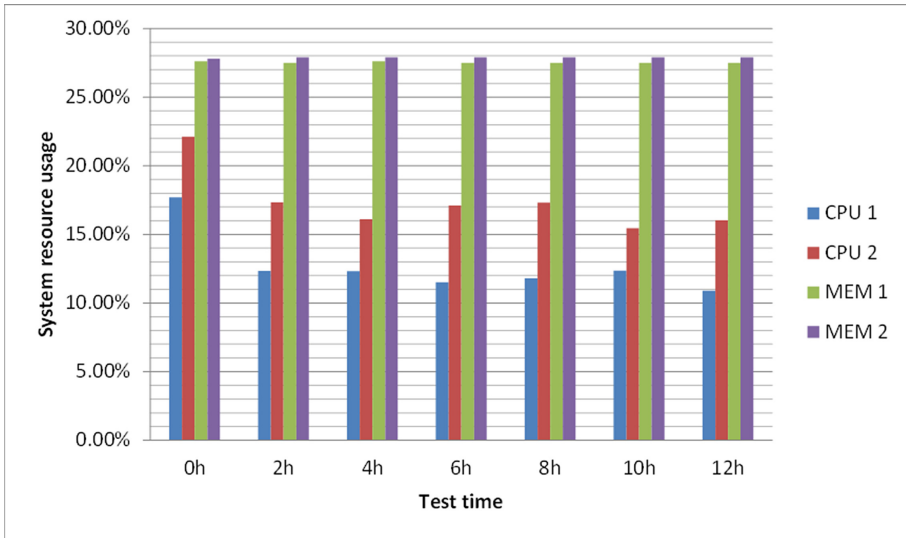
**Table 3.** Media codec information on VLC

Audio codec	Parameters
Codec	PCM S16 BE (s16b)
Channels	Mono
Sample rate	16000 Hz
Bits per sample	16

We mainly evaluate this project from two aspects: system performance and real-time performance.

## 4.1 System Performance

For system performance, test CPU and memory usage of the camera during twelve hours. We first test the system resource occupation of the camera when only real-time video is captured and transmitted. Then add real-time audio to test the occupancy of system resources when audio and video are transmitted simultaneously, the result is as shown in Fig. 4:



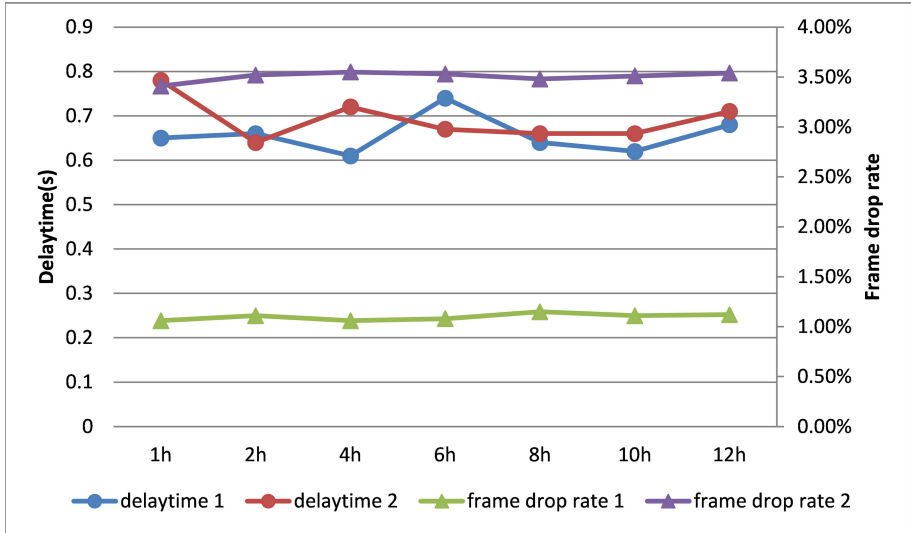
**Fig. 4.** System resource occupancy. CPU 1 means CPU usage when only video thread is working. CPU 2 means CPU usage when audio and video thread are working together. It is also suitable for MEM.

It can be seen from the above results that added audio collection and transmission threads can increase CPU occupancy rate, but even if the real-time audio and video collection and transmission threads are called at the same time, CPU resource utilization rate can be stabilized at about 20%; the CPU occupancy rate will not fluctuate greatly with time, indicating the thread are relatively stable. The proportion of memory space occupied by added audio thread is very small, only occupying about 0.2% more memory space; the memory occupancy rate is flat with time, which means that there is no memory leak in our project.

## 4.2 Real-Time Performance

For real-time testing, we use a stopwatch timer to intuitively test the delay of the Live555 project. We define the delay as the difference between the stopwatch time on the computer and the stopwatch time displayed on the player. In our LAN environment, the network download bandwidth can reach 5 MB/s and upload bandwidth can reach 1.4 MB/s; when





**Fig. 5.** Real-time performance. Frame drop rate 1 means frame drop rate when only video thread is working. Frame drop rate 2 means frame drop rate when audio and video thread are working together. It is also suitable for delaytime.

playing PCM audio and 1080p video, set the video code rate to 1200 kb/s. The frame drop rate and delay is as shown as Fig. 5:

From the results from above figure, delay can be controlled at about 0.6 s, real-time performance is good. Though frame drop rate has increased when the audio collection and transmission threads are added, the playback effect still has a good performance.

## 5 Conclusion

Most of the traditional real-time transformation schemes based on live555 project are only about video, there are few studies on the collection and transmission of real-time audio. In this paper, an audio collection module is added to get real-time audio data in Live555. Related classes are inherited and related methods are rewritten to realize the transmission of real-time audio and the synchronization of audio and video. We can use VLC player to play this stream in LAN environment. Besides the low delay, this Live555 can maintain long-term stable work. Experiments show that the real-time playback effect is good and can be used normally.

RTMP protocol is more widely used in the live broadcast industry [18, 19]. In future, we will build a cloud platform in the public network [20], try to convert RTSP protocol to RTMP protocol in the camera and push RTMP data to Public network to achieve forwarding.

**Acknowledgment.** This work was sponsored by Zhejiang Lab (NO. 2020LE0AB02).

## References

1. Muruti, G., Rahim, F.A., Zawawi, N.A.: Motion activated security camera using raspberry Pi: an IoT solution for room security. *Adv. Sci. Lett.* **24**(3), 1698–1701 (2018)
2. LIVE555. <http://live555.com/>. Accessed 12 Apr 2020
3. Lv, S.-J., Zhou, Y.-P.: Real-time streaming media transmission system based on Live555. *Jisuanji Xitong Yingyong – Comput. Syst. Appl.* **24**(1), 56–59 (2015)
4. Huang, J., Yin, H.: An embedded multifunctional media system for mobile devices in terrestrial DTV relaying. *J. Inf. Process. Syst.* **14**(5), 1272–1285 (2018)
5. Ambarella. <https://www.ambarella.com/>. Accessed 25 Aug 2019
6. Huan, S., Li, P., Hao, W., Xuefang, Z.: Design of H.264 hard codec video transmission system based on ARM11. *Comput. Meas. Control* (2018)
7. Qian, J., Srisa-An, W., Seth, S., Hong, J., Pan, Y.: Exploiting FIFO scheduler to improve parallel garbage collection performance. *ACM SIGPLAN Not.* **51**(7), 109–121 (2016)
8. Wei, C.Y., Zhang, H.L.: Design and implementation of mobile real-time broadcast system based on Live555. *Comput. Eng. Des.* (2016)
9. Hao, F., Li, A., Liu, Y.: Design of embedded video monitoring system based on DM6437. *Adv. Mater. Res.* **1003**(01), 249–253 (2014)
10. ALSA. <https://www.alsa.com/en/web/bus/home>. Accessed 1 Sept 2019
11. Li, L., Mingjiang, W., Boya, Z., Anli, Y.: Design and implementation of embedded WM8960 audio driver and multi-thread player. In: *The Institute of Electrical and Electronics Engineers, Inc. (IEEE) Conference Proceedings*, pp. 292–295 (2015)
12. Zhang, X.-L., Zhou, Y.-P., Deng, C.-M.: Streaming media server based on android. *Jisuanji Xitong Yingyong - Comput. Syst. Appl.* **22**(12), 180–183 (2013)
13. Shibeshi, Z.S., Terzoli, A., Bradshaw, K.: An RTSP proxy for implementing the IPTV media function using a streaming server. *Informatica* **36**(1), 37–45 (2012)
14. Wei, C., Zhang, H.: Applications of a streaming video server in a mobile phone live streaming system. *J. Softw. Eng. Appl.* **7**(12), 975–982 (2014)
15. Kang, D., Lee, W., Park, C.: Kernel thread scheduling in real-time Linux for wearable computers. *ETRI J.* **29**(3), 270–280 (2007)
16. Choi, E., Jang, M., Kim, B.: Improving and evaluating real-time performance for smart devices. *Int. Inf. Inst. (Tokyo) Inf.* **16**(8), 5733–5743 (2013)
17. Hart, D., Stultz, J., Ts'o, T.: Real-time Linux in real time. *IBM Syst. J.* **47**(2), 207–220 (2008)
18. Aloman, A., Ispas, A.I., Ciotirnae, P., Sanchez-Iborra, R., Cano, M.D.: Performance evaluation of video streaming using MPEG DASH, RTSP, and RTMP in Mobile Networks. In: *The Institute of Electrical and Electronics Engineers, Inc. (IEEE) Conference Proceedings*, pp. 144–151 (2015)
19. Xue, Y., Tan, Y.-A., Liang, C., Li, Y., Zheng, J., Zhang, Q.: RootAgency: a digital signature-based root privilege management agency for cloud terminal devices. *Inf. Sci.* **444**, 36–50 (2018)
20. Zhang, Q., Gong, H., Zhang, X., Liang, C., Tan, Y.-A.: A sensitive network jitter measurement for covert timing channels over interactive traffic. *Multimed. Tools Appl.* **78**(3), 3493–3509 (2019)



# UAV-Enabled Social Internet of Vehicles: Roles, Security Issues and Use Cases

Chaogang Tang<sup>1</sup>, Xianglin Wei<sup>2</sup>(✉), Chong Liu<sup>3</sup>, Haifeng Jiang<sup>1</sup>,  
Huaming Wu<sup>4</sup>, and Qing Li<sup>5</sup>

<sup>1</sup> School of Computer Science and Technology,  
China University of Mining and Technology, Xuzhou 221116, China  
{cgtang, jhfeng}@cumt.edu.cn

<sup>2</sup> National University of Defense Technology, Changsha 410073, China  
wei\_xianglin@163.com

<sup>3</sup> The School of Engineering and Applied Science,  
The George Washington University, Washington, DC 20052, USA  
cliu15@gwu.edu

<sup>4</sup> Center for Applied Mathematics, Tianjin University, Tianjin 300072, China  
whming@tju.edu.cn

<sup>5</sup> Department of Computing, The Hong Kong Polytechnic University,  
Hong Kong, China  
csqli@comp.polyu.edu.hk

**Abstract.** Social internet of vehicle (SIOV), also termed vehicular social network (VSN), endeavors to integrate social networking related concepts into IoV, with an aim to make vehicles capable of social communication and low-cost infotainment service provisioning. In spite of potential prospects, some issues pertaining to SIOV remain to be addressed such as security and privacy. Specifically, we in this paper propose an Unmanned Aerial Vehicles (UAVs) enabled security framework to protect the security and privacy of SIOV. On one hand, we split the evolvement of SIOV into two phases and elaborate the roles and functionalities of vehicles in each stage; on the other hand, owing to high flexibility, fast deployment, and low-cost maintainability, we incorporate UAVs into SIOV with the purpose of accomplishing multiple functions including communication range extension, data processing improvement and security protection. Use cases are also given in hope to provide some insights within UAV enabled SIOV.

**Keywords:** Social internet of vehicle · Security · Vehicular social network · UAV · Privacy preservation

## 1 Introduction

Internet of Thing (IoT) that benefits from the development of information and communication technology (ICT) has gained widespread attention in both academia and industry, with the purpose of making everyday objects connected

to Internet and interactive to each other. IoT has a wide range of applications (e.g., Industrial Internet of thing (IIoT) and Internet of Vehicle (IoV)) [1, 2]. It further lays the foundation for smart cities [3]. For instance, as a subecosystem of IoT, IoV envisions a scenario where vehicles are the Internet enabled objects. Besides, it integrates internal vehicle network, inter-vehicle network and vehicle-mounted mobile Internet. In-car sensors are used for perceiving information related to vehicular state and the surroundings. Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication technologies are needed for data delivery and information dissemination while vehicle loaded computer system enables on-site data processing and analysis. All these technologies are intended for intelligent traffic management, vehicular service provisioning, and smart vehicle controls.

Against this background, a new paradigm, named social internet of vehicle (SIOV) [4], also termed vehicular social network (VSN) [5], has been ushered in, with an aim to make vehicles capable of social communication and low-cost infotainment service provisioning, by integrating social networking related concepts into IoV [6]. VSN is stimulated by the fact that people have instinctive desires to socialize with each other, even if they are vehicle travelers on the road. To guarantee the driving safety and enjoy the driving pleasure, popular social networking software are expected to integrate into smart vehicles in SIOV, and thus people can use voice control commands to socialize with each other. Several promising technologies are pushing the social interactions on the road to shift from the smart phone centric way to the smart vehicle centric way. For example, nature language understanding (BLU), deep learning (DL), text-to-speech (TTS) technologies can be integrated to realize voice command control. Vehicular edge computing (VEC) and vehicular fog computing (VFC) [7, 8] can be used for local data processing, analysis, and reasoning. These newly emerging computing paradigms become an essential part of SIOV, which helps vehicles think, act, and socialize with others link a real person. Furthermore, extensive attention has been paid to resource scheduling and allocation in the context of VEC or VFC [9–11]. In our previous work [7], fog computing is adopted to predict the number of parking places and realize smart parking for vehicles which try to find parking slots in peaking hours.

However, we notice that few of existing works have focused on the issues emerging along with the development of SIOV. For example, how to address the security and privacy related issues in SIOV is also a big concern in SIOV. Different from the traditional social networks, the behaviors of malicious nodes can cause immediate damages to other vehicular nodes in SIOV, e.g., communication interruption, privacy disclosure, information tamper, driving unsafe, etc. As a consequence, to address these issues, we propose a new framework called Unmanned Aerial Vehicles (UAV) enabled social internet of vehicles, where UAVs are integrated into SIOV such that UAVs can assist in security and privacy protection, data processing, and communication range extension. To be specific, the contributions of the paper can be summarized as follows:

1. We elaborate the roles and functionalities vehicles in SIOV are supposed to play and perform, respectively from different perspectives, and analyze the main limiting factors that restrict the development of SIOV.
2. A UAV enabled security framework is proposed to cope with the privacy and security related issues in SIOV. The multiple functionalities of UAVs are discussed respectively. For example, combining the advantages of UAVs, they can serve as a local authority to investigate the legality of vehicular nodes and insure the security of interactive contents among vehicles in SIOV.
3. Several use cases are given in this paper, with purpose of providing some insights within this framework.

The rest of paper is organized as follows. In Sect. 2, we discuss the SIOV evolvement based on different stages. Section 3 introduces the commercial applications of UAVs. In Sect. 4, we analyze the advantages from different perspectives, when UAVs are incorporated into social internet of vehicles. In Sect. 5, two examples are given to motivate our works in this paper. Finally, the conclusion comes at Sect. 6.

## 2 SIOV Evolvement

### 2.1 Roles and Functionalities of Vehicles in SIOV

Considering the potential benefits of SIOV, people expect that the conventional social skills and methods can be extended to the internet of vehicles [4, 16, 17]. Driving safety is the primary issue to be considered whenever travelers (e.g., drivers and passengers) are on the road. Currently, socializing by smart phone has become the most dominant way in daily life, especially for the younger generations. Driver distraction caused by smart phone usage has contributed to numerous traffic crashes worldwide. The need for friendly human-vehicle interactive environment, e.g., socializing by smart vehicles instead of mobile phone becomes increasingly urgent, which constantly stimulates the development of SIOV. To achieve this goal, automotive intelligence design needs to fuse BLU, DL, TTS technologies. Also, auto manufacturers gradually turn their attention to SIOV in recent years. For example, vehicular operating systems including BMW iDrive, Audi MMI, and Mercedes-Benz COMMAND have already embedded social oriented applications, and gained positive reviews from consumers.

As shown in Fig. 1, we classify the functions of intelligent vehicles into eight categories based on consumers' expectation toward what an intelligent vehicle is supposed to have. Some of these functions have already been available in vehicles while others remain the conceptual phase. For example, remote control in the category seven has been realized in most of vehicles, which style themselves as intelligent vehicles such as LYNK&CO and ROEWE from the recent rise of Chinese carmakers. Moreover, according to the characteristics of SIOV, we split the evolvement of SIOV into two different phases – human-vehicle oriented internet of vehicles and vehicle-vehicle oriented internet of vehicles, respectively.

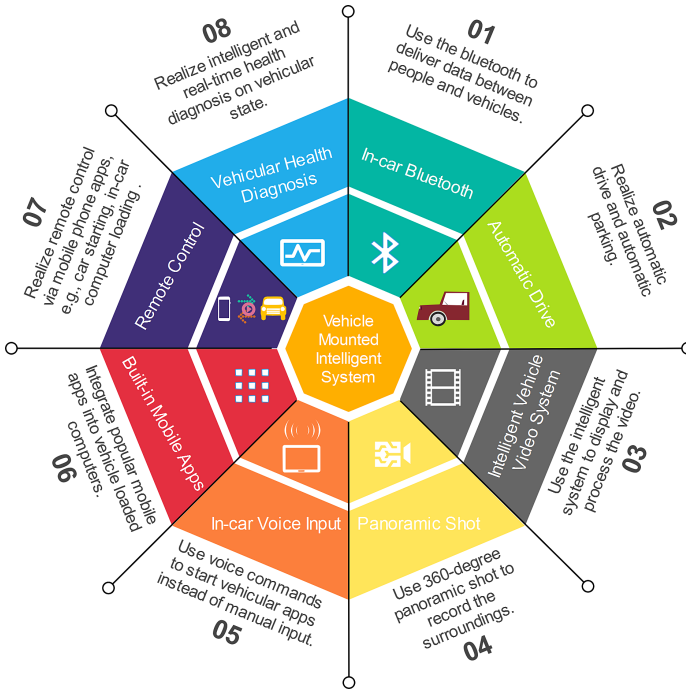


Fig. 1. Functional classification in smart vehicles

In the first phase, SIoV mainly focuses on human-vehicle oriented service provisioning. Most of functions depicted in Fig. 1 belong to this phase. People can make social connections via vehicles instead of mobile phones. However, as far as intelligence is concerned, vehicles in this stage are not smart enough to be engaged in social behaviors. It is still a challenge to integrate social networking software into vehicular operating systems, not to mention the social behaviors of vehicle.

In the second phase, interactions have already shifted from the human-vehicle way to the vehicle-vehicle way. Specifically, a comparison between the two phases from different perspectives is shown in Fig. 2. In contrast to the first phase, phase two truly realizes the social internet of vehicles. A miraculous scenario is anticipated where intelligent vehicle can think, act, and socialize with others link a real person. On one hand, interactions between people and vehicles occur frequently, which can improve the driving pleasure. Furthermore, voice control in vehicles will be the dominant way to socialize among people. On the other hand, spontaneous interactions among vehicles themselves are very common. This kind of interaction usually does not need the involvement of drivers at the beginning. People usually only need to make a decision at the end.

## 2.2 Security Issues in SIoV

Although the prospect of SIoV is tempting, quite a few obstacles remain to be removed from SIoV. One of big concerns is the security issue. The traditional social networks (e.g., Twitter) connect people who have known each other in the real world or are closely associated through certain social connections [4]. However, the vehicular social networks connect vehicles/drivers who are anonymous or unknown to each other beforehand. Drivers tend to trust each other, if they have similar commuter routes or the same brand of vehicles. Then, some potential common interests can be shared and disseminated in the vehicular social networks. During this process, different kinds of attacks can be launched by malicious vehicles, which undermines vehicular social networks and restricts the development of SIoV. Furthermore, these attacks can even cause life-threatening accidents. Specifically, the attack behaviors include, but not limited to:

- Denial of service (DoS) attack. It is an attack behavior that tries to make network connection malfunction, e.g., by jamming vehicular social networks using a vast amount of data and information. These irrelevant messages keep the wireless channels so busy that other legitimate vehicles cannot utilize the communication resources. Thus, data delivery and service provisioning are not available any longer in VSN.
- False message injection. Malicious vehicular node in VSN can broadcast a false message to the network for its own benefits [1]. Thus, the attacker can manipulate the traffic flow and interfere the reasoning of other vehicles, giving rise to anticipated damages to the vehicles in SIoV.
- Impersonation attack. The attacker accesses the resources of the network in the disguise of a legitimate node in VSN. Then false messages can be broadcast on the behalf of that node, which may incur life-threatening damages to drivers.

Features	Phase I	Phase II
Sponsor	Human	Vehicle
Terminator	Vehicle	Vehicle
Automation Degree	Low	High
Man-Machine Interaction	Low Frequency	High Frequency
Human Involvement	Not Much	Much
Safety	Low	High
Privacy	Not Safe	Safe
Social Content	Types Limited	Types Not Limited
Data Size	Average	Extremely Large

**Fig. 2.** Comparison of two phases in the evolvement of SIoV

- Social trust disguise. The attacker disguises its own social trust level, with an aim to obtain the others' trust in the vehicular social network. Then, the attacker may obtain others' privacy information for their illegal benefits.

In addition to these representative attack behaviors, other forms of attack also exist in vehicular social networks [1]. Extensive attention has been focused on vehicular networks protection such as [12–14]. Nevertheless, few of works have been done on the security of vehicular social networks. With the development of SIOV, the issues pertaining to security and trust become significantly important. Accordingly, in this paper we focus our attention on the UAV enabled SIOV security framework, in hope to address the aforementioned issues.

### 3 Commercial Applications of UAVs

UAVs have broken away the originally military application restraint and become ubiquitous in the commercial field. Specifically, the common uses for UAVs in business domain can be outlined as follows. First, UAVs can act as aerial base stations to establish wireless communication links between two entities out of each other's communication range. For instance, UAVs can be applied to the post disaster relief where the terrestrial infrastructures for communication are damaged. Second, better line-of-sight (LOS) characteristic brings about better remote sensing capability, which makes UAVs suitable for information collection and dissemination, e.g., in some harsh environments where people cannot make personal appearance. Third, a new computing paradigm termed aerial fog computing (AFC) is proposed that enhances UAV with computing capabilities, e.g., by equipping them with powerful computing facilities. As a consequence, UAV can provision computing resources such that the captured data can be processed and analyzed on site. Thus, the response latency can be reduced to a great extent in contrast to data processing in the remote cloud center.

Besides the aforementioned use cases, many efforts have also been made to realize the UAV-to-Vehicle (U2V) and Vehicle-to-UAV (V2U) communications [15]. The sensing capability can be further enhanced when Flying Ad-hoc Network (FANET) works collaboratively with Vehicle Ad Hoc Network (VANET). More important, AFC can be combined with VFC or VEC to fully exploit the idle computing resources in vicinity. For example, tasks from UAVs or vehicles can be accomplished with the aid of each other's computing capabilities. By doing so, on one hand, the response latency can be reduced because of local data processing; on the other hand, the pressure over the core network can be mitigated thanks to the reduction of task offloading via the backhaul links.

### 4 UAV Assisted Social Internet of Vehicles

Following the aforementioned introduction about UAV, we in this paper argue that UAV is helpful for social internet of vehicles owing to its high flexibility, fast deployment, and low-cost maintainability. To be more specific, first, UAV can

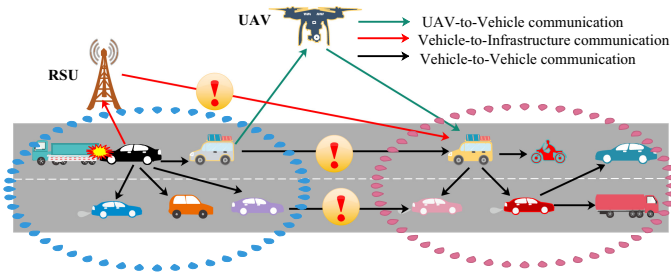


seamlessly connect to the vehicular social networks. Second, UAV can fulfill the key requirements of SIoV such as low response latency and wide communication range. Third, UAVs can assist in coping with security and trust related issues in SIoV. In the next, we will detail these functionalities respectively.

#### 4.1 Network Connectivity

Network connectivity determines the performance of network access of SIoV. And it should be the primary function for vehicles when they function as smart agents in the vehicular social network. Before socializing with each other, vehicles should have the ability to communicate with other entities at the beginning. These entities usually include vehicles and RSUs. VANET provides a foundation for the ubiquitous communications between vehicles themselves, and also between vehicles and RSUs. A variety of wireless communication technologies are available for data delivery and information sharing, e.g., 4G, WiFi, WLAN, Zig-Bee, Bluetooth, and dedicated short-range communication (DSRC). On another hand, with the advent of 5G technology, the communication between IoT devices will become much easier than before, for the reason that 5G supports end-to-end communications with higher data rates. All these advantages will boost the prosperity of SIoV.

However, an obvious drawback in current wireless communication technologies applied to VANET and VSN is that the communication range is limited. With the help of UAVs, the communication range can be extended. For example, an example is given in Fig. 3, where a car accident happens, however, the information about the incident cannot be disseminated due to lack of suitable relay nodes to forward the information. When UAV becomes engaged with this traffic incident, the problem can be readily solved. In this scenario, UAV can serve as a relay node to broadcast the information in real time. As a whole, UAV can boots the network connectivity of SIoV to a great extent.



**Fig. 3.** An example of communication range extension using UAV[15]

## 4.2 Data Processing

Each vehicular node in SIOV should be equipped with powerful computing and storing resources such that they are capable of local data processing. This feature is vital to SIOV, since the “intelligence” of vehicle depends on the processing capability of vehicular brain (i.e., vehicle loaded computers). For instance, fast data processing and analysis reduces the response latency during the decision making, which is vitally important to driving safety. Furthermore, local data processing is one of the key enablers for vehicles to logically reason, train, compute, and analyze. If these oriented tasks in application are executed at the cloud center, the response latency could be extremely long, attributed to task offloading via the backhaul links. Last, vehicles as intelligent agents can be stimulated by the potential benefits. One of these benefits is the payoff earned by contributing their computing resources. When it comes to finding a suitable scheme to fulfill these functions, VFC and VEC are proposed that fully exploit the computing capabilities of vehicles, with an aim to accomplish these goals.

On another hand, the development of SIOV will bring about mountains of data. Of these data, some need the wireless communication resources (e.g., multimedia data sharing) while others need the computing resources such as vehicular applications and tasks. Against this background, the number of vehicular applications in SIOV is also explosively increasing. Thus, the limited computing capabilities of vehicle loaded computer systems can no longer satisfy the demands. It is not a good choice to turn to cloud computing for help as discussed above. On another hand, feasible solutions are discussed in the vehicular edge computing, where RSU connected to edge servers can act as the edge node to perform the computation tasks. Attributed to expensive deployment and maintenance, full coverage of RSU currently has not been achieved.

As a result, we in this paper argue that UAV can help turn this situation around, since UAVs equipped with powerful computing resources can constitute aerial fog computing, a new computing paradigm which can provision computing services on the wing. We believe that aerial fog computing can assist social contacts of vehicles nodes in SIOV, when tasks or applications need to be outsourced.

## 4.3 Security Capability

Social internet of vehicles evolves from the internet of vehicles while incorporating IoTs, machine learning, cloud computing and edge computing. Like other types of social networks, SIOV should address well the issues pertaining to security and privacy. When socializing with other entities (e.g., people, vehicles, RSUs), how to prevent and detect the attack behaviors is really one of big concerns faced by SIOV. The sensitive information may be disclosed to the malicious nodes and used for illegal benefits. Vehicular social networks are of high dynamics where vehicular nodes can join or leave for free at any time. The social information in this background is vulnerable to attacks like eavesdropping, tampering,

forgery and replay. Worse still, it lacks efficient and legitimate entities to detect whether information is pushed under malicious human intervention.

Owing to the merits of UAVs, they can be very helpful for preventing and detecting the attack behaviors of malicious nodes in VSN. For example, flying above the serving area, they can collect, analyze, detect the behaviors of the vehicular nodes in SIOV, with the purpose of finding out and tracking the malicious nodes. Then the regular vehicles can be informed of the malicious ones to further isolate them. In UAV, various trustworthiness mode can be also defined such that new security evaluation and management framework can be established to ensure the security of SIOV.

## 5 Use Cases

In this section, two examples are given to motivate our work in this paper.

**Example 1.** Traffic jam at morning peak hours annoys every driver on the road. How to avoid the traffic jam and improve the efficiency of road poses a major challenge to government as well as citizens. In the era of SIOV, we believe that the traffic jam can be mitigated dramatically. A social internet of vehicles can be established where each vehicle can independently think, socialize and reason. With the help of UAVs, global information about the traffic can be captured such as queue length, phase timing and phase sequence at each intersection. Besides, vehicular information including the destination, location, and velocity are also shared in VSN. These information is helpful for assisting decision-making. Each vehicle acting as an intelligent agent makes the best decision toward their optimization objective by reasonably adjusting the velocity, route and so on.

**Example 2.** Assume that there is an attacker on the road. He wants to manipulate the traffic flow by broadcasting a false car accident in VSN, and reminds other vehicles of avoidance. If the previous behaviors of this attacker are trustworthy, other vehicles may trust him this time. Then, he succeeds in manipulating the traffic flow. However, we can avoid this situation with the help of UAVs. For example, thanks to great LoS, UAVs can easily capture the picture of the road, and further judge whether a car accident exists. On another hand, UAVs can also obtain the information of passing vehicles prior to the attacker on that road and analyze their velocity and acceleration to aid the decision-making. Algorithms can also be applied to car accident detection.

## 6 Conclusion

Internet of Things, cloud computing, edge computing and mobile internet are considered to be the main moving forces that stimulate the development of SIOV. We in this paper envision an enticing scenario where UAVs meet the social internet of vehicles. FANET and AFC can be leveraged to assist SIOV in extending the communication range, boost the data processing abilities and improving the security. We in this paper talk about these affects from different

perspectives and discuss how UAVs can seamlessly connect to SIOVs. For the further works, we plan to design efficient strategy to evaluate the trustworthiness of VSN. We also need appropriate measures to detect and track the malicious vehicles to ensure the security of SIOV during the social contacts.

## References

1. Maglaras, L.A., Al-Bayatti, H.A., He, Y., Wagner, I., Janicke, H.: Social internet of vehicles for smart cities. *J. Sens. Actuator Netw.* **5**(1), 3 (2016)
2. Pandey, M.K., Subbiah, K.: Social networking and big data analytics assisted reliable recommendation system model for internet of vehicles. In: Hsu, C.-H., Wang, S., Zhou, A., Shawkat, A. (eds.) IOV 2016. LNCS, vol. 10036, pp. 149–163. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-51969-2\\_13](https://doi.org/10.1007/978-3-319-51969-2_13)
3. Bouk, S.H., Ahmed, S.H., Kim, D., Song, H.: Named datanetworking based its for smart cities. *IEEE Commun. Mag.* **55**(1), 105–111 (2017)
4. Luan, T.H., Lu, R., Shen, X., Bai, F.: Social on the road: enabling secure and efficient social networking on highways. *IEEE Wirel. Commun.* **22**(1), 44–51 (2015)
5. Wu, H., Tang, H., Dong, L.: A novel routing protocol based on mobile social networks and internet of vehicles. In: Hsu, R.C.-H., Wang, S. (eds.) IOV 2014. LNCS, vol. 8662, pp. 1–10. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-11167-4\\_1](https://doi.org/10.1007/978-3-319-11167-4_1)
6. Atzori, L., Iera, A., Morabito, G., Nitti, M.: The social internet of things (SIoT) - when social networks meet the internet of things: concept, architecture and network characterization. *Comput. Netw.* **56**(16), 3594–3608 (2012)
7. Tang, C., Wei, X., Zhu, C., Chen, W., Rodrigues, J.P.C.: Towards smart parking based on fog computing. *IEEE Access* **6**, 70172–70185 (2018)
8. Tang, C., Wei, X., Zhu, C., Wang, Y., Jia, W.: Mobile vehicles as fog nodes for latency optimization in smart cities. *IEEE Trans. Veh. Technol.* (2020). <https://doi.org/10.1109/TVT.2020.2970763>
9. Xu, C., Lei, J., Li, W., Fu, X.: Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Trans. Netw.* **24**(5), 279–2808 (2016)
10. Guo, J., Song, Z., Cui, Y., Liu, Z., Ji, Y.: Energy-efficient resource allocation for multi-user mobile edge computing. In: 2017 IEEE Global Communications Conference, Singapore, December, pp. 1–7 (2017)
11. Hou, X., Li, Y., Chen, M., Wu, D., Jin, D., Chen, S.: Vehicular fog computing: a viewpoint of vehicles as the infrastructures. *IEEE Trans. Veh. Technol.* **65**(6), 3860–3873 (2016)
12. Kumar, N., Iqbal, R., Misra, S., Rodrigues, J.J.: An intelligent approach for building a secure decentralized public key infrastructure in VANET. *J. Comput. Syst. Sci. (JCSS)* **81**, 1042–1058 (2015)
13. Scheppe, H., et al.: Securing car2X applications with effective hardware software codesign for vehicular on-board networks. In: Proceedings of the 27th Joint VDI/VW Automotive Security Conference, Berlin, 11–12 October 2011 (2011)
14. Lu, R., Lin, X., Liang, X., Shen, X.: A dynamic privacy-preserving key management scheme for location-based services in VANETs. *IEEE Trans. Intell. Transp. Syst.* **13**, 127–139 (2012)
15. Tang, C., Zhu, C., Wei, X., Peng, H., Wang, Y.: Integration of UAV and fog-enabled vehicle: application in post-disaster relief. In: 25th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2019, Tianjin, China, 4–6 December, pp. 548–555 (2019)

16. Loke, S.W.: Cooperative automated vehicles: a review of opportunities and challenges in socially intelligent vehicles beyond networking. *IEEE Trans. Intell. Veh.* **4**(4), 509–518 (2019)
17. Li, T., Zhao, M., Liu, A., Huang, C.: On selecting vehicles as recommenders for vehicular social networks. *IEEE Access* **5**, 5539–5555 (2017)



# An Efficient Influence Maximization Algorithm Based on Social Relationship Priority in Mobile Social Networks

Xinxin Zhang<sup>1,2</sup>, Li Xu<sup>1,2(✉)</sup>, and Min Gao<sup>1,2</sup>

<sup>1</sup> College of Mathematics and Informatics,  
Fujian Normal University, Fuzhou, Fujian, China  
xuli@fjnu.edu.cn

<sup>2</sup> Fujian Provincial Key Laboratory of Network Security and Cryptology,  
Fujian Normal University, Fuzhou, Fujian, China

**Abstract.** The mobile social network (MSN) combines techniques in social science and wireless communications for mobile networking. The MSN can be considered as a system which provides a variety of data delivery services involving the social relationship among mobile users. The key problem in MSNs is Influence Maximization (IM), which aims at finding the top- $k$  influential users from the mobile social network and contributing to the spread of maximum information, the users may have different attitudes (positive/negative) towards a message when the message appears in MSNs. In this paper, we first model the mobile social network as the topological graph based on social priority topological to study the social influence. Then we innovatively propose a scheme which integrates ITÖ algorithm into PSO algorithm to solve the problem of maximizing the influence in MSNs. Finally, experimental evaluation shows that the scheme we proposed to identify influential nodes is more accurate and efficient than other schemes by comparison, and the probability of maximizing the influence of our scheme can reach to 56%.

**Keywords:** Mobile social networks · Influence Maximization · First-priority relation graph · PSO algorithm · ITÖ algorithm

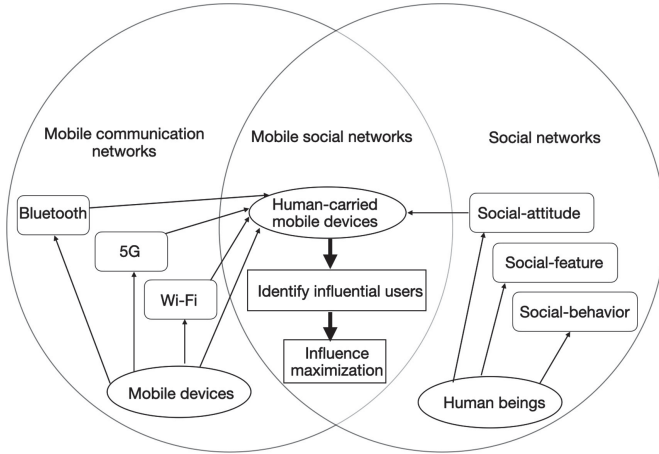
## 1 Introduction

The mobile social network (MSN) is defined by Professor Jie Wu of Temple University in 2013 [1] (see Fig. 1). It is a cross network composed of social network and mobile communication network, where users communicate with one another using mobile phones. The mobile social network is a special communication network composed of data or information transmission and interaction by mobile devices. Nowadays, the mobile social network plays an important role in spreading information. This spreading process has huge practical importance

---

Supported by National Natural Science Foundation of China.

in selecting influential twitters [2], personalized recommendation, target advertisement [3], etc. Therefore, scholars recent years have witnessed a significant attention in the study of *influence propagation* in MSNs.



**Fig. 1.** The mobile social network

Consider the case of personalized recommendation, the goal of the platform is to attract the users. The best way to do this is to select a set of highly influential users and distribute them information. If they like the information, they will share the information to their neighbors, many of the neighbors will share the information to their neighbors. If the information can be distributed among the highly influential users and the problem here bottom down to select influential users from the network. This problem is known as *Social Influence Maximization Problem*. The key problem in this is Influence Maximization (IM), which aims at finding the top- $k$  influential users (seed users) from the mobile social network [4]. Kempe *et al.* establish that the optimization problem is NP-hard [5]. Most existing influence spread models did not consider the attitude of users. In reality, due to the diversity of the mobile social network and users' preference, product quality or other reasons, people may have different attitudes towards an entity (products, news, etc.). The users may have different attitudes (positive/ negative) towards a message when the message appears in MSNs. Users have their own choice to accept or reject information. It is necessary to take users' attitude into account in the process of influence spread.

A hybrid approach which combines PSO algorithm and ITÖ algorithm has been adopted in this paper. The mobile social network is not completely regular, the activity of users in mobile social network is a stochastic process. The users in the networks frequently communicate with their close neighbors and also by chance to connect with some distant vertices. In this way, this network has the particle motion feature. Particle Swarm Optimization (PSO) [6] is a heuristic

search method that simulates the movements of a flock of birds which aim to find food. The relative simplicity of PSO and the population-based technique and information sharing mechanism associated with this method have made it a natural candidate to be extended for single as well as multi-objective optimization. The original intent was to graphically simulate the graceful but unpredictable choreography of a bird flock. ITÖ algorithm analyzes the movement of particles from the microscopic point of view, and mimic kinetic studies of particles' interactive colliding in the particle system from designing algorithm and solving problems. In the hybrid algorithm, the advantages of the two algorithms are taken and a helpful strategy is employed to solve function optimization problems, that is we simultaneously exert an additional wave process to drift intensity and fluctuate intensity.

To summarize, we present our main contributions as follows.

- We consider an extension of the well-know Influence Maximization Problem in mobile social networks based on social priority relationship.
- We propose an efficient influence maximization algorithm which integrates ITÖ algorithm into PSO algorithm in MSNs.
- Experimental evaluation shows that the scheme we proposed to identify influential nodes is more accurate and efficient than other schemes by comparison, and the probability of maximizing the influence of our scheme can reach to 56%.

The rest of the paper is organized as follows: Sect. 2 reviews related work. Section 3 gives the mathematical model. The proposed scheme is discussed in Sect. 4. Section 5 shows experimental results and we conclude the paper in Sect. 6.

## 2 Related Work

The previous works have extensively studied in Influence Maximization Problem over the past years. The greedy algorithms [7, 8], the heuristic algorithms [9, 10] and fluidspread greedy algorithm [11] are the typical solutions. We briefly review the related work in this section.

### 2.1 Influence Spread

Currently, many efforts have been made to discover the most influential nodes for influence in social networks. Bian *et al.* [12] reviewed and classified existing literature on top- $k$  nodes identification. Liu *et al.* [13] proposed a fast and efficient algorithm for mining top- $k$  nodes in complex networks. Fei, Mo and Deng [14] studied a novel method is proposed to identify influential nodes based on combining of the existing centrality measures. Zhang *et al.* [15] proposed a trust-based most influential node discovery method for discovering influential nodes discovery to improve the degree evaluation mechanism. Amir, Ali and Ahmad [16] proposed a new method to identify influential users in a social network by



considering those interactions that exist among the users. After identifying the most influential nodes, it is also very important to spread the influence of these nodes effectively, this problem has also attracted the attention of many scholars. Li *et al.* [17] surveyed and synthesized a wide spectrum of existing studies on IM from an algorithmic perspective. Taninmis, Aras and Altinel [18] deal with a competitive Influence Maximization Problem which can be formulated as Stackelberg game. Gao *et al.* [19] proposed the scheme of influence maximization based on activity degree in mobile social networks.

## 2.2 Particle Swarm Optimization and ITÖ Algorithm

Particle Swarm Optimization (PSO) [20] was first proposed by Kennedy and Eberhart in 1995, it is a global optimization algorithm based on the swarm intelligence optimization which is inspired from animal behavior and social psychology. Eberhart and Shi [21] focused on the engineering and computer science aspects of developments, applications and resources related to particle swarm optimization. Gong *et al.* [22] proposed a discrete particle swarm optimization algorithm to optimize the influence criterion. In particle swarm optimization algorithm, the operation is try to find the optimal solution according to the two equations by successive iteration, particles get updated to move in the direction of global optimal solution by tracking two extreme values as follows [23]:

$$v_{id}^{t+1} = wv_{id}^t + c_1r_1(p_{id} - x_{id}^t) + c_2r_2(p_{gd} - x_{id}^t) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (2)$$

In which  $i = 1, 2, \dots, N$ ,  $N$  is number of particles in the population;  $t = 1, 2, \dots, M$ ,  $M$  represents the maximum iteration count of the population;  $v_{id}^t$  denotes the  $d$ -dimensional component of the  $i$  th particle's velocity vector in the  $t$  th iteration and  $x_{id}^t$  tells position vector,  $p_{id}$  represents the  $d$ -dimensional component of the  $i$  th particle's individual extreme value (the particle's historical optimal value);  $p_{gd}$  refers to the  $d$ -dimensional component of the current global optimal value;  $r_1$  and  $r_2$  are random numbers that generated subject to  $U(0, 1)$  distribution;  $c_1$  and  $c_2$  are learning factors;  $w$  is the inertia factor.

ITÖ algorithm [24] inspired by itö stochastic process is a new class of evolutionary algorithm, it analyzes the movement of particles from the microscopic of view, and mimic the kinetic studies of particles' interactive colliding in the particle system for designing algorithm and solving problems. Yi *et al.* [23] designed a hybrid algorithm that integrates ITÖ algorithm with PSO algorithm for solving numerical optimization problem. Wang *et al.* [25] proposed a novel efficient ITÖ algorithm to solve the Influence Maximization problem. ITÖ algorithm [26] suppose  $X = (X(t), t \leq 0)$  satisfying itö integral as for  $0 \leq t_0 < T$ ,  $X(t) - X(t_0) = \int_{t_0}^t b(s, X(S))ds + \int_{t_0}^t \delta(s, X(S))dB(s)$ , then  $X$  is called itö (stochastic) process, where  $\int_{t_0}^t b(s, X(S))ds$  is called as drifting rate and denotes the general tendency of stochastic process. The item  $\int_{t_0}^t \delta(s, X(S))dB(s)$  the

tracks fluctuation of variable  $X$  and it is called fluctuation rate. Furthermore, ITO uses environmental temperature to control the motion ability of the population, and uses particles radius to describe the characteristics of a particle in Brown motion.

### 3 Mathematical Model

In this section, we model mobile social network as social relationship topological graph, model the influence spread process as a swarm intelligence optimization, respectively. They provide convenience for the next section on maximizing the influence of mobile social network with graph and stochastic process algorithm.

#### 3.1 Mobile Social Networks

The mobile social network not only defines the behavior of all entities (people, devices, or systems) but also helps to understand different relations among them (see in Fig. 2). In principle, a social network is a structure of entities (individuals, organizations, and systems) that are connected to each other through one or more interdependencies [27]. As the mobile devices are carried by people, the knowledge of social behavior and structure can be one of the key information or designing and providing efficient and effective data communications services [28], thus the main body of a mobile social network is human, when a message is arrived, the user’s attitude determines whether to receive and transmit the message. Usually a user has two attitudes towards a message, acceptance or rejection. We define acceptance (rejection) as a positive (negative) attitude.

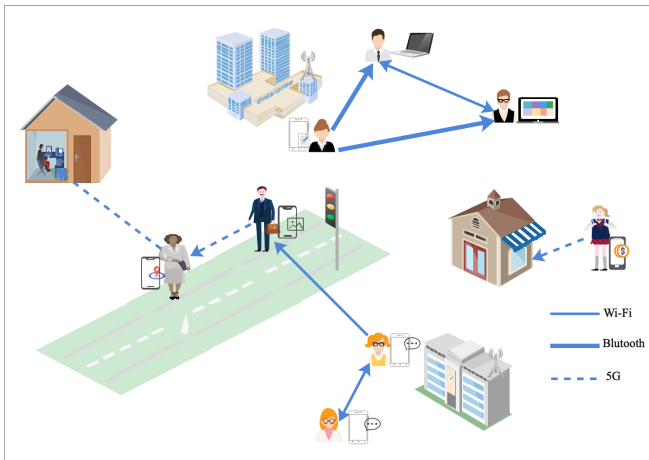


Fig. 2. The model of mobile social networks

We consider model the mobile social network as a *social relationship priority topology graph*. We use a graph  $G = (V(G), E(G))$  to represent a mobile social network, where *node-set* denoted as  $V(G)$  is to users, *edge-set* denoted as  $E(G)$  is to the interconnection between users.  $uv$  or  $(u, v) \in E(G)$  represents an edge between nodes  $u$  and  $v$ . The *neighborhood* denoted as  $N_{(G)}(v)$  of node  $v$  in  $G$  is defined as the set of all nodes which are adjacent to  $v$ , i.e.,  $N_{(G)}(v) = \{u \in V(G) | uv \in E(G)\}$ . When  $G$  is clear from the context, we use  $N(v)$  to replace  $N_{(G)}(v)$  [29]. In the model, initially each node in the network holds one of the two opposite opinions, we denote them A or B, respectively, and we assume that opinion A is positive and opinion B is negative.

### 3.2 Model the Influence Spread Process

In the mobile social network  $G = (V(G), E(G))$ , the influence spread originates from the node  $v \in V$ . The node  $v$  is called the *seed node*, and we use the notation  $S$  to denote the set of seed node. Each node  $v \in V$  in MSNs holds different *attitudes* to an entity. The different attitude should be considered in the process of influence propagation. The node  $v$  can be regarded as *active node* if it is willing to receive the message and send it to the next node. In the mobile social network  $G = (V(G), E(G))$ , given an influence diffusion model and the seed set  $S$ , the *influence spread* is defined as the inactive users become active through the process of influence diffusion.

The process of influence spread is a stochastic process, it similar with swarm intelligence optimization. The particle swarm concept originated as a simulation of a simplified social system. ITÖ algorithm belongs to a variant of swarm intelligence optimization algorithm, which simulates the irregular movement of pollens in the surface of water [30]. We apply the ITÖ algorithm to model the influence spread. The seed node  $v \in V$  is denoted as the *particles*. In the Browns motion, the small particles will be attracted by the big particles. *Drift intensity* represents the general trend of random process, it corresponds to the change of particle position. In mobile social networks, affected by the actual environment, the user's attitude towards a message is likely to change from A to B, thus, *fluctuate intensity* represents the mutation of user's attitude. We use the temperature  $T$  to denote the users attitude,  $T^+$  denotes positive attitude A,  $T^-$  denotes negative attitude B.

## 4 Proposed Scheme

In this section, we propose the approach about social influence maximization in mobile social networks. We first put forward the calculation method to identify influential nodes, then we use the stochastic process algorithm to maximize the influence.

#### 4.1 Identify Influential Nodes

After modeling, in the process of identify influential nodes, we should not only consider the properties of nodes in the topological graph, but also the user's personal will in mobile social networks. Because the main body of mobile social networks is human, everyone is an individual with social emotion and social attitude. When the information is coming, each user has the right to accept or reject. Thus, our approach combines the nodes' in-degree and the user's personal will to calculate the user's influence.

**Definition 1** (Initial active nodes). In mobile social networks, each node in the network holds one of the two opposite opinions A or B, initially, we denote  $V_A$  ( $V_B$ ) as  $v_i \in V(G)$  has opinion A (B). We further divide the active status into opinion A active and opinion B active, nodes will be activated at different times, so we denote that  $S^A(t) = \{v_i | v_i \in V_A, v_i \text{ is active at time } t\}$ ,  $S^B(t) = \{v_i | v_i \in V_B, v_i \text{ is active at time } t\}$ , then the initial active node set of A is  $S^A(0) = \{v_i | v_i \in V^A, t = 0\}$ . Since  $T$  to denote the users attitude,  $T^+$  denotes positive attitude A,  $T^-$  denotes negative attitude B. When node  $v_i$  transmits information, for  $v_j \in N(v_i)$ , no matter that  $v_j$ 's attitude is  $T^+$  or  $T^-$ , we make  $\sum_{v_j \in N(v_i)} |T_{ij}| \leq 1$ ,  $-1 \leq T_{ij}^- < 0$ ,  $0 < T_{ij}^+ \leq 1$ ,  $T_{ij}$  is  $v_j$ 's personal attitude to the information from  $v_i$ .

**Definition 2** (Contribution factor). In mobile social networks, we use  $f(v_i)$  as an index to measure the contribution of neighbor nodes to receiving information. Since each node has two attitudes for  $v_i$ , so is  $v_j \in N(v_i)$ . Initially ( $t = 0$ ), we divide the set  $N(v_i)$  into two sets, we have  $S^A(0) \subseteq N(v_i)$ ,  $S^B(0) \subseteq N(v_i)$  and  $S^A(0) \cap S^B(0) = \emptyset$ . Then we define that contribution factor is  $f(v_i) = \sum_{v_j \in S^A(0)} T_{ij}^+ + \sum_{v_j \in S^B(0)} T_{ij}^-$ . Defined by Definition 1 we know that  $\sum_{v_j \in N(v_i)} |T_{ij}| \leq 1$ . Thus, more users are willing to receive the information when  $f(v_i) > 0$ , that is the positive attitude have a greater influence. The bigger  $f(v_i)$ , the more positive users. We obtain the influence of initial active nodes in mobile social networks in Sect. 4.1, the next step is to spread the influence of those nodes. As we know, the process of influence spread is stochastic process, closely related to time. The elements in set  $S(t)$  is determined by  $S(t - 1)$ , we proposed that

$$v_j \in \begin{cases} S^A(t), & \sum_{v_j \in S^A(t-1)} T_{ij}^+ + \sum_{v_j \in S^B(t-1)} T_{ij}^- > 0; \\ S^B(t), & \sum_{v_j \in S^A(t-1)} T_{ij}^+ + \sum_{v_j \in S^B(t-1)} T_{ij}^- \leq 0. \end{cases} \quad (3)$$

In this way, users with positive attitude about  $v_i$  can be identified, this paves the way for the most influential users computing.

**Definition 3** (Intimacy factor). In mobile social networks, the premise for users to receive messages is that messages are arrived, this mapped to social relationship priority topology graph is the nodes have in-degree. We define that the

intimacy factor of  $v_j$  is  $C_{ij} = \frac{d_j^{in}}{m_{ij}}$ , in which  $m_{ij}$  is all the edges between  $v_i$  and  $v_j$ ,  $d_j^{in}$  is the in-degree of  $v_j$ , we use the index measure the probability of information arrival.

**Definition 4** (Influence function). Therefore, the probability of a node being activated is  $C_{ij} \cdot f(v_i)$ . However, in mobile social networks, the loss of information is inevitable in the process of information transmission, the available information transmitted by the second hop will be less than that transmitted by the first hop. We give a weight  $\omega$  ( $\omega \leq 1$ ) to the availability of information,  $\omega_{ij}$  is the availability of the information transmitted by  $v_i$  to  $v_j$ . Thus, the influence function is denote as

$$I = \lambda C_{ij} \cdot f(v_i) + (1 - \lambda) \cdot \omega_{ij} \quad (4)$$

where,  $\lambda$  is a parameter, the function  $I$  can calculate the social influence of users in mobile social networks, we can find the top  $k$  most influential users.

## 4.2 Influence Maximization

Mobile social networks have had a great impact on information propagation and become a good platform for people to exchange opinions and to propagate information. The goal of the influence maximization is to use those influential users to influence as many direct and indirect friends as possible. The particles swarm optimization concept consist of, at each time step, changing the velocity each particle toward its global version. ITÖ algorithm to a variant of swarm intelligence optimization algorithm, we use it to maximize mobile social influence, which will have a good effect.

**Definition 5** (Drift intensity). For each particle, evaluate the desired optimization fitness function in variables. Drift intensity controls the particles more toward the global optimum solution. We can obtain top  $k$  influential nodes from Algorithm 1 in Sect. 4.1. A particle represents an user in mobile social networks, we use the movement trend of particles to the optimal solution to represent the users' yearning for the most influential users, it is a good description of the user's mobile trend.

**Definition 6** (Fluctuate intensity). The fluctuate operator controls the random fluctuation of particles in the whole solution space, and ensures the diversity of the population. In mobile social networks, every user has his own social thought, his attitude is likely to sudden change for environmental reasons, these sudden changes reflect the complexity of social networks. Thus, we map the fluctuate operator of the population to the attitude changing in mobile social networks.

## 4.3 An Efficient Influence Maximization Algorithm

In the above work, we have proposed the scheme of identifying the key nodes and the scheme of maximizing the influence. In this section, we design an algorithm

to use particle motion to solve the problem of stochastic process of maximizing influence in mobile social networks. We integrate the drift intensity and fluctuate intensity as the factor  $c_1$  and  $c_2$  to PSO algorithm simultaneously, and particle radius represents the trend of an optimization process. In this way, the search process can be very likely to jump out of the local optimal solution and lead the population to move in the direction of global optimal solution. The specific algorithm is as follows (see Algorithm 1), we combine the characteristics of particle motion to set initial position of the particle and set its initial velocity, and then calculate the fitness value of all particles according to Influence function (4.2). We use the indicators in ITÖ algorithm which we define in Sect. 4.2 to solve the fluctuate and drift in particle motion, use drift intensity and fluctuate intensity to update the velocity and location of population according to PSO function (2.1), (2.2), finally, global optimal nodes are obtained and realize the Influence Maximization.

---

**Algorithm 1.** An Efficient Influence Maximization Algorithm

---

**Require:** Initial position of the particle and set its initial velocity

**Ensure:** Global optimal nodes

```

1: for Influence function (4.2) do
2:   if Calculate the fitness value of all particles then
3:     The global optimal value
4:   else
5:     Use drift intensity and fluctuate intensity to update the velocity and location
       of population according to PSO function (2.1), (2.2)
6:   end if
7: end for
8: return Global optimal nodes

```

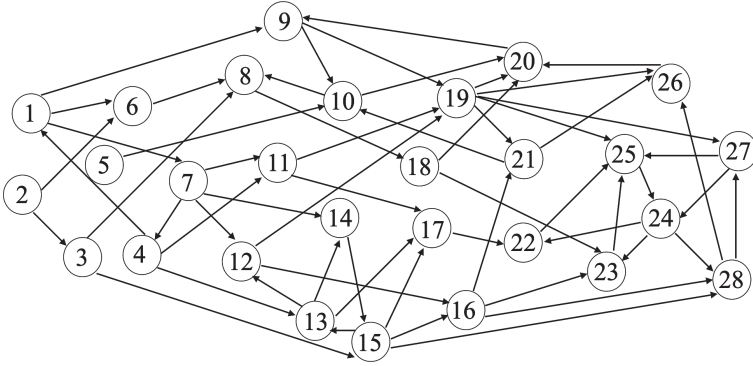
---

## 5 Experimental Evaluation

In this section, we first use a experiment to verify the accuracy of the proposed method for identifying important nodes, then we use the data to evaluate the efficiency of our influence maximization method.

### 5.1 Accuracy

The accuracy here refers to whether the important nodes identified by our scheme are accurate, and whether these identified nodes are really important nodes in the real network. We conduct experiments on parameters of experimental equipment: Windows10, 64 bit operating system +16 GB memory+Jupyter Notebook+Multiple Python packages (networkx matplotlib, numpypackage). Jupyter notebook is a web-based application for interactive computing, Jupyter notebook is a web-based application for interactive computing, Matplotlib is a powerful Python drawing and Data Visualization Toolkit, which can be used with numpy, and is very suitable for interactive drawing.



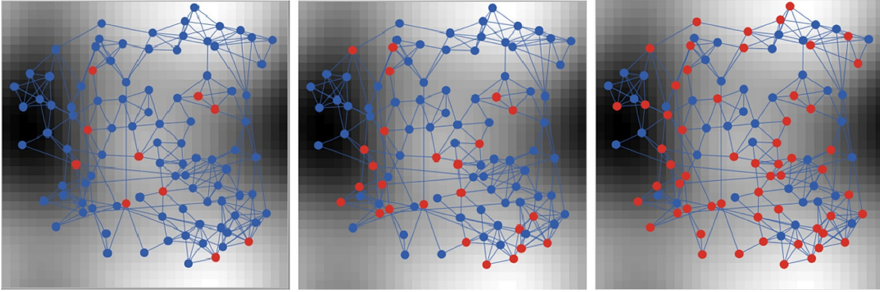
**Fig. 3.** Academic exchange network in mobile social networks

As show in Fig. 3, the sample network is a real small academic exchange digraph with 28 nodes in mobile social networks. Nodes represent authors, edges represent the relationship between reference and referenced. According to the figure, we can know that the paper of node 2 is referenced by node 3 and node 6, which means that node 3 and node 6 have a positive attitude towards the information transmitted by node 2. We calculate the value of influence according to influence function  $I = \lambda C_{ij} \cdot f(v_i) + (1 - \lambda) \cdot \omega_{ij}$ . We set  $\lambda = 0.5$ ,  $\omega = 0.8$ ,  $k = 10$ , and we generate the top 10 important nodes in turn which are 19, 20, 10, 17, 24, 21, 3, 8, 13, 25. Since we don't know which nodes are important in advance, we use the correlation between calculation and other methods to measure the accuracy of our method. The method referred to is classical, we take the absolute value of correlation calculation results and set the value range of the correlation as  $[0, 1]$ , the higher the correlation (i.e. the closer to 1) is, the higher the accuracy of our method can be considered. We use our sample network and two real datasets from Stanford University to do the experiment (they are respectively wiki vote and P2P Gnutella 0). In Table 1, we can see that in our sample network, the correlation of our scheme and betweenness centrality is 0.815, the correlation of our scheme and closeness centrality is 0.318, the correlation of our scheme and katz centrality is 0.611, the correlation of our scheme and harmonic centrality is 0.659. In wiki vote data (<http://snap.stanford.edu/data/wiki-Vote.html>.) the correlation of our scheme and betweenness centrality is 0.334, the correlation of our scheme and closeness centrality is 0.480, the correlation of our scheme and katz centrality is 0.700, the correlation of our scheme and harmonic centrality is 0.529. In P2P Gnutella 0 data (<http://snap.stanford.edu/data/p2p-Gnutella08.html>). We can see that the correlation of our scheme and betweenness centrality is 0.423, the correlation of our scheme and closeness centrality is 0.555, the correlation of our scheme and katz centrality is 0.324, the correlation of our scheme and harmonic centrality is 0.525. The results of these correlations show that our scheme is very effective and feasible.

**Table 1.** The correlation coefficient in three networks

Networks	BC	CC	KC	HC
Sample network	0.815	0.318	0.611	0.659
Wiki vote data	0.334	0.480	0.700	0.529
P2P Gnutella 0	0.423	0.555	0.324	0.525

## 5.2 Influence Spread



**Fig. 4.** Simulation experiment process of influence maximization in Netlogo (Color figure online)

In this section, we use experiments to show the effect of influence spread of our scheme. We simulate the experiment process of influence maximization in Netlogo (see in Fig. 4) first, then we illustrate the rate of influence propagation and the probability of influence maximization by experiments. We conduct experiments on parameters of experimental equipment: Windows10, 64 bit operating system +16 GB memory + Netlogo 6.1.1 simulation platform (a multi-agent programmable modeling environment). In Netlogo (Netlogo is a programmable environment for simulating natural and social phenomena), each turtle turn corresponds to each particle in the PSO algorithm, and also to each node in the network. Each turtle has its own tile patch, which corresponds to the particle position in PSO algorithm. Some parameters for generating the initial network: the number of nodes is 100, average node degree is 5, seed is 10. The PSO algorithm spread chance is 1.0, particle is speed limit is 5, attraction to personal best is 2.0, attraction to global best is 1.5.

Figure 4 shows the simulation experiment process of influence maximization in Netlogo, from left to right are the initial generation graph, after 5-step optimization of the Algorithm 1 generation graph and the final result of executing Algorithm 1 respectively. In this experimental network, the total number of nodes is 100, the red nodes represent key nodes, and the blue nodes represent the remaining nodes. We can see clearly that there were only 10 key nodes at



the beginning. After five steps of the algorithm, there are 24 red nodes, the influence of 10 key nodes spread to 24 right now. The final result of executing Algorithm 1 is 56 red nodes, the ultimate number of nodes influence is 56. We can know that the probability of maximizing influence is 56%, the results are considerable (since influence maximization is NP-hard problem, the upper limit of its optimization is  $1 - \frac{1}{e} - \epsilon \approx 63\%$  [5]).

## 6 Conclusion

We consider an extension of the well-know Influence Maximization Problem in a mobile social network based on social priority relationship. We propose an efficient influence maximization algorithm which is integrates ITÖ algorithm into PSO algorithm. Experimental evaluation shows that the scheme we proposed to identify influential nodes is more accurate and efficient than other schemes by comparison, and the probability of maximizing the influence of our scheme can reach to 56%.

**Acknowledgement.** The authors would like to thank the National Science Foundation of China (Nos. U1905211, 61771140, 61702100, 61702103).

## References

1. Wang, Y., Yang, W.-S., Wu, J.: Analysis of a hypercube-based social feature multi-path routing in delay tolerant networks. *IEEE Trans. Parallel Distrib. Syst.* **24**(9), 1706–1716 (2013)
2. Weng, J., Lim, E.-P., Jiang, J., He, Q.: TwitterRank: finding topic-sensitive influential twitterers. In: *The Third ACM International Conference on Web Search and Data Mining*, pp. 261–270 (2010)
3. Li, Y., Zhang, D., Tan, K.-L.: Real-time targeted influence maximization for online advertisements. *Proc. VLDB Endow.* **8**(10), 1070–1081 (2015)
4. Banerjee, S., Jenamani, M., Pratihari, D.K.: A survey on influence maximization in a social network. *Knowl. Inf. Syst.* **62**(9), 3417–3455 (2020). <https://doi.org/10.1007/s10115-020-01461-4>
5. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: *Internet Conference Knowledge Discovery Data Mining*, pp. 137–146 (2003)
6. Rini, D.P., Shamsuddin, S.M., Yuhani, S.S.: Particle swarm optimization: technique, system and challenges. *Int. J. Comput. Appl.* **14**(1), 19–27 (2011)
7. Wang, Y., Cong, G., Song, G., Xie, K.: Community-based greedy algorithm for mining top- $k$  influential nodes in mobile social networks. In: *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1039–1048. ACM, Washington, D.C. (2010)
8. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks, pp. 199–208(2009)
9. Zhou, C., Zhang, P., Guo, J., Zhu, X., Guo, L.: UBLF: an upper bound based approach to discover influential nodes in social networks. In: *ICDM*, pp. 907–916 (2013)

10. Pal, S.K., Kundu, S., Murthy, C.D.: Centrality measures, upper bound, and influence maximization in large scale directed social networks. *Fundamenta Informaticae* **130**(3), 317–342 (2014)
11. Wang, F., Jiang, W., Li, X., Wang, G.: Maximizing positive influence spread in online social networks via fluid dynamics. *Future Gener. Comput. Syst.* **86**, 1491–1502 (2018)
12. Bian, R., Koh, Y.S., Dobbie, G., Divoli, A.: Identifying top- $k$  nodes in social networks: a survey. *ACM Comput. Surv.* **52**(1), 1–33 (2019)
13. Dong, L., Yun, J., Jing, Z., Wenjun, W., Guojie, S.: A fast and efficient algorithm for mining top- $k$  nodes in complex networks. *Sci. Rep.* (2017). <https://doi.org/10.1038/srep43330>
14. Fei, L., Mo, H., Deng, Y.: A new method to identify influential nodes based on combining of existing centrality measures. *Mod. Phys. Lett. B* **31**(26), 1–17 (2017)
15. Zhang, B., Zhang, L., Mu, C., Zhao, Q., Song, Q., Hong, X.: A most influential node group discovery method for influence maximization in social networks: a trust-based perspective. *Data Knowl. Eng.* **21**, 71–87 (2019)
16. Sheikahmadi, A., Nematbakhsh, M.A., Zareie, A.: Identification of influential users by neighbors in online social networks. <https://doi.org/10.1016/j.physa.2017.05.098>
17. Li, Y., Fan, J., Wang, Y., Tan, K.-L.: Influence maximization on social graphs: a survey. *IEEE Trans. Knowl. Data Eng.* **30**(10), 1852–1872 (2018)
18. Taninmiş, K., Aras, N., Altinel, I.K.: Influence maximization with deactivation in social networks. *Eur. J. Oper. Res.* **278**(1), 105–119 (2019)
19. Gao, M., Xu, L., Lin, L., Huang, Y., Zhang, X.: Influence maximization based on activity degree in mobile social networks. *Concurr. Comput.: Pract. Exp.* **32**(11), e5677 (2020)
20. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, no. 2, pp. 1942–1948 (1995)
21. Russell, C.E., Shi, Y.: Particle swarm optimization: development, applications and resources. In: *Proceedings of the Congress on Evolutionary*. <https://doi.org/10.1109/CEC.2001.934374>
22. Gong, M., Yan, J., Shen, B., Ma, L., Cai, Q.: Influence maximization in social networks based on discrete particle swarm optimization. *Inf. Sci.* **367**, 600–614 (2016)
23. Yi, Y., Lin, X., Sheng, K., Jiang, L., Dong, W., Cai, Y.: Hybrid ITO algorithm for solving numerical optimization problem. In: Huang, D.-S., Bevilacqua, V., Premaratne, P. (eds.) *ICIC 2014*. LNCS, vol. 8588, pp. 21–31. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-09333-8\\_3](https://doi.org/10.1007/978-3-319-09333-8_3)
24. Dong, W., Zhang, W., Yu, R.: Convergence and runtime analysis of ITÖ algorithm for on class of combinatorial optimization. *Chin. J. Comput.* **34**(4), 636–646 (2011)
25. Wang, Y., Dong, W., Dong, X.: A novel ITÖ algorithm for influence maximization in the large-scale social networks. *Future Gener. Comput. Syst.* **88**, 755–763 (2018)
26. Nogueras, R., Cotta, C.: Self-healing strategies for memetic algorithms in unstable and ephemeral computational environments. *Nat. Comput.* **16**(2), 189–200 (2016). <https://doi.org/10.1007/s11047-016-9560-7>
27. Kayastha, N., Niyato, D., Wang, P., Hossain, E.: Applications, architectures, and protocol design issues for mobile social networks: a survey. *Proc. IEEE* **12**(99), 2130–2158 (2011)

28. Boldrini, C., Conti, M., Passarella, A.: ContentPlace: social-aware data dissemination in opportunistic networks. In: Proceedings of the 11th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems, pp. 203–210 (2008)
29. Xu, J.: Combinatorial Theory in Networks. Science Press, Beijing/China (2013)
30. Dong, W., Hu, Y.: Time series modeling based on ITO algorithm. In: International Conference on Natural Computation, pp. 671–678 (2007)



# Key Nodes Recognition in Opportunistic Network

Zhifei Wang<sup>1,2</sup>, Gang Xu<sup>1,2(✉)</sup>, Fengqi Wei<sup>1,2</sup>, Zhihan Qi<sup>1,2</sup>, and Liqiang He<sup>3</sup>

<sup>1</sup> College of Computer Science, Inner Mongolia University, Hohhot, China  
csxugang@imu.edu.cn

<sup>2</sup> Inner Mongolia A.R. Key Laboratory of Wireless Networking and Mobile Computing, Hohhot, China

<sup>3</sup> Geomechanica Inc., Toronto, Canada

**Abstract.** There are always communication fragmented regions in opportunistic networks, and Ferry nodes which can periodically commute between different fragmented regions always be placed in opportunistic networks. At present, the research on Ferry nodes in opportunistic networks mainly focus on the cache management, energy balance and routing algorithm optimization, meanwhile, researches on identifying Ferry nodes in a strange network are less. On the basis of the importance of structure holes and k-cores, this paper puts forward the index to evaluate the dynamic importance of nodes in opportunistic network, and proposes an importance evaluation algorithm of nodes in opportunistic networks based on it, which is used to identify the Ferry nodes clusters in strange networks. Conclusions can draw through experiments that the proposed model has good applicability and can identify Ferry nodes in networks accurately.

**Keywords:** Opportunistic network · Ferry node recognition · Structural hole · K-Core · Betweenness

## 1 Introduction

Opportunistic network is a kind of self-organized wireless network which does not need a complete link between nodes and it relies on the chances of meeting with each other to transmit data. Since the communication regions are fragmented in the most of time in opportunistic network, Ferry Nodes are placed between the disconnected areas in order to realize the communication between fragmented regions and enhance the overall performance of the opportunistic network. At present, the researches on ferry nodes mainly includes the routing algorithm based on the Ferry nodes [1–8], or how to organize the movement path of Ferry nodes, so as to realize the network coverage of the no signal area by Ferry nodes and improve the communication quality of opportunistic networks [9–14].

In the complex and changeable opportunistic networks, in order to solve the problem that the fixed ferry nodes cannot be applicable for the network coverage, it is necessary to automatically discover and elect ferry nodes, so as to realize the connectivity of the disconnected areas in opportunistic networks, so we proposed a Ferry node recognition model in this paper which can discover and elect ferry nodes automatically.

## 2 Related Work

There have been many researches focusing on key nodes mining of the complex networks and social networks currently, while researches on the discovery and election of Ferry nodes in the opportunistic network is relatively less. In the existing key nodes mining algorithm of complex networks, the researchers mainly used indicators including degree centrality, betweenness centrality, and near-centrality to calculate the importance of nodes in complex networks from different perspectives, but methods suit for complex network is hard to perform well in the opportunistic network with nodes change dynamically.

In reference [15] proposed the algorithm of key nodes mining based on the degree centrality index of nodes, analyzed the number of neighboring nodes but their places in the network were neglected, which makes key nodes have only global importance. In reference [16], the author proposed the betweenness centrality index. It considers the importance of the nodes from the global perspective which can efficiently judge the bridging nodes in the network. But its time complexity was high, and because of the dynamics of the nodes in opportunistic networks, the number of the shortest path to other nodes is also in dynamic changing. Therefore, the betweenness index cannot be applied to the opportunistic network accurately.

Kitsak et al. [17] proposed that the importance of a node depends on its location in the whole network, and they illustrated that the degree centrality and the betweenness centrality cannot describe the importance of nodes accurately. The paper adopted K-Shell to calculate the number of core nodes, and used K-core to describe the propagation ability of nodes. But this method cannot be applied to complex networks with multiple propagation sources. Burt puts forward the Structure Holes Theory [18], and pointed out that nodes with large structure hole is significant to the communication to neighboring nodes in the network. The structure holes theory can calculate the structure relations of multi nodes, and solved the problem of K-core index cannot reflect the relations among neighboring nodes.

In reference [19, 20], the author pointed out that since the importance of nodes in complex networks is influenced by multiple factors, the existing methods based on single importance evaluation index cannot discover key nodes accurately in the network. In reference [21], the author integrated the local features, moving features and global features of nodes to mine key nodes. And based on the method of time slice segmentation, the dynamic topology in opportunistic networks was turned into static topology collection, which offers new methods for mining key nodes.

To sum up, existing key nodes mining methods based on complex networks analyzing cannot be applied to opportunistic network directly. In our algorithm, the running time of the entire opportunistic network was divided into suitable time slices, and the dynamic topology of the opportunistic network in its running time was mapped to each slice according to their sequence so that we can mining the key nodes group in the time slices, thus, elect the Ferry nodes group in the whole lifetime of the network.

The rest of the paper is organized as follows. Section 2 contains two indicators of node importance in opportunistic networks we defined. In Sect. 3, we propose the key nodes recognition model based on the importance of structural holes and K-Core. Section 4 presents a comprehensive set of simulation results for various opportunistic network

scenarios; the results are also analyzed and explained in detail. We then summarize our conclusions in Sect. 5.

### 3 Preliminaries

In this section, we will review some preliminary knowledges, including Structural Holes Theory and K Core Importance Theory.

#### 3.1 Structure Holes

Structure holes is a classic theory of social network brought by Burt [18], which is often used to evaluate the importance of nodes in local networks. If Node B and Node C are both neighbors of Node A, and Node B and C are not adjacent and can only communicate via Node A, there exists a structure hole between Node B and C, or there is a structure hole on Node A. The more structure holes a node possess, the stronger its communicating ability.

**Computing Methods of the Importance of Structure Holes.** Suppose the number of nodes in the network is  $n$ , establish a matrix  $A$  of  $n * n$ , use matrix  $A$  to represent the connection status of nodes in the network,  $a_{ij} = 0$  means that node  $i$  and node  $j$  are disconnected, and  $a_{ij} = 1$  means that node  $i$  and node  $j$  are connected.

Suppose  $k(i)$  is the degree of Node  $i$ ,  $k(i)$  is calculate as Eq. (1) where  $G$  is the set of all the nodes in the topology map.

$$k(i) = \sum_{j \in G} a_{ij} \quad (1)$$

$Q(i)$  –the adjacency degree of node  $i$ , is the sum of degrees of all the neighbors of node  $i$ , as shown in Eq. (2).

$$Q(i) = \sum_{\omega \in r(i)} k(\omega) \quad (2)$$

In Eq. (2),  $r(i)$  is the set of neighbor nodes of node  $i$ .

The network constraint coefficients of nodes are related to multiple factors, such as the connection with others, and the structure between the node and its neighbor. Therefore, the degree and the topology structure of its neighborhood  $P_{ij}$  should be taken into consideration when calculating the network constraint coefficient of nodes.  $P_{ij}$  is calculated as Eq. (3):

$$P_{ij} = \frac{Q(j)}{\sum_{v \in r(i)} Q(v)} \quad (3)$$

The difficulty for a node to form structural holes is represented by the network constraint coefficient of node  $RC_i$ , which is also a measure of the size of a node structure

hole. The network constraint coefficient of a node is inversely proportional to its degree of structural hole, and it is calculated as Eq. (4):

$$RC_i = \sum_{j \in r(i)} (P_{ij} + \sum_q P_{iq}P_{qj})^2 \quad (4)$$

$q$  in Eq. (4) is a node in the intersection of neighbors of nodes  $i$  and  $j$  which is not equal to  $i$  or  $j$ .

The constraint coefficient of the node  $i$  is the structural hole importance index of the node, and the ratio of the sum of the constraint coefficients of all nodes in the network, is calculated as Eq. (5):

$$L_i = \frac{1 - RC_i}{\sum_{j=1}^n (1 - RC_j)} \quad (5)$$

In this paper, an algorithm for calculating the importance of structural holes in opportunistic networks is presented, as shown in Algorithm 1.

---

**Algorithm 1** The calculation method of the importance of Structural Hole

---

**Input:** node set  $G$

**Output:** The Structural hole constraint coefficient of all nodes in  $G$

```

1: function CSHI( $G$ ): //  $G$  is the node set of all nodes in the topology
2:   for  $i \in G$  do:
3:     Calculate  $K(i)$  //Eq.(1)
4:   end for
5:   for  $i \in G$  do:
6:     Calculate  $Q(i)$  //Eq.(2)
7:   end for
8:   for  $i \in G$  do:
9:     for  $j \in r(i)$  do: //  $r(i)$  is the set of all neighbor nodes of node  $i$ 
10:      Calculate  $P_{ij}$  //Eq.(3)
11:      for  $q \in (r(i) \cap r(j))$  do:
12:        Calculate  $P_{iq}$  and  $P_{qj}$  //Eq.(3)
13:      end for
14:      Calculate  $RC_i$  //Eq.(4)
15:    end for
16:  end for
17:  for  $i \in G$  do:
18:    Calculate  $L_i$  //Eq.(5)
19:  end for
20:  return  $L$ 
21: end function

```

---

### 3.2 K-Core Importance

Being a classic in graph theory, K core theory calculates the influence of nodes in the network based on the degree of nodes. The steps of k core decomposition: recursively

delete nodes with a degree of  $k$  or less in the network and assign K-Shell values to the deleted nodes. Repeat the process until all nodes in the network are assigned K-Shell values. In the  $k$  core decomposition algorithm, a large number of nodes are at the same network level, which lead to the incapability of the algorithm when further calculating their node importance.

**Calculation of K-Core Importance.** In the initial stage, record  $k_i^m = k(i)$  for every node, remove the nodes with smallest  $k_i^m$  value from the topology map, and assign  $k_i^m$  to  $K_{s_i}$  of these nodes. Then, update  $k_i^m$  of the remaining nodes through  $k_i^m = k_i^r + \lambda k^e$ , where  $\lambda$  is the adjustment factor, and  $0 \leq \lambda \leq 1$ ,  $k^e$  is the removed degree of previous stage,  $k_i^r$  is the degree of the remaining nodes. Repeat the above process until all the nodes get the  $K_{s_i}$  value. Then  $K_{s_i}$  is the K-Core index of the node  $i$ .

According to the calculation method of  $k$ -core importance, this paper gives the calculation algorithm of K-Core index of nodes in opportunistic network, which is shown in Algorithm 2.

---

**Algorithm 2** The calculation algorithm of K-Core index

---

**Input:** node set  $G$

**Output:** K-Core importance of all nodes in set  $G$

```

1: function CKCI( $G$ ); //  $G$  is the set of all nodes in the topology graph
2:   for  $i \in G$  do:
3:      $k_i^m = k(i)$  //Eq.(1)
4:   end for
5:    $G_n = G$ 
6:   while  $G_n \neq \emptyset$  do:
7:      $k^e = 0$ 
8:     for  $j \in G_n$  do: //  $G_n$  represents the set of remaining nodes in the topology graph
9:       if  $k_i^m < \forall k_q^m$  then: //  $q \in G_n, q \neq j$ 
10:         $k_{s_j} = k_j^m$ 
11:         $k^e = k_j^m$ 
12:         $G_n = G_n \setminus \{j\}$ 
13:       end if
14:     end for
15:     for  $l \in G_n$  do:
16:        $k_l^m = k_l^m + \lambda k^e$ 
17:     end for
18:   end while
19:   return  $K_s$ 
20: end function

```

---

The K-Core importance of node  $i$  refers to the ratio of  $K_{s_i}$  of node  $i$  and the sum of  $K_{s_j}$  of all nodes, which can be calculated as Eq. (6) after the K-Core index of node  $i$  is known:

$$M_i = \frac{K_{s_i}}{\sum_{j=1}^n K_{s_j}} \quad (6)$$



## 4 Key Nodes Recognition Model of Opportunistic Network

Topology of opportunistic network is in dynamically changing. Therefore, in order to solve the problem of key nodes mining in opportunistic networks, we divided the network into several snapshots with equal run time, established static topology of the opportunistic network in the snapshots, mined the key nodes in each snapshots, and determined the Ferry nodes in the opportunistic network based on the frequency which key nodes are selected.

### 4.1 Key Node Recognition Algorithm

The model for evaluating the importance of opportunistic network nodes based on the index proposed in this paper adopted the importance of the K-Core of the node  $M_i$  and the importance of structural holes  $L_i$ , and combines the two indicators as a comprehensive evaluation index of the importance of the node. The larger the value of the index, the more important the node is. We established time slice snapshots when the opportunistic network is running so as to create the time slice topology of the opportunistic network. On the basis of static topology maps, mined key nodes based on our model, and then calculated the number that one node is selected as key node. The selected node is taken as the Ferry node, and the one with highest select number is taken as the most important node in all nodes. The index is calculated as Eq. (7):

$$I_i = \alpha M_i + \beta L_i \quad (7)$$

After several groups of experiment, it is found that the algorithm can achieve better results when  $\alpha = 2$  and  $\beta = 1$ .

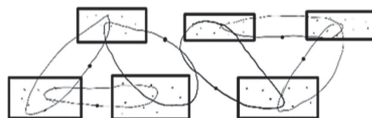
## 5 Experiments

Opportunistic network environment simulator (ONE) is an important experimental simulation platform for opportunistic networks. The paper uses ONE1.4.1 for simulation experiments and the report is ConnectivityDtnsim2Report. The parameter settings of experimental environment are shown in Table 1. This paper sets up two different experimental scenarios, with the betweenness based algorithm by Matteo Riondato et al. [22] (denoted as VC) as comparison. We used the model proposed in this paper and VC respectively to identify Ferry nodes in different scenarios, and analyze the performance of the node identification model in different environments.

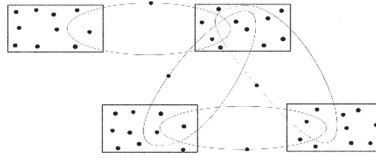
**Table 1.** The parameter of the simulation scenario.

Category	Parameter	Values
Computer configuration	CPU	i7 9700 K
	OS	Windows10 Professional
	RAM	8G
Scenario settings	Simulation area size	200 * 200 m <sup>2</sup>
	Simulation time	24 h(86400 s)
	Message transmission carrier	Bluetooth device
	Message transmission range	50 m
	Nodes movement model in the region	MapRouteMovement (MRM)
	Node movement model between regions	RandomWaypoint (RWP)
	Number of nodes in the region	10
	Nodes moving speed in the region	1 m/s
	Nodes moving speed between regions	5 m/s
dense multi-parallel opportunistic network	Number of experimental regions	6
	Number of nodes between regions	6
sparse multi-parallel opportunistic network	Number of experimental regions	4
	Number of nodes between regions	4

The experimental environments of the dissertation are: the node-intensive opportunity network operation environment and the node-sparse opportunity network operation environment (denote as scenario 1 and scenario 2 below). The simulation environment set up in the paper can cover the actual environment of the opportunistic network, which are shown in the figure below (Figs. 1 and 2).

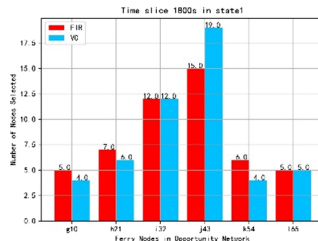


**Fig. 1.** Sketch map of Sparse Dense-parallel Ferry opportunistic network

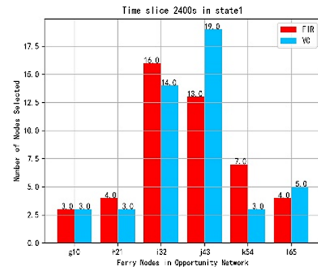


**Fig. 2.** Sketch map of Sparse multi-parallel Ferry opportunistic network

Under the condition that the time slice is 1800 s and 2400 s respectively, our model and VC are respectively used to carry out simulation experiments on the above scenes. The experimental comparison results are shown in Fig. 3, 4, 5 and Fig. 6 below.



**Fig. 3.** Time slice length of 1800 s



**Fig. 4.** Time slice length of 2400 s

Figure 3 to Fig. 4 are the simulation results under scenario 1. From the experimental results, it can be seen that in Scenario 1, the Model proposed in this paper and VC have achieved relatively close results in identifying Ferry nodes. This is because when the nodes distribution in the opportunistic network is dense, the topology between nodes is more stable, and VC Algorithm based on the betweenness centrality can achieve better results. The more the Ferry node choosed, the more important the role in the Ferry node group, the greater the impact on the network.

Figure 5 to Fig. 6 are the comparison results of the simulation experiment of scenario 2. According to the analysis to the results, it can be found that the Model proposed in this paper can accurately identify all Ferry nodes when time slice is set to 1800 s, while VC can only identify part of them. While time slice is 2400 s, both our Model and VC can only identify part of Ferry nodes, but the recognition rate of our Model, 75%, is

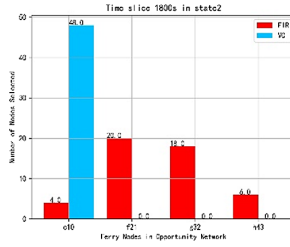


Fig. 5. Time slice length of 1800 s

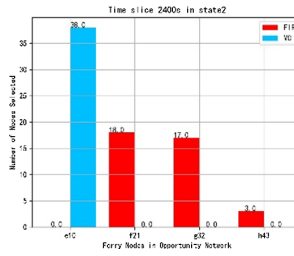


Fig. 6. Time slice length of 2400 s

bigger than if of VC. The analysis can fully prove the effectiveness of proposed Model in opportunistic networks where nodes are sparse.

## 6 Conclusion

Based on the importance of structure holes and k-core, this paper proposed a index to evaluate the dynamic importance of nodes in opportunistic networks, and proposes the Ferry Nodes recognition Model in Opportunistic Network according to the index proposed. Compared with the existing VC algorithm based on betweenness, this model can accurately identify Ferry nodes groups in a variety of application scenarios. The index model we proposed can assess the influence of nodes on the performance of Internet in a certain period of time, and then identify the Ferry nodes in opportunistic networks. Experiments show that the model proposed in this paper can accurately and efficiently identify Ferry nodes in opportunistic networks under low time complexity, and solve the problem of VC algorithm missing Ferry nodes. It provides an important research basis for the automatically select Ferry nodes in opportunistic networks, improving the signal coverage of communication network, and improving the efficiency and quality of the information transmission of opportunistic networks.

**Acknowledgment.** The authors wish to thank Natural Science Foundation of China under Grant NO. 61841109, 61662054, Natural Science Foundation of Inner Mongolia under Grand NO. 2019MS06031.

## References

1. Chen, P., et al.: All coverage and low-delay routing algorithm based on message ferry in opportunistic networks. *Appl. Res. Comput.* **34**(03), 819–823 (2017)
2. Tao, C., Gao, J.: Modeling mobile application test platform and environment: testing criteria and complexity analysis. In: *Proceedings of the 2014 Workshop on Joining AcadeMiA and Industry Contributions to Test Automation* (2014)
3. Zhang, T., Gao, J., Cheng, J., Uehara, T.: Compatibility testing service for mobile applications. In: *2015 IEEE Symposium on Service-Oriented System Engineering*, pp. 179–186 (2015)
4. Li, S., Qu, W., Liu, C., Qiu, T., Zhao, Z.: Survey on high reliability wireless communication for underwater sensor networks. *J. Netw. Comput. Appl.* **148**, 102446 (2019)
5. Ikenoue, K., Ueda, K.: Routing method based on data transfer path in DTN environments. In: Barolli, L., Hellinckx, P., Enokido, T. (eds.) *Advances on Broad-Band Wireless Computing, Communication and Applications*, pp. 544–552. Springer, Cham. [https://doi.org/10.1007/978-3-030-33506-9\\_49](https://doi.org/10.1007/978-3-030-33506-9_49)
6. Chen, W., Chen, Z., Li, W., Zeng, F.: An enhanced community-based routing with ferry in opportunistic networks. In: *2016 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI)*. pp. 340–344 (2016)
7. Krug, S., Helbig, M., Seitz, J.: Poster: Utilization of additional nodes in hybrid DTN-manet scenarios. In: *Proceedings of the 12th Workshop on Challenged Networks*, pp. 35–37 (2017)
8. Vallikannu, R., George, A., Srivatsa, S.K.: Routing and charging scheme with ferry nodes in mobile adhoc networks. In: *2017 International Conference on Intelligent Computing and Control (I2C2)*, pp. 1–4 (2017)
9. Anguswamy, R., Thiagarajan, M., Dagli, C.H.: Systems methodology and framework for problem definition in mobile ad hoc networks. In: *2008 2nd Annual IEEE Systems Conference*, pp. 1–7 (2008)
10. Wang, T., Low, C.P.: Reducing message delay with the general message ferry route (MFR\*) problem. In: *2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 380–387 (2011)
11. Ali, A.S., Mahmoud, K.R., Naguib, K.M.: Optimal caching policy for wireless content delivery in d2d networks. *J. Network Comput. Appl.* **150**, 102467 (2020)
12. Ahmed, K.K.: A mobile agent and message ferry mechanism based routing for delay tolerant network. Thesis (2018)
13. Hu, C., Lin, H., Hsu, Y., Huang, S., Hui, L., Zhang, Z.: Message forwarding with ferries in delay-tolerant networks. In: *2019 28th Wireless and Optical Communications Conference (WOCC)*, pp. 1–5 (2019)
14. Diwaker, C., et al.: An enhanced cluster based movement model using multiple ferries nodes in vanet. *Int. J. Manage. IT Eng.* **6**(10), 68–78 (2016)
15. Zhaolong, H., Jianguo, L., Zhuoming, R.: Analysis of voluntary vaccination model based on the node degree information. *Acta Physica Sinica* **62**(21), 512–517 (2013)
16. Freeman, L.C.: A set of measures of centrality based on betweenness. *Sociometry* **40**(1), 35–41 (1977)
17. Kitsak, M., Gallos, L.K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H.E., Makse, H.A.: Identification of influential spreaders in complex networks. *Nat. Phys.* **6**(11), 888–893 (2010)
18. Burt, R.S., Kilduff, M., Tasselli, S.: Social network analysis: foundations and frontiers on advantage. *Annual Rev. Psychol.* **64**(1), 527–547 (2013)
19. Yu, H., Cao, X., Liu, Z., Li, Y.: Identifying key nodes based on improved structural holes in complex networks. *Physica A: Stat. Mech. Appl.* **486**, 318–327 (2017)
20. Gang, H., Hao, G., Xiang, X.: Identify important nodes in complex network based on aggregation of multi-attribute preference information. *J. Zhejiang Sci-Tech Univ.* **41**(04), 482–488 (2019)

21. Shu, J., et al.: Cartical nodes evaluation of opportunistic networks based on topological condensation graph. *J. Beijing Univ. Posts Telecommun.* **42** (02), 57–62 (2019)
22. Riondato, M., Kornaropoulos, E.M.: Fast approximation of betweenness centrality through sampling. *Data Min. Knowl. Disc.* **30**(2), 438–475 (2015). <https://doi.org/10.1007/s10618-015-0423-0>



# A Novel Measure to Quantify the Robustness of Social Network Under the Virus Attacks

Bo Song<sup>1,2,3,4</sup>(✉), Zhengjun Jing<sup>5</sup>, Y. Jay Guo<sup>2</sup>, Ren Ping Liu<sup>2</sup>, and Qian Zhou<sup>1,3,6</sup>

<sup>1</sup> School of Modern Posts and Institute of Modern Posts,  
Nanjing University of Posts and Telecommunications, Nanjing 210003,  
People's Republic of China  
songbo@njupt.edu.cn

<sup>2</sup> University of Technology Sydney, Sydney, NSW 2007, Australia

<sup>3</sup> Post Big Data Technology and Application Engineering Research Center of Jiangsu Province,  
Nanjing University of Posts and Telecommunications, Nanjing 210003,  
People's Republic of China

<sup>4</sup> Post Industry Technology Research and Development Center of the State Posts  
Bureau (Internet of Things Technology), Nanjing University of Posts and Telecommunications,  
Nanjing 210003, People's Republic of China

<sup>5</sup> Jiangsu University of Technology, Changzhou 213001, People's Republic of China

<sup>6</sup> State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093,  
People's Republic of China

**Abstract.** Combining the spread velocity, the epidemic threshold and the infection scale at steady state, a new network robust measure with respect to the virus attacks is proposed in this paper. Through examples, we show that spread velocity plays an important role on the network robustness. By using the SI and SIS epidemic model, we analyze the robustness of homogeneous networks. The results show that the irregularity in node degrees decreases the robustness of the networks. Moreover, the simulation results show that the network becomes more fragile as the average degree grows in both homogeneous and heterogeneous networks.

**Keywords:** Social network · Network robustness · Virus spread · Spreading velocity

## 1 Introduction

As one of the typical scenarios of attacks, epidemic spreading in social network has a huge impact on our daily activities. At present, the COVID-19 has spread all over the world within six months [1–4]. Based on the Situation Report-133 of World Health Organization (WHO) [5], the data as received by WHO from national authorities by 10:00 CEST, 01 June 2020 has shown that there are 6057853 cases confirmed as infected globally and 371166 people dead. The robustness of complex networks against virus attacks has become one of the most concerned topics in complex network study [6–11].

The study on epidemics has a very long history and classic epidemic models are built to describe the virus spreading, such as the susceptible-infected (SI) model, susceptible-infected-susceptible (SIS) model, the susceptible-infected-recovered (SIR) model [8].

Besides the epidemics, a lot of other cyberattacks exist in almost any kind of networks, for example, the DDoS attacks on the Internet [12, 13], the cascading failures in the power grids [14] and the epidemic spread in social networks [8]. As a malicious and deliberate attempt by an individual (organization) to breach the information system of another individual (organization), the attacks spread from one node to another in the network. Once one or some components are attacked, it may cause incalculable losses to the entire network and other related networks. Recently, the epidemic models have been adapted to study the spread of the above cyberattacks.

Robustness of the network refers to the ability of the network to maintain a certain degree of structural integrity and function after being subjected to a fault or attack [15]. Epidemic spreading models have been used to study the network robustness with respect to the virus attacks. Epidemic threshold is the most commonly used measure for the network robustness with respect to virus attack, i.e., the larger the epidemic threshold, the more robust a network is against the spread [11, 16]. Recently, Mina Youssef [9–11] proposed a new measure to assess the robustness of complex networks with respect to the spread of SIS epidemics. The results shown that the proposed measure of network robustness with respect to the virus attacks is effective for the epidemics with different final infection scales.

Since the epidemic threshold and the final infection scale are considered to measure the network robustness, the spread velocity [17–19], as another important indicator describing the epidemics, should also be taken into account to measure network robustness. On the one hand, many epidemics will eventually achieve network-wide infection or immunity, such as the epidemic process described by SI or SIR model. In this case, the final infection scale of the epidemics in different networks are the same. On the other hand, network structures show great impact on the spread velocity, i.e., the virus spread velocity differs in different networks. In addition, the trends of spread velocity, epidemic threshold and the final infection scale in the network are different. Therefore, the spread velocity is one of the key factors that cannot be ignored to measure the network robustness with respect to the epidemics.

In this paper, we propose a novel metric, combining the spreading velocity, the infected population and the epidemic threshold, to measure the robustness with respect to the virus attacks in social networks. First, we show some examples of networks where the epidemic threshold and/or infection scale fail to assess their robustness. Then the new metric is introduced, based on which the network robustness with respect to virus spreading are analyzed. The simulation results show that as the average degree grows in both homogeneous and heterogeneous networks, the network becomes more vulnerable to the attacks. Moreover, in homogeneous networks, the network robustness increases because of irregular connection.

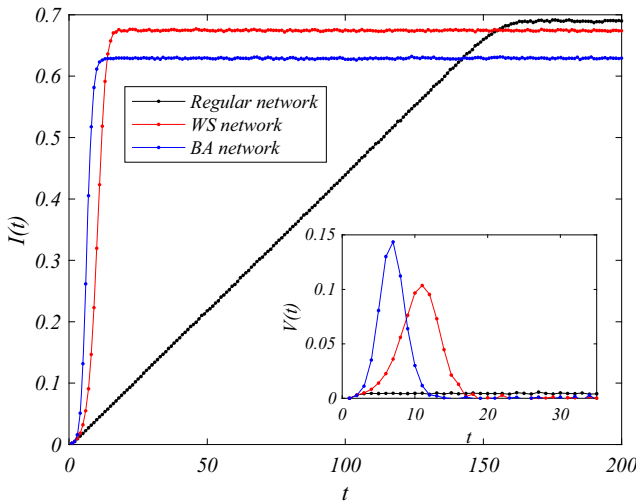
The rest of this paper is organized as follows. In Sect. 2, we analyze the necessity of putting forward the new metric, and then introduce the novel metric to quantify the network robustness based on epidemic spread in Sect. 3. We present the simulation results in Sect. 4, and the main conclusions and future work are summarized in Sect. 5.



## 2 The Network Robustness with Respect to Epidemic Spread

Epidemic threshold is the most commonly used measure for the network robustness with respect to the spread of epidemic, i.e., the larger the epidemic threshold, the more robust a network is against the spread. The existed literatures shown that large BA networks [21] consequently are more vulnerable to spread of epidemics than WS networks [20] considering the epidemic threshold. Then the researchers found that the epidemic threshold may fails to assess the network robustness, a new metric to quantify the robustness of networks considering both the epidemic threshold and fraction of infection at steady state was proposed in SIS epidemic model [10].

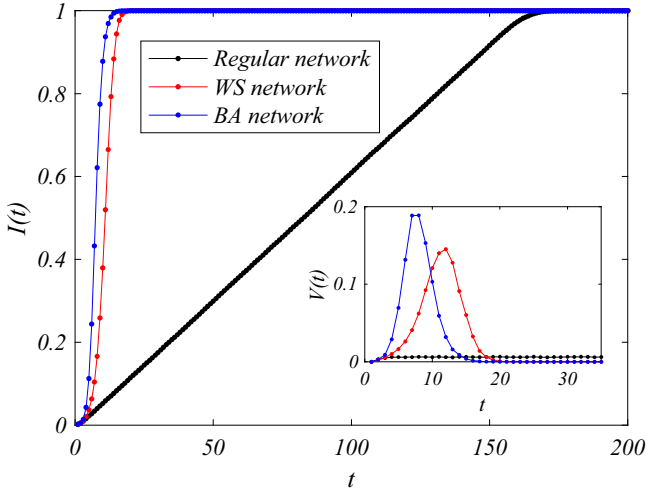
In fact, it is not comprehensive to use epidemic threshold and/or infection scale to measure network robustness with respect to the virus attacks. For example, Fig. 1 shows the SIS epidemic spreading process in 3 different networks. We observe that the final density of the infection nodes of BA network is smaller than the final density of the infection nodes in the WS network and Regular network, i.e.,  $I_{Regular} > I_{WS} > I_{BA}$ . From the perspective of the infection scale of the steady state, BA network is more robustness than WS network and Regular network. However, comparison of the spread velocities shows that the spread velocity of BA network is the fastest, and the spread velocity of regular network is much slower than that of both BA network and WS network, i.e.,  $V(t)_{BA} > V(t)_{WS} > V(t)_{Regular}$ . That is to say, one single indicator, for example, the fraction of infection at steady state or the spread velocity, cannot accurately measure the network robustness with respect to the virus attack.



**Fig. 1.** The SIS epidemic spreading process in different networks ( $\beta=\gamma = 0.3$ ).

And when the final infection densities at the steady state are the same, such as in the SI model shown in Fig. 2, we can see that the same results of the spread velocity of different networks in SIS epidemic model, i.e.,  $V(t)_{BA} > V(t)_{WS} > V(t)_{Regular}$ . As

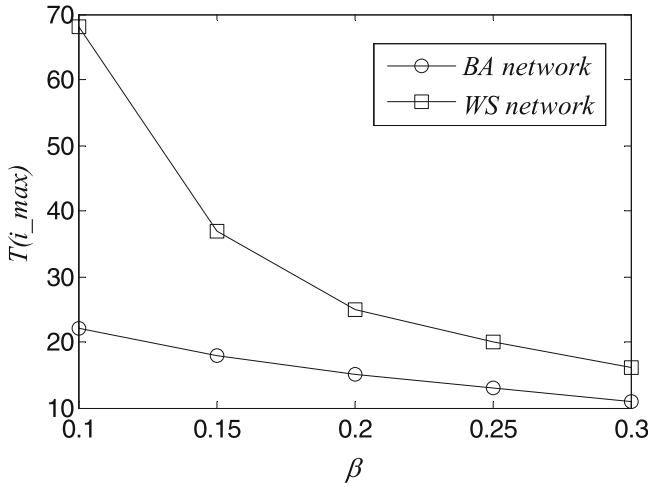
we all know, the epidemic threshold in the homogeneous network is larger than in the heterogeneous network. Therefore, even in the epidemic models with the same infection scale at the steady state, one single indicator is not enough to measure the network robustness precisely.



**Fig. 2.** The SI epidemic spreading process in different networks ( $\beta = 0.3$ ) (Color figure online).

Moreover, due to the difference of the spread velocity, the time to reach a stable state of epidemic spread is different. We did simulations on WS network and BA network, Fig. 3 counts the time points when the epidemics reach steady state ( $T(i_{max})$ ) under different infect rates in BA and WS networks. We can see more intuitively from Fig. 3 that the time points of reaching the steady states are different in the 2 networks, and as the infection rate decreases, this gap becomes more and more obvious. Therefore, under the condition of low infection probability, the difference of spread velocity in the networks is very large.

It can be seen from the above analysis that a single indicator (epidemic threshold, fraction of infection at steady state or spreading velocity) may fail to measure the robustness of the network. Comparing the propagation processes in the three different networks, we can conclude that the spread velocity is the fastest in the BA network, but the infection scale is the smallest. In the WS network and the regular network, although the spread velocity is slower, the infection scale is larger. Therefore, it is necessary to propose a network robustness measurement considering of multiple indicators with respect to epidemic spread.



**Fig. 3.** The time of reaching the steady states of the epidemic spreading in different networks.

### 3 The Novel Metric to Quantify the Network Robustness Under the Virus Attacks

Combining the epidemic threshold, propagation scale and spreading velocity together, we propose a multiple-indicator-based measurement to quantify the network robustness with respect to virus attacks. Supposing that in the SI/SIS epidemic model, the rate of a susceptible node being infected by a single infected neighbor is  $\beta$ , and the infected node recovered with the rate  $\delta$  in the SIS model. For the sake of simplicity, we set  $\delta=1$  in the rest of the paper. The effective infection rate is defined as  $\tau=\beta/\delta=\beta$ , and the density of infected nodes at time  $t$  is described as  $i(t)$ . Then the steady state of the infection under the effective infection rate  $\tau$  can be written as

$$i_{\infty}(\tau) = \lim_{t \rightarrow \infty} i_t(\tau),$$

we first define the cumulative infection  $C_{i_t}(\tau)$  as the sum of infection density at each time slot under the effective infection rate  $\tau$ ,

$$C_{i_t}(\tau) = \sum_{t'=0}^t i_{t'}(\tau).$$

Taking into account of all values of  $\tau$ , supposing that  $t_s$  is the first time the network reaches the stable state, the new robustness measure with respect to virus attack,  $R_{VA}$ , can be written as

$$R_{VA} = \frac{1}{t} \int_{\tau_s}^{\infty} C_{i_t}(\tau) d\tau = \frac{1}{t} \int_{\tau_s}^{\infty} \sum_{t'=0}^t i_{t'}(\tau) d\tau, \quad t \leq t_s, \tag{2.1}$$

where  $\tau_s$  is the epidemic threshold.

Introducing the effective cure rate  $s = 1/\tau$  in 10, (2.1) can be rewritten as

$$R_{VA} = \frac{1}{t} \int_0^\rho C_{i_t}(s) ds = \frac{1}{t} \int_0^\rho \sum_{t'=0}^t i_{t'}(s) ds, \quad t \leq t_s. \tag{2.2}$$

When  $t = t_s$ , the robustness of network  $G$  can be written as

$$R_{VA}^G = \frac{1}{t_s} \int_0^\rho C_{i_{t_s}}(s) ds = \frac{1}{t_s} \int_0^\rho \sum_{t'=0}^{t_s} i_{t'}(s) ds. \tag{2.3}$$

In (2.3), the length of  $t_s$  represents the spreading velocity, and  $i_t(s)$  represents the density of infected nodes at time  $t$ . Then the proposed  $R_{VA}^G$  considering of epidemic threshold, the infection scale and the spread velocity. Since  $t_s$  changes under different effective cure rate, it is hard to make statistics of  $t_s$ . In order to avoid the impact of the density of stable infection within the time period  $[t_s, t_\infty]$  on the network robustness, we use the  $i_{max}(s) - i_t(s)$  instead of  $i_t(s)$  in (2.3). Therefore, the network robustness at  $t$ -time can be written as

$$R_{VA}(t) = \frac{1}{t} \int_0^\rho \sum_{t'=0}^t (i_{max}(s) - i_{t'}(s)) ds, \tag{2.4}$$

the robustness of network  $G$  can be written as

$$R_{VA}^G = \frac{1}{t_s} \int_0^\rho \sum_{t'=0}^{t_s} (i_{max}(s) - i_{t'}(s)) ds. \tag{2.5}$$

We can see from (2.4) and (2.5) that, when  $t < t_s$ , the network robustness at  $t$ -time,  $R_{VA}(t)$ , depends on the epidemic threshold and the infection scale at time  $t'$ ,  $t' \in [0, t]$ . While the robustness of network  $G$  depends on the length of  $t_s$ , besides the epidemic threshold and the infection scale at time  $t'$ ,  $t' \in [0, t_s]$ . Therefore, the shorter time it takes to reach steady state and/or the larger the infection scale at time  $t'$ , the larger the  $R_{VA}^G$  is, and the more vulnerable of the network under the virus attack, accordingly.

We choose two epidemic models to analyze the network robustness, one is the epidemic process that the whole network is infected finally, i.e., the SI epidemic model, the other one is that the infection density is stable at a non-1 value, i.e., the SIS epidemic model.

**Case 1. The robustness of homogeneous network with respect of SI epidemic spreading.** The state of each node in the SI model is infected or healthy, and the change of infected individuals over time can be described as

$$\frac{di}{dt} = \beta \langle k \rangle i(1 - i). \tag{2.6}$$

By separating variables, (2.6) can be written as

$$\frac{di}{i(1 - i)} = \beta \langle k \rangle dt, \tag{2.7}$$

integrating both sides of (2.7), we can obtain

$$\ln \frac{1 - i(t)}{i(t)} = -\beta \langle k \rangle t + c.$$

The density of the infected nodes at time  $t$  can be written as

$$i(t) = \frac{1}{1 + (1/i_0 - 1)e^{-\beta \langle k \rangle t}}. \tag{2.8}$$

The final density of the infection of SI model equals to 1, i.e.,  $i_\infty = 1$ . Based on (2.8), the robustness of homogeneous network  $G$  with respect of SI epidemic spreading can be written as

$$\begin{aligned} R_{VA}^{SI} &= \frac{1}{t_s} \int_{\tau_s}^{\infty} \sum_{t'=0}^{t_s} (i_\infty(\tau) - i_t(\tau)) d\tau \\ &= \frac{1}{t_s} \int_{\tau_s}^{\infty} \sum_{t'=0}^{t_s} \left(1 - \frac{1}{1 + (1/i_0 - 1)e^{-\beta \langle k \rangle t'}}\right) d\tau. \end{aligned} \tag{2.9}$$

**Case 2. The robustness of homogeneous network with respect of SIS epidemic spreading.** Ignoring the degree correlations of nodes in homogeneous network, the density of infected nodes at time  $t$ , i.e.,  $i(t)$ , satisfies (2.10)

$$\frac{di}{dt} = -i + \beta \langle k \rangle i(1 - i). \tag{2.10}$$

Integrating both sides of (2.10),

$$\int_0^t dt = \int_{i_0}^{i(t)} \frac{1}{-i + \beta \langle k \rangle i(1 - i)} di, \tag{2.11}$$

then (2.11) can be rewritten as

$$t = \frac{1}{\beta \langle k \rangle - 1} \int_{i_0}^{i(t)} \frac{1}{i} di + \frac{\beta \langle k \rangle}{\beta \langle k \rangle - 1} \int_{i_0}^{i(t)} \frac{1}{\beta \langle k \rangle - \beta \langle k \rangle i - 1} di, \tag{2.12}$$

we can obtain that

$$\begin{aligned} e^{(\beta \langle k \rangle - 1)t} &= \frac{i(t)}{\beta \langle k \rangle - \beta \langle k \rangle i(t) - 1} \bigg/ \frac{i_0}{\beta \langle k \rangle - \beta \langle k \rangle i_0 - 1}, \\ \frac{i(t)}{\beta \langle k \rangle - \beta \langle k \rangle i(t) - 1} &= \frac{i_0 e^{(\beta \langle k \rangle - 1)t}}{\beta \langle k \rangle - \beta \langle k \rangle i_0 - 1}, \end{aligned}$$

$$i(t)(\beta \langle k \rangle - \beta \langle k \rangle i_0 - 1) = i_0 e^{(\beta \langle k \rangle - 1)t} (\beta \langle k \rangle - \beta \langle k \rangle i(t) - 1).$$

The density of the infected nodes at time  $t$  can be written as

$$i(t) = \frac{(\beta \langle k \rangle - 1) i_0 e^{(\beta \langle k \rangle - 1)t}}{\beta \langle k \rangle - \beta \langle k \rangle i_0 - 1 + i_0 \beta \langle k \rangle e^{(\beta \langle k \rangle - 1)t}}. \tag{2.13}$$

Let (2.10) equals to 0, we can get

$$\frac{di}{dt} = -i + \beta \langle k \rangle i(1 - i) = 0,$$

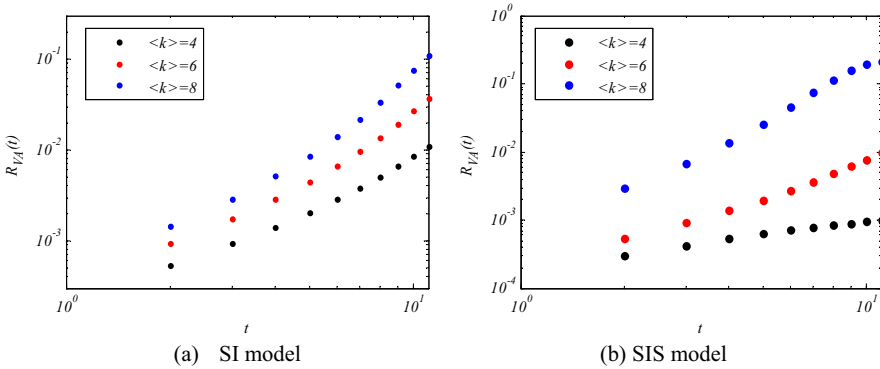
when  $\tau = \frac{\beta}{\delta} = \beta > \tau_c$ , the infection density of the final stable state is

$$i_\infty = 1 - \frac{1}{\beta \langle k \rangle}. \tag{2.14}$$

Based on (2.13) and (2.14), the robustness of homogeneous network  $G$  with respect of SIS epidemic spreading can be written as

$$\begin{aligned} R_{VA}^{SIS} &= \frac{1}{t_s} \int_{\tau_s}^{\infty} \sum_{t'=0}^{t_s} (i_\infty(\tau) - i_t(\tau)) d\tau \\ &= \frac{1}{t_s} \int_{\tau_s}^{\infty} \sum_{t'=0}^{t_s} \left( 1 - \frac{1}{\beta \langle k \rangle} - \frac{(\tau \langle k \rangle - 1) i_0 e^{(\tau \langle k \rangle - 1)t'}}{\tau \langle k \rangle - \tau \langle k \rangle i_0 - 1 + i_0 \tau \langle k \rangle e^{(\tau \langle k \rangle - 1)t'}} \right) d\tau \end{aligned} \tag{2.15}$$

We present the numerical solutions of (2.9) and (2.13) in Fig. 4 and Table 1. We can see that as the average degree of the network grows, the network becomes more vulnerable to the virus attack. Figure 4 shows the robustness at  $t$ -time ( $t < t_s$ ) in homogeneous network with respect to SI and SIS epidemic spreading.



**Fig. 4.** The robustness of homogeneous networks at  $t$ -time with respect to SI epidemic spreading (Fig. 4(a)) and SIS epidemic spreading (Fig. 4(b)). ( $\beta = 0.2, \delta = 1$ )

**Table 1.** The robustness of homogeneous network  $G$  with respect to SI and SIS epidemic spreading.

Network Robustness	$\langle k \rangle = 4$	$\langle k \rangle = 6$	$\langle k \rangle = 8$
$R_{VA}^{SI}$	0.1454	0.2798	0.3502
$R_{VA}^{SIS}$	0.1261	0.2106	0.2777

## 4 Simulations

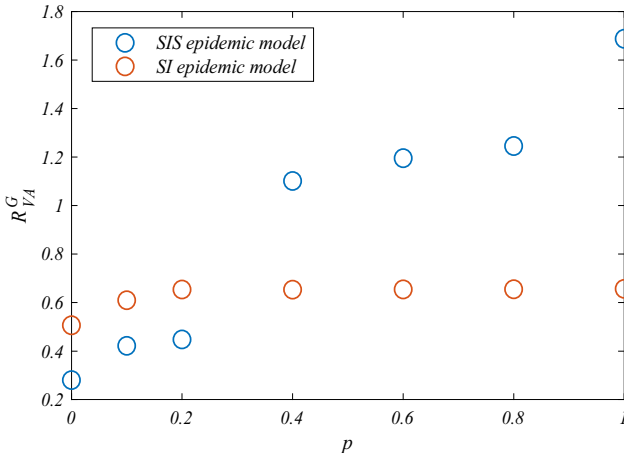
In this section, Monte-Carlo simulations are used to further explore the robustness of different networks with respect to the virus attacks. Simulations are carried out in different networks with  $N = 500$  nodes. All the simulations are averaged over 500 runs.

First, the simulations are carried out in WS small-world networks with the rewiring rate  $p$ . Based on the construction algorithm of the WS model, at the beginning, the network is a regular graph, and then randomly reconnects each edge in the network with probability  $p$ , that is, one endpoint of the edge remains unchanged, and the other endpoint is taken as the network. In the above model,  $p = 0$  corresponds to a completely regular network,  $p = 1$  corresponds to a completely random network, and the transition from a completely regular network to a completely random network can be controlled by adjusting the  $p$  value. We capture a set of networks where the rewiring rate  $p$  changes from 0 to 1 and analyze the robustness of these networks. The examples of networks are of the same average degree, i.e.,  $\langle k \rangle = 6$ , and almost have the same epidemic thresholds, in which the epidemic threshold hardly works on measuring the network robustness.

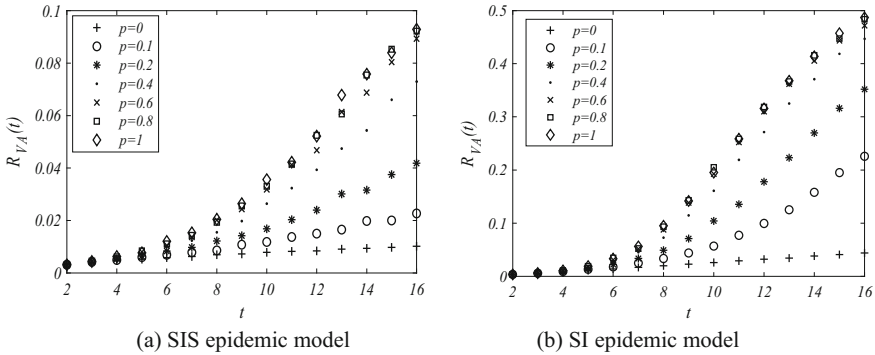
Figure 5 shows the network robustness  $R_{VA}^G$  of different networks. In SIS epidemic model (red circle), we can see that as  $p$  grows,  $R_{VA}$  becomes larger, that is, the network becomes more vulnerable. When  $p = 0$ , the network has a small value of  $R_{VA}$ , that is because in regular network ( $p = 0$ ), the virus spreads very slowly, as shown in Fig. 2, that is to say, the spread velocity plays a greater role on measuring the network robustness than the steady infection. In SI model (blue circle), both the epidemic threshold and the steady infection are the same, then our robustness measure of network is only related to the spread velocity. We can see that from Fig. 5 that the robustness of regular network is smaller than other networks due to the slow spread velocity. However, the WS networks and random network are almost have the same robustness as the difference of spread velocity is small in these networks.

We further count the network robustness at time  $t$  ( $t < t_s$ ), the result in Fig. 6 shows a better robustness at time  $t$  in regular network. As  $p$  increases, the robustness of network at time  $t$  increases, i.e., the network becomes more fragile.

The simulations are also carried out in BA networks with different average degree  $\langle k \rangle$ , we can see from Table 2 and Fig. 7 that BA network becomes more vulnerable to the virus attacks as the average degree of the network grows. In addition, compared with the results in Table 1, BA networks shows stronger robustness than the homogeneous networks with the same average degree. Although the irregularity in node degrees decreases the robustness of the WS networks, as shown in Fig. 5 and Fig. 6, we can



**Fig. 5.** The robustness of WS networks with respect of SIS/SI epidemic model. (Color figure online)



**Fig. 6.** The robustness of WS networks at  $t$ -time with respect to SI epidemic spreading (Fig. 4(a)) and SIS epidemic spreading (Fig. 4(b)).  $\beta = 0.25$

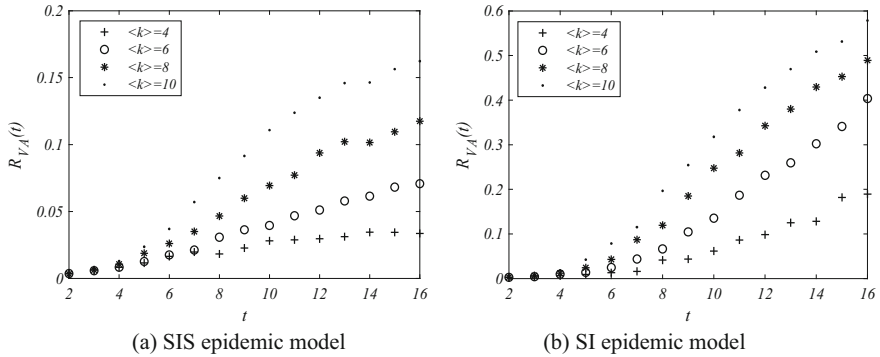
still conclude that the robustness of BA network is not always weaker than that of WS network based on our robustness measurement.

**Table 2.** The robustness of BA network with respect to SI and SIS epidemic spreading.

Network Robustness	$\langle k \rangle = 4$	$\langle k \rangle = 6$	$\langle k \rangle = 8$	$\langle k \rangle = 10$
$R_{VA}^{SI}$	0.5589	0.5594	0.6035	0.7053
$R_{VA}^{SIS}$	0.0101	0.0115	0.1223	0.1643

Based on the measurement proposed in this paper, the simulation results show that in both homogeneous and heterogeneous networks, the network becomes more vulnerable





**Fig. 7.** The robustness of BA networks at  $t$ -time ( $t < t_c$ ) with respect to SI epidemic spreading (Fig. 4(a)) and SIS epidemic spreading (Fig. 4(b)).  $\beta = 0.15$

as the average degree grows. In homogeneous networks, the robustness of networks with respect to virus spreading decreases as  $p$  increases, i.e., the irregularity in node degrees makes the network vulnerable.

### 5 Conclusion

Considering the spread velocity, the epidemic threshold and steady infection, a new robustness measure with respect to virus attacks in social networks is proposed. Simulation results show that the network becomes more vulnerable to the virus attacks as the average degree of the network grows in both homogeneous networks and heterogeneous networks. Our new measure confirms that the irregularity in node degrees decreases the robustness of the homogeneous networks. In the future work, we will focus on the heuristic to compute the  $R_{VA}$  for large networks with high accuracy. Moreover, the impact of other characteristics of network, e.g., clustering coefficient, weight distribution, on the network robustness will also be analyzed.

### References

1. Lancet, T.: The gendered dimensions of COVID-19. *The Lancet* **395**(10231), 1168 (2020)
2. Dave, M., Seoudi, N., Coulthard, P.: Urgent dental care for patients during the COVID-19 pandemic. *The Lancet* **395**(10232), 1257 (2020)
3. Cameron, H., Brian, O., Nicholas, S., et al.: Will COVID-19 fiscal recovery packages accelerate or retard progress on climate change? *Oxford Review of Economic Policy* (2020)
4. Odeh, N.D., Babkair, H., Abu-Hammad, S., et al.: COVID-19: present and future challenges for dental practice. *Int. J. Environ. Res. Public Health* **19**(9), 3151 (2020)
5. [https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200601-covid-19-sitrep-133.pdf?sfvrsn=9a56f2ac\\_4](https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200601-covid-19-sitrep-133.pdf?sfvrsn=9a56f2ac_4)
6. Wesley, C., Mata Angélica, S., Ferreira, S.C.: Robustness and fragility of the susceptible-infected-susceptible epidemic models on complex networks. *Phys. Rev. E* **98**(1), 012310 (2018)

7. Jiang, Y., Hu, A., Huang, J.: Importance-based entropy measures of complex networks' robustness to attacks. *Cluster Comput.* **22**, 3981–3988 (2018)
8. Pastor-Satorras, R., Castellano, C., Mieghem, P.V., et al.: Epidemic processes in complex networks. *Rev. Modern Phys.* **87**(3), 120–131 (2014)
9. Mieghem, P.V.: The viral conductance of a network. *Comput. Commun.* **35**(12), 1494–1506 (2012)
10. Mina, Y., Robert, K., Caterina, S.: Viral conductance: quantifying the robustness of networks with respect to spread of epidemics. *J. Computat. Sci.*, **2**(2011), 286–298 (2011)
11. Socievole, A., De Rango, F., Scoglio, C., et al.: Assessing network robustness under SIS epidemics: the relationship between epidemic threshold and viral conductance. *Comput. Netw.* **103**, 196–206 (2016)
12. Zargar, S.T., Joshi, J., Tipper, D.: A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Commun. Surv. Tutorials* **15**(4), 2046–2069 (2013)
13. Song, B., Wang, X., Ni, W., et al.: Reliability analysis of large-scale adaptive weighted networks. *IEEE Trans. Inf. Forensics Security* **15**, 651–665 (2019)
14. Song, B., Zhang, Z.H., Song, Y.R., et al.: Preferential redistribution in cascading failure by considering local real-time information. *Physica A: Statist. Mech. Appl.* **532**, 121729 (2019)
15. Guillaume, D., Diana, A., Alexander, D., et al.: Molecular mechanisms of paralogous compensation and the robustness of cellular networks. *J. Experimental Zoology Part B Molecular Dev. Evol.* **322**(7), 488–499 (2014)
16. Banerjee, S., Chatterjee, A., Shakkottai, S.: Epidemic thresholds with external agents. In: *Proceedings IEEE Infocom* (2013)
17. Zhu, G.H., Chen, G.R., Zhang, H.F., et al.: Propagation dynamics of an epidemic model with infective media connecting two separated networks of populations. *Commun. Nonlinear Sci. Numerical Simulat.* **20**(1), 240–249 (2015)
18. Perasso, A.: Global stability and uniform persistence for an infection load-structured SI model with exponential growth velocity. *Commun. Pure Appl. Anal.* **18**(1), 15–32 (2019)
19. Li, C.C., Jiang, G.P., Song, Y.R., et al.: Modeling and analysis of epidemic spreading on community networks with heterogeneity. *J. Parallel Distributed Comput.* **119**, 136–145 (2018)
20. Watts, D.J., Strogatz, S.H.: Collective dynamics of “small-world” networks. *Nature* **393**, 440–442 (1998)
21. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. *Science* **286**(5439), 509–512 (1999)

# **Privacy-Preserving and Security**



# Effectiveness Analysis of Traditional Chinese Medicine for Anti-Alzheimer's Disease Based on Machine Learning

Jingwen Lu<sup>1</sup>, Peng Tang<sup>1</sup>, Weidong Qiu<sup>1</sup>(✉), Hao Wang<sup>2</sup>, and Jie Guo<sup>1</sup>

<sup>1</sup> School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai, China  
qiuwd@sjtu.edu.cn

<sup>2</sup> Department of Pharmacology and Chemical Biology, School of Medicine,  
Shanghai Jiao Tong University, Shanghai, China

**Abstract.** Alzheimer's disease is a major disease that endangers people's health. Its occurrence and development involve complex biological systems in the human body. Traditional Chinese medicine pays attention to comprehensive regulation and control, which is in line with complex biological systems. With thousands of years of history, some classic pharmaceutical formulas and prescriptions have been formed. Through in-depth mining of Chinese medicine data for treating Alzheimer's disease, we extract the drug attributes of traditional Chinese medicine prescriptions for treating Alzheimer's disease, and using machine learning methods and deep learning methods to model and analyze the properties of medicines, at the same time mining effective drug attributes. The model can also predict the effectiveness of new Chinese medicine prescriptions for treating Alzheimer's disease. In this paper, different machine learning algorithms are used to model the drug properties in traditional Chinese medicine prescriptions. The highest accuracy of the model can reach more than 62%. The experimental results show that the method proposed has certain research value and prospect in the study of traditional Chinese medicine.

**Keywords:** Alzheimer's disease · Traditional chinese medicine · Machine learning

## 1 Introduction

Alzheimer's disease(AD) is a neurodegenerative disease characterized by cognitive and memory dysfunction. According to World Health Organization (WHO) statistics, there are currently nearly 50 million dementia patients worldwide, and the global cost of treatment is about \$100 billion per year. The number of this patient population may reach 150 million by 2050, and the cost of treatment will reach 200 billion US dollars per year by 2030. Therefore, AD has become the third major disease that seriously threatens the health of the elderly after cardiovascular diseases and malignant tumors, and its research has been a hot spot in the international medical community.

Traditional Chinese medicine (TCM) has a long history in treating AD and has accumulated a lot of treatment experience. Although there is no record of “Alzheimer’s disease” in ancient medical books, there are records of “forgetfulness”, “idiot”, “dementia” and other symptoms. TCM believes that dementia is caused by marrow depletion and brain elimination, and can be divided into a variety of syndrome types such as insufficient marrow sea, phlegm dampness, and internal stasis. Its classic prescriptions include “Kaixinsan”, “Smart Soup”, “Yizhi Soup”, “Dihuang Yinzi” and so on.

With the rise and widespread application of big data technology and machine learning technology, machine learning and even deep learning can provide strong support for TCM research. The amount of information in TCM and its prescriptions is large, so the idea based on big data, using machine learning and deep learning-related models to discriminate the underlying characteristic information of the data and find a reasonable and efficient combination of agents is a feasible analysis method, but it also has far-reaching research significance and broad development prospects.

## 2 Related Work

In recent years, many researchers have conducted in-depth research and clinical trials on the treatment of Alzheimer’s disease with TCM. Researchers have summarized the treatment of Alzheimer’s disease with TCM through literature and a large number of experiments. Li et al. [1] summarized the work on the use of TCM in the treatment of Alzheimer’s disease in recent years, including the influence and mechanism of various TCM on animal models and cell models. Sun et al. [2] summarized the pharmacological effects of TCM in the treatment of senile dementia and other diseases in recent years. By some researchers’ study of TCM contains compounds to find the natural active ingredients for the treatment of AD, for example some flavonoids, Gao et al. [3] summarized the flavonoids isolated from TCM in recent years and its mechanism of action, research shows that flavonoids drugs in the future may become a kind of effective drug of AD. Ping et al. [4] introduced the history and experience of TCM in the treatment of AD, and analyzed the advantages of TCM in the treatment of AD from the aspects of the etiology of AD, TCM therapy, and herbal extracts for the treatment of AD. Acetylcholinesterase inhibitors (AChEI) are the first FDA-approved drugs for the treatment of mild to moderate Alzheimer’s disease. Most of these drugs, such as huperzine, were first isolated from TCM. Lin et al. [5] based on the Ellman method, the AChE inhibitory activity of traditional Chinese herbal extracts for insomnia and brain function disorders was detected. The experimental results show that these TCM have great potential for the treatment of AD. HupA is one of an effective AChEI. Jiang et al. [6] reviewed the clinical, pharmacological, chemical and biological structure of HupA.

With the development of the large data analysis and machine learning technology, machine learning and artificial intelligence technology also play an important role in disease treatment, Andong et al. [7] used smartphones combined with machine learning method to quantify the Parkinson’s disease, Spathis et al. [8] with the help of machine learning methods, asthma and chronic obstructive pulmonary disease in diagnosis and clinical trials, The results show that the machine learning method has a high accuracy rate, reaching 97.7% in the diagnosis of asthma. Vijayan et al. [9] summarized the

diagnosis and prediction of diabetes using machine learning methods, and compared the experimental results of different machine learning algorithms. There are also many studies on machine learning in the treatment of Alzheimer's disease. Christian et al. [10] used the machine learning method to identify and diagnose the related biomarkers in MRI, so as to find the early Alzheimer's disease and carry out timely defense and treatment. Joshi et al. [11] established a classification model of Alzheimer's disease by using neural networks, and found that certain specific factors have a greater impact on Alzheimer's disease. Machine learning method is also used in TCM in the treatment of Alzheimer's research, Pang et al. [12] collected 13 kinds of prescriptions of traditional Chinese medicines in the treatment of AD and the AD 25 related targets, selection of 7 kinds of typical Chinese herbal medicine for machine learning classification model, and study by machine learning model target projections for the active components of TCM prescription, this method offers a new way for TCM research and development, but the lack of a mass of data analysis, and an overall analysis of TCM prescription. Chen et al. [13] used deep learning and random forest method to study the Chinese medicine candidates to find the best Chinese medicine formula for the treatment of senile dementia, but this method is only for glycogensynthase kinase 3 (GSK3).

### 3 Methodology

#### 3.1 Properties of TCM

The nature, flavor and channel tropism of TCM are the core parts of the theoretical system of TCM. Nature, flavor and channel tropism are attributes of a drug.

According to the 2015 edition of the Chinese Pharmacopoeia, there are five types of nature of drugs: "calm", "cold", "hot", "warm", "cool". These natures are the predecessors in the long-term observation and experience summary on the yin-yang reconciliation and cold and temperature adjustment of the human body. For example, if the patient has evidence of "hot" such as fever, dryness, etc., it is better to choose "cool" or "cold" drugs during treatment, such as *Coptis chinensis*, *Rhizoma Anemarrhenae*, etc. These can reduce or eliminate symptoms. On the contrary, if the patient shows "cold" symptoms such as cold hands and feet, chills, pale complexion and other symptoms of deficiency and cold, it is better to choose "warm" or "hot" drugs such as dried ginger and aconite. That is basic principles of TCM recorded in the Huangdi's Internal Classic, "Warms the cold and cools the hot".

There are six types of flavor of drugs: "pungent", "sweet", "sour", "astringent", "bitter", "salty". There are two ways to determine the "flavor" of drugs, one is to directly taste the actual taste of drugs, and the other is to observe the different effects of drugs on the human body based on long-term experience, and summarize the theory of drug's "flavor". For example, "pungent" drugs make patients sweat and promote blood circulation, such as ginger, *Radix Aucklandiae*, *Flos Carthami*. And "sweet" drugs have the effect of treating deficiency and relieving pain, such as Ginseng, *Radix rehmanniae preparata*, *Radix Glycyrrhizae*.

There are six types of flavor of drugs: "pungent", "sweet", "sour", "astringent", "bitter", "salty". There are two ways to determine the "flavor" of drugs, one is to directly taste the actual taste of drugs, and the other is to observe the different effects of drugs

on the human body based on long-term experience, and summarize the theory of drug's "flavor". For example, "pungent" drugs make patients sweat and promote blood circulation, such as ginger, *Radix Aucklandiae*, *Flos Carthami*. And "sweet" drugs have the effect of treating deficiency and relieving pain, such as Ginseng, *Radix rehmanniae preparata*, *Radix Glycyrrhizae*.

There are eleven types of channel tropism of drugs: "heart channel", "kidney channel", "lung channel", "liver channel", "spleen channel", "stomach channel", "gallbladder channel", "large intestine channel", "bladder channel", "tri-jiao channel". In TCM, the role of drugs is closely linked to the internal organs in order to illustrate the selectivity of drugs for certain parts of the body. For example, some drugs with eyesight effects belong to "liver channel". Drugs that can treat sores in the tongue belong to "spleen channel". There are also drugs that can belong to multiple channels, such as *Radix Angelicae Sinensis* can belong to "heart channel", "liver channel", and "spleen channel".

### 3.2 Data Processing

**Collection and Validity Labeling of TCM Prescriptions.** This article extracts 224 prescriptions from China HowNet, Weipu, Wanfang and other eligible libraries from 1988 to the present.

According to the data of each prescription in the literature, the proportion of the number of patients with effective treatment to the total number of patients, is regarded as the effective rate of prescription. The prescriptions with an effective rate greater than or equal to 85% are labeled as the category with better effect, and the sample labels are marked as '1', the prescriptions with an effective rate less than 85% are labeled as the category with average effect, and the sample labels are marked as '0'.

**Standardized Data of Drugs.** According to Sect. 3.1, drugs have five nature types, six flavor types, and eleven channel tropism types. If a certain drug has one of the above attributes, set the attribute to "1", otherwise set to "0". For example, *Radix rehmanniae preparata* is "warm", "sweet" and it belongs to "kidney channel" and "liver channel", so we set "00010" as nature attributes in the order of "calm, cold, hot, warm, cool", set "010000" as flavor attributes in the order of "pungent, sweet, sour, astringent, bitter, salty" and set "0101000000" as tropism channel attributes in the order of "heart channel, kidney channel, lung channel, liver channel, spleen channel, stomach channel, gallbladder channel, large intestine channel, bladder channel, trijiao channel". Then the standardized data corresponding to *Radix rehmanniae preparata* is the code "0001010000001010000000" with 22 bits. For other drugs, the corresponding standardized data can be obtained in the same way.

**Standardized Data of Prescriptions.** According to previous section, each drug in a prescription are replaced with a 22 bits code. Since the collected prescriptions contain a maximum of 18 drugs, we supplement the prescriptions of less than 18 drugs to 18 that set all the supplemented medicine codes to "0000000000000000000000". The final standardized data length of each prescription is 18 \* 22. These can be used for deep learning through simple reshape processing. We ended up with a data set of 224 prescriptions.

### 3.3 Training Method

**Multilayer Perceptron.** Multilayer perceptron is a kind of artificial neural network with forward structure. It can get a set of corresponding output vectors from a set of input vectors. The layers of the multilayer perceptron are fully-connected layers, that any neuron in the previous layer is connected to all neurons in the next layer. The multilayer perceptron has a three-layer structure: an input layer, hidden layers, and an output layer, which are constructed in this order. Figure 1 is a simple multilayer perceptron model. The ordinary perceptron can only solve the linear separable problem, and the multilayer perceptron can solve the non-linear separable problem, which greatly improves the network performance.

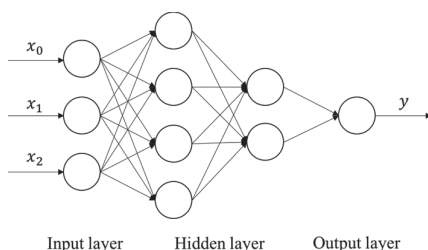


Fig. 1. Simple multilayer perceptron model.

In the paper, the input of multilayer perceptron is a two-dimensional feature matrix of sample, and the output is a one-dimensional vector of sample labels. An example is shown in Fig. 2.

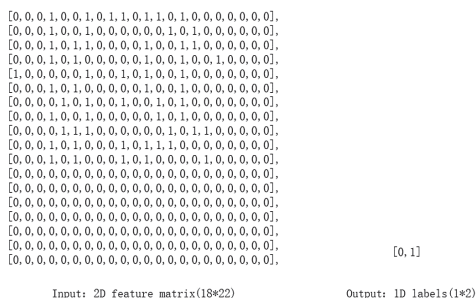


Fig. 2. IO of multilayer perceptron.

**Convolutional Neural Network.** Convolutional neural network is a kind of feedforward neural network. A basic convolutional neural network consists of three distinct layers: a convolutional layer, a pooling layer, and a fully-connected layer. The convolutional layer is the core layer of the convolutional neural network, which is used to remember the characteristics of the network. The pooling layer is periodically inserted



between consecutive convolutional layers. Its role is to reduce the spatial size of the data volume and make the resources required for calculation become less. The fully-connected layer is placed after the convolutional layer and the pooling layer. What the fully connected layer completes is to highly refine the features to complete the final classification or regression. Figure 3 shows an example of a convolutional neural network.

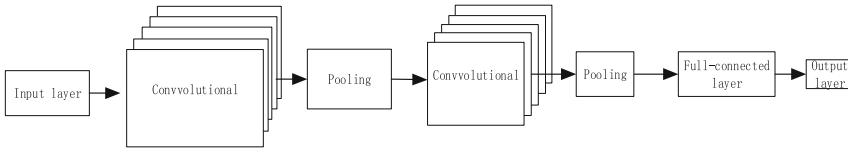


Fig. 3 Simple convolutional neural network.

In the paper, the input of convolutional neural network is a two-dimensional feature matrix of sample, and the output is a one-dimensional vector of sample labels. An example is shown in Fig. 2.

**Support Vector Machines.** Support vector machines (SVM) is a binary classification model. Its basic model is a linear classifier with the largest interval defined in the feature space. The largest interval makes it different from the perceptron; SVM also includes kernel techniques. This makes it a substantially non-linear classifier. SVM’s learning strategy is to maximize the interval, which can be formalized as a problem for solving convex quadratic programming, which is also equivalent to the minimization of the regularized hinge loss function. The learning algorithm of SVM is an optimization algorithm for solving convex quadratic programming.

In the paper, the input of support vector machines is a two-dimensional feature matrix of sample, and the output is a one-dimensional vector of sample labels. An example is shown in Fig. 4.

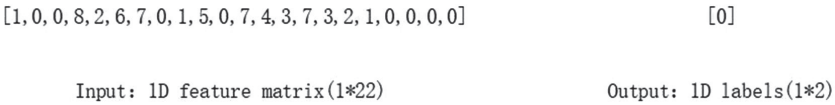


Fig. 4. IO of support vector machines.

## 4 Experimental Results and Analysis

### 4.1 Experimental Results

**Multilayer Perceptron.** The multi-layer perceptron model used in the experiment is shown in Fig. 1.

The model's input data is 18 in height and 22 in width. There are two hidden layers, each with 32 nodes and a dropout rate of 0.8. The hidden layer uses relu as the activation function and the output layer uses softmax as the activation function. The cross-entropy function is used as the loss function, and the Adam algorithm is used as the optimization algorithm. The learning rate is set to 0.001. The data set is divided using the "hold-out" method. The number of training sets is 174, the number of test sets is 50, and the number of training rounds is 80. There are 560 steps in total. The specific parameters are shown in Table 1.

**Table 1.** Multilayer perceptron parameters.

Parameter type	Value/Method
Dropout rate of hidden layers	0.8
Activation function of hidden layers	Relu function
Activation function of output layer	Softmax function
Loss function	Crossentropy function
Optimizer algorithm	Adam algorithm
Learning rate	0.001

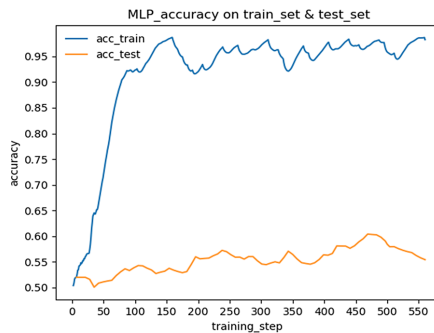
The dropout rate of the hidden layer refers to the probability that the nodes of the hidden layer are retained. Adjust this parameter to prevent overfitting.

The activation function of the hidden layer (/output layer) refers to the function that controls the output of the hidden layer (/ output layer) node.

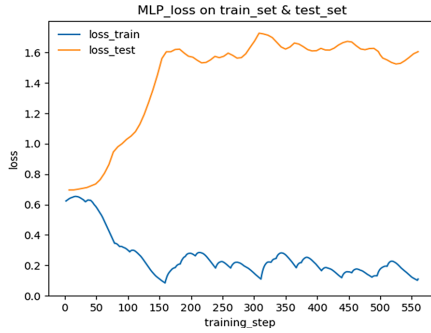
The loss function is a function used to measure the quality of the model's prediction. It represents the degree of gap between the prediction result and the actual data.

The optimizer algorithm is an algorithm that reduces the loss function to find the optimal solution of the model. The learning rate is the step size of the optimizer algorithm and determines whether the loss function can converge to the optimal solution under the optimizer algorithm.

The experimental results are shown in Fig. 5 and Fig. 6.



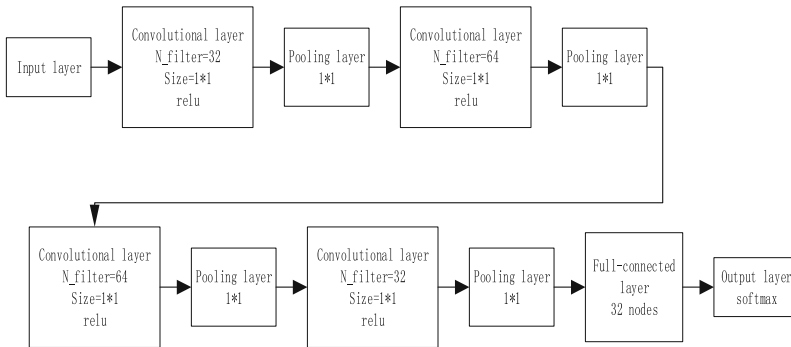
**Fig. 5.** The variation trend of accuracy on training set and test set.



**Fig. 6.** The variation trend of loss in training set and test set.

According to Fig. 5 and Fig. 6, after 550 rounds of training, the loss function of the multilayer perceptron on the training set converges to 0.1–0.3, and the accuracy (prediction accuracy) converges to about 95%, which shows that the multi-layer perceptron can be fitted on the training set and has a good classification results. It can be seen that the loss function on the test set cannot converge, shows an upward trend. And its acc also fluctuates around 0.57. This suggests that the network has overfitted. Generally, the causes of overfitting include too complex model, too few samples, unreasonable hypothesis model. After re-dividing the data set and simplifying the network, the experiment was repeated. However, the experimental results still show overfitting. Therefore, it is determined that the data set is too small resulting in overfitting.

**Convolutional Neural Network.** The Convolutional neural network used in the experiment is shown in Fig. 7.



**Fig. 7.** The Convolutional neural network.

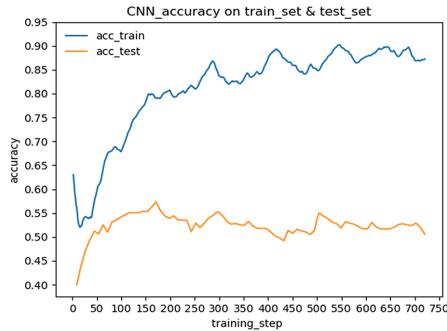
The model’s input data is 18 in height, 22 in width and 1 in depth. The network contains four sets of convolutional layers and pooling layers, and a fully connected layer. The four convolutional layers contain 32, 64, 64, 32 filters, the filter size is 1 \* 1, the pooling layer filter size is 1 \* 1, the number of fully connected layer nodes is 32, and the dropout rate is 0.8. The convolutional layer uses relu as the activation function and the output layer uses softmax as the activation function. The cross-entropy function is

used as the loss function, and the Adam algorithm is used as the optimization algorithm. The learning rate is set to 0.001. The data set is divided using the “hold-out” method. The number of training sets is 164, the number of test sets is 60, and the number of training rounds is 80. There are 720 steps in total. The specific parameters are shown in Table 2.

**Table 2.** Convolutional neural network parameters.

Parameter type	Value/Method
Dropout rate of hidden layers	0.8
Activation function of hidden layers	Relu function
Activation function of output layer	Softmax function
Loss function	Crossentropy function
Optimization algorithm	Adam algorithm
Learning rate	0.001

The experimental results are shown in Fig. 8 and Fig. 9.



**Fig. 8.** The variation trend of accuracy on training set and test set.

According to Fig. 8 and Fig. 9, after 700 rounds of training, the loss function of the convolutional neural network on the training set converges to 0.35, and the accuracy (prediction accuracy) converges to about 90%, which shows that the convolutional neural network can be fitted on the training set and has a good classification results. It can be seen that the loss function on the test set shows an upward trend, and its acc also fluctuates. This shows that the network has overfitted, too. And it is determined that the overfitting should occur because the data set is too small.

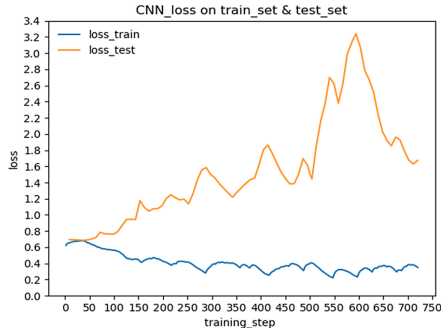


Fig. 9. The variation trend of loss in training set and test set.

**Support Vector Machines.** The algorithm used in this section is NuSVC. The input data width of the model is 22. The upper limit (Nu) of the error rate in the training set training is set to 0.5, Nu is an upper bound on the fraction of training errors, which ranges from (0,1). The kernel function is set to a Gaussian kernel function 'rbf'. And the gamma value is set to 'auto', gamma is a coefficient of the kernel function, which reflects the influence of each training sample, the default value is 'auto', which is 1/number of features.

We made the training set: test set = 3:1, and we used the k-fold cross-validation method to train the NuSVC model on the training set, where  $k = 8$ . The training results are shown in Fig. 10. Using the trained model to predict on the test set, the accuracy was 62.5%.

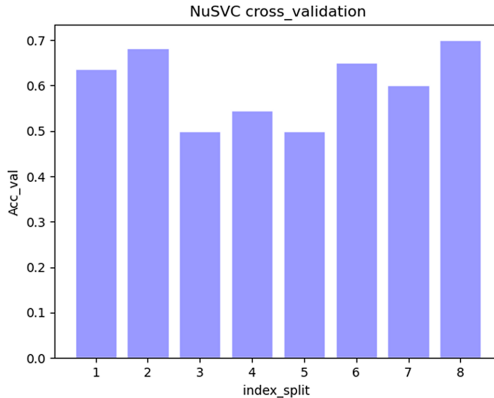


Fig. 10. Cross-validation results of NuSVC.

NuSVC's prediction results on the test set have slightly improved. And according to Fig. 10, it can be seen that NuSVC's performance in the cross-validation method is relatively stable, mostly not less than 60%. This experiment shows that there is feasibility of applying support vector machines to the data set and problem.

## 4.2 Results Analysis

In the problem of predicting the effectiveness of TCM prescriptions, the multi-layer perceptron on the training set increases with the number of training steps, the loss function value decreases and converges to about 0.2, and the accuracy increases and converges to about 0.95. This reflects that the multilayer perceptron algorithm is feasible to fit the effectiveness predictions of TCM prescriptions. However, on the test set of the trained multi-layer perceptron, both the loss function value and the accuracy rate fluctuated greatly, which indicates that the training results appear to overfitted.

The experimental results of the convolutional neural network are similar to the multilayer perceptron. The reasons for overfitting include too many parameters, too complicated model, the number of samples that is too small, and the assumed model which cannot reasonably exist. After trying the methods of re-dividing the data set and simplifying the network, over-fitting will continue to occur in the end.

The number of input layer nodes, which is the sample feature dimensions, of these two types of neural networks is 396 ( $22 * 18$ ). During the experiment, the trained parameters will be several times that of 396, especially the convolutional neural network. In contrast, the data set has a total of 224 samples, and the training set under the leave-out method has only 168 samples. Therefore, it can be considered that too few samples lead to overfitting.

The 22 features' value of the 18 herbs in the prescription were added up to reduce the 396-dimensional features to 22 dimensions. SVM algorithms are used to learn and train the simplified sample data set and observe its effect. The accuracy of SVM on the test set is more than 60%, which proves that the effectiveness of TCM prescriptions could be predicted by machine learning algorithms.

## 5 Conclusion

In this paper, the effectiveness of TCM against AD was analyzed by using machine learning and deep learning methods. The main work of this paper included the establishment of a data set of TCM against AD and the analysis of the attributes of TCM by using machine learning methods. In this paper, 22 properties of each drug were extracted for modeling. Due to the small number of data samples, the neural network model performed poorly on the test set, but SVM performed well on the test set. In future work, we will focus on using machine learning methods for modeling, and we will also use data enhancement methods to expand data sets.

**Acknowledgements.** This work was supported by the National Key Research and Development Program of China under Grand 2017YFB0802704 and National Natural Science Foundation of China under Grand 61972249.

## References

1. Li, L., Zhang, L.: Action characteristics of traditional chinese medicine in treatment of Alzheimer's disease. *Prog. Biochemistry and Biophys*, **039**(8), 816–828 (2013)

2. Sun, Z.-K., Yang, H.-Q., Chen, S.-D.: Traditional Chinese medicine: a promising candidate for the treatment of Alzheimer's disease. *Translational Neurodegeneration* **2**(1), 6 (2013)
3. Gao, J.J., Inagaki, Y., Li, X., Kokudo, N., Tang, W.: Research progress on natural products from traditional Chinese medicine in treatment of Alzheimer's disease. *Intractable Rare Dis Res* **7**(2), 46–57 (2013)
4. Ping, L., Kong, M., Yuan, S., Liu, J., Wang, P.: History and experience: a survey of traditional chinese medicine treatment for Alzheimer's disease. *Evidence-Based Complementary and Alternative Med.* **2014**, 1–5 (2016)
5. Lin, H.Q., Ho, M.T., Lau, L.S., Wong, K.K., Shaw, P.C., Wan, D.C.C.: Anti-acetylcholinesterase activities of traditional Chinese medicine for treating Alzheimer's disease. *Chemico-Biol. Interact.* **175**(1), 352–354 (2008)
6. Jiang, H., Luo, X., Bai, D.: Progress in clinical, pharmacological, chemical and structural biological studies of huperzine A: a drug of traditional chinese medicine origin for the treatment of Alzheimer's disease. *Current Med. Chem.* **10**(21), 2231–2252 (2003)
7. Andong, J., et al.: Using Smartphones and Machine Learning to Quantify Parkinson Disease Severity: The Mobile Parkinson Disease Score, *Jama Neurology*
8. Spathis, D., Vlamos, P.: Diagnosing asthma and chronic obstructive pulmonary disease with machine learning. *Health Inform. J.* **1204**, 146045821772316 (2016)
9. Vijayan, V.V., Anjali, C.: Prediction and diagnosis of diabetes mellitus — A machine learning approach (2016)
10. Christian, S., Antonio, C., Petronilla, B., Aldo, G. M.C.Q., Isabella, C.: Magnetic resonance imaging biomarkers for the early diagnosis of Alzheimer's disease: a machine learning approach. *Front. Neurosci.* **9**, 625–631 (2017)
11. Joshi, S., et al.: Classification of Alzheimer's Disease and Parkinson's Disease by Using Machine Learning and Neural Network Methods (2010)
12. Pang, X.C., Wang, Z., Fang, J.S., Lian, W.W., Du, H.: Network pharmacology study of effective constituents of traditional Chinese medicine for Alzheimer's disease treatment. *Yao xue xue bao = Acta pharmaceutica Sinica*, **51**(5), 725–731 (2016)
13. Chen, H.-Y., et al.: Deep learning and random forest approach for finding optimal TCM formula for treatment of Alzheimer's disease. *J. Chem. Inf. Model.* **59**, 4 (2019)



# A Methodology of Fake Cell Test Based on the RRC Redirection or Reselection Priorities from the 5G Network

Yanan Zhang, Chao Ma, Dong Wang<sup>(✉)</sup>, Tianyu Liu, and Zhi Wu

China Automotive Technology and Research Center, Tianjin 300380, People's Republic of China  
{zhangyanan, machao, wangdong2019}@catarc.ac.cn

**Abstract.** The fake cell test shall be an important method to research the network information or privacy safety in many scenarios. This article will discuss the possibility of a proposal that how to conduct a fake cell test based on the 5G NR radio control layer message. By using the redirection (cell reselection priority) method, the UE (user equipment) shall be redirected to a specified cell which is described by the ARFCN [1] (absolute radio frequency channel number) and Cell ID. The radio resources control (RRC) layer, which is responsible for the cell camp, selection and reselection operations [2]. Also, the RRC message can be easily decoded by some open-source software, which means that the intruder can obtain the details of the target cell and construct a fake one. The intruder shall be able to catch the user's data and extract the privacy information. Especially with the development of automobile networks, mobile communication has been widely used on V2X, and the hacking of vehicles is considered more destructive than using mobile phones. The paper introduces a test method of the fake cell and successfully builds the fake cell to test, which also illustrates the limitation of this kind of fake cell test.

**Keywords:** Fake cell · 5G new radio · Cell redirection · Radio Resource Control

## 1 Introduction

From GSM to NR, cellular networks have taken a big step forward. There is no doubt that LTE or NR networks are sufficient to meet human's private requirements for data transmission. However, the problem is with 2G, or GSM, which is the oldest mobile system in the currently operating network system.

Due to the design time of GSM, Ericsson and Nokia [3] did not fully consider the development of network security. Since then, although we have the substitution of GSM like EDGE, WCDMA, it still plays an important role during this period. To give a specific example, due to the failure of TD-SCDMA technology, the China CMCC network 3G mode is nearly dysfunctional, so GSM serves as a network backup solution, and it is necessary to provide CS (circuit-switched) connection whenever the UE does. As the oldest system, GSM is considered an ideal intrusion tool for cellular systems.



In this paper, we proposed a new methodology of conducting the fake cell attack that focus on connection between the GSM and 4G/5G. After decoding the specific messages of RRC, it is possible to know the redirecting cell information and construct the fake cell and wait for the UE to attach.

## 2 Inter-RAT Mobility and Methods of UE

RAT stands for radio access technologies, and inter-RAT mobility means cooperation among the different network systems [4]. For example, sometimes, the UE may need to be transferred from one system to another with or without connection suspension. The connection between the NR network and GSM is not so clear, but it is possible for a UE to switch from the NR network to the GSM network. One of them is called handover. For any UE in the status of RRC (radio resource control) connected, the RAN (radio access network) [3] controls its mobility, which means the UE must follow the instructions from the network. In order to better control those UEs, the network will periodically require the UEs to perform cell measurements and help them to move from a bad network to a good one, this process is called handover. Generally, RAN considers that GSM cell as low priority to be handover to but sometimes GSM has a better coverage and RSRP (Received Signal Reference Power, the RSRQ parameter is not applied to the GSM network), which helps GSM cell raise its the cell priority. e.g. The UE is shifting from good LTE coverage to poor LTE coverage with good GSM coverage. Then, the network will help the UE to transfer from the LTE cell to the GSM cell [2].

Another method is called redirection or cell reselection priority interference. As shown from the overview of the system, LTE and NR networks do not provide CS connectivity. In LTE, it uses the CS fallback and VoLTE (Voice over LTE) as a remedy for the lack of CS during incoming and outgoing voice [5]. For NR, VoNR is still under construction. In more cases, it drops the UE to LTE, and then decides to use CS fallback or VoLTE according to the network situation. From the perspective of RRC, whenever the network needs a CS fallback solution, it shall send an RRC CONNECTION RELEASE message with *redirectedCarrierInfo* included. Alternatively, it shall use the *cellReselectionPriorities* to control the UE cell selection procedure during the idle mode, it has the same function with redirection as well as providing a specific priority order for a set of available cells [4]. A typical *redirectedCarrierInfo* message entity structure is as follows:

```

RedirectedCarrierInfo ::= CHOICE {
  nr CarrierInfoNR,
  eutra RedirectedCarrierInfo-EUTRA,
  ...
}
RedirectedCarrierInfo-EUTRA ::= SEQUENCE {
  eutraFrequency ARFCN-ValueEUTRA,
  cnType ENUMERATED {epc,fiveGC} OPTIONAL -- Need N
}
CarrierInfoNR ::= SEQUENCE {
  carrierFreq ARFCN-ValueNR,
  ssbSubcarrierSpacing SubcarrierSpacing,
  smtc SSB-MTC OPTIONAL, -- Need S
  ...
}

```

**Fig. 1.** NR RRC *RedirectedCarrierInfo* Structure is generally included in the RRC CONNECTION RELEASE message optionally to specify the target frequency of cell [4]

The *cellReselectionPriorities* is given as if the network wants to override the cell priorities which were broadcast by SIBs (system information blocks). This is used to control the cell selection/reselection procedures for UE in idle mode. The definitions can be found in the specification *TS 38.304 NR: User Equipment (UE) procedures in idle mode*. A typical *cellReselectionPriorities* message entity is displayed as Fig. 2:

```

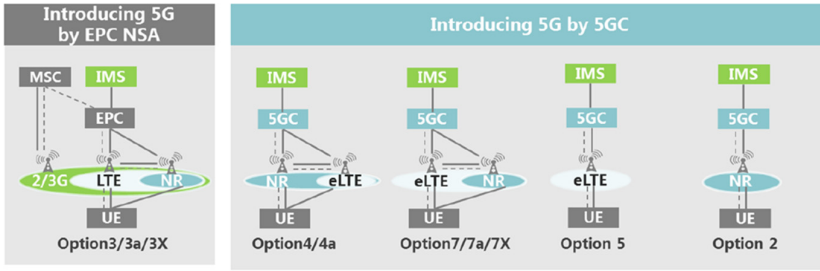
CellReselectionPriorities ::= SEQUENCE {
  freqPriorityListEUTRA FreqPriorityListEUTRA OPTIONAL, -- Need M
  freqPriorityListNR FreqPriorityListNR OPTIONAL, -- Need M
  t320 ENUMERATED {min5, min10, min20, min30, min60, min120, min180,
  spare1} OPTIONAL, -- Need R
  ...
}

```

**Fig. 2.** NR RRC *CellReselectionPriorities* structure can be used to adjust the cell priorities so that to control the cell selection/reselection procedure for UE in the idle mode [4]

### 3 Feasibility Analysis of Redirection Method to a GSM Cell from 5G NR Network

As mentioned above, under the current configuration of the network, there are two methods for moving a UE from NR to GSM. However, the handover method will not be taken into consideration in this article. The reasons are below [6]:



**Fig. 3.** 5G provides so many construction plans for the better development, and the voice Solution options are various [3]

- a. The handover procedure is completely conducted under UE RRC connected status whose mobility is controlled by MME. The hacking behavior to the core network is not as easy as implementing a fake cell.
- b. To ensure the data communication continuity, the data and channel synchronization is necessary. The intruder cannot perform these by himself.
- c. The trigger conditions of handover are stricter, by including the measurements results, loading balance, and system overall configurations.

What we shall focus on is the redirection method, the redirection method is usually used in the CS fallback solution. Figure 3 provides detailed information about the 5G construction architecture, the first one is NSA (non-standalone). NSA aims to expand the coverage of 5G by using existed core network systems. For example, when a UE needs call service in the NR network, it shall be dropped to the LTE system. Then, the LTE system shall evaluate the current network situation and channel quality to determine whether a VoLTE conversion can be conducted. If the channel quality is poor, or, the IMS system loading is high, the LTE system shall initiate the CS fallback with redirection technology.

From this perspective, it is possible to move a UE from NR to GSM/UMTS without any user’s manual configuration. On the other hand, we can find the specification descriptions of RRC release sending prerequisites (Table 1):

**Table 1.** Prerequisite of RRC Release Message [2]

Message	P	A-I	A-C
RRCRelease	+	-	-

Where, P stands for messages that can be sent (unprotected) prior to AS security activation. A – I means messages that can be sent without integrity protection after AS security activation. A – C refers that messages that can be sent unciphered after AS security activation. Although, the *redirectedCarrierInfo* is a field that can be included in the messages, another annotation indicates: Justification for P: If the RRC connection only for signaling not requiring DRBs or ciphered messages, or the signaling connection

has to be released prematurely, this message is sent as unprotected. *RRCRelease* message sent before AS security activation cannot include *deprioritisationReq*, *suspendConfig*, *redirectedCarrierInfo*, *cellReselectionPriorities* information fields [4].

This sentence indicates that *redirectedCarrierInfo* should be strictly protected. Generally, information is encrypted with AS security (the cryptographic algorithm used is usually optional, such as RSA or AES, but they are safe and reliable). However, if the intruder owns a cellphone with test mode, the information may be visible to him. Modem designers (Qualcomm, Hisilicon, etc.) often use log catcher software to analyze and debug the modem system issues. In this kind of software, information can be decoded correspondingly with other open-source software like 3GPP decoder [7].

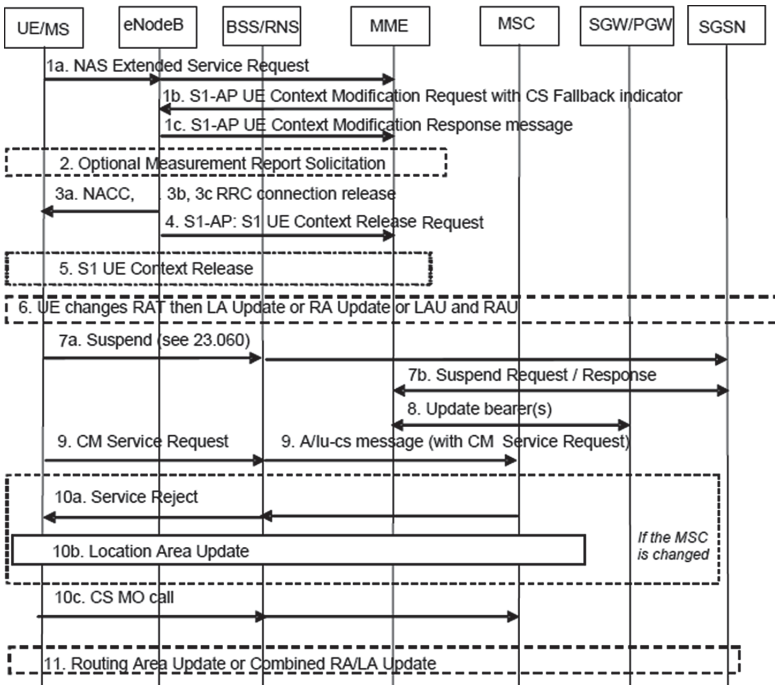


Fig. 4. CS Call Request in E-UTRAN, Call in GERAN/UTRAN without PS HO [5]

The distributed *redirectedCarrierInfo* messages are generally changeable on the scope of large area and a long time, but they are also stable on the scope of a small area (typically, a microcell coverage) and instant. The fake cell coverage does equal to a femtocell coverage (50–200 m) because of the strong requirement of concealment. The large area unauthorized usage of LFB (licensed frequency band) is easy to be detected.

The intruder shall use a cellphone to dial a cs fallback call with CMCC service firstly, to do this, he needs to switch off the VoLTE function in his phone setting to ensure the call will be outgoing under GSM or UMTS network. After this, he shall be able to catch the system log and find the *redirectedCarrierInfo* field within it. Figure 4 gives you the full procedure details of CS fallback.

In this procedure, the intruder can obtain the specific redirected cell info from step 3b and then implement a GSM fake cell with the same cell ID and ARFCN, as Fig. 1 displayed, the *redirectedCarrierInfo* only specifies the carrier frequencies. With no other authentication information, or, the GSM system authentication method is not secure enough. The next part of this article introduces the implementation method of an openBTS fake cell system. Also, if a fake cell had been set up in the same area that intruder got the information, the UE has no ability to distinguish the real one and the fake one. On the other hand, by the descriptions of RR management protocol (radio resource management in GSM), once MTS (mobile terminal station) connected to the cell, it could not release the RR connection by itself unless the situation of the radio link failure.

A condition shall be supposed: an intruder has successfully obtained the redirected information of a microcell and he has built a fake cell with the same cell information and there are many users here who are undergoing CS calls including MO (mobile oriented) and MT (mobile terminated). The result shall be: some of the users may connect to the fake cell during the MT procedure, the MT one is similar to the MO procedure [5].

However, the UE may camp to the fake one and initiate the CS call by waiting for the CC\_ALERT message. Definitely, the fake cell system does not have a connection to the core network and cannot deliver the CC\_ALERT message to the UE. With timer expiry, the UE considers the CS call failure and the GSM cell should confirm it and then let UE move to newer RAT by using redirection or handover. Alternatively, it releases the RR connection and leave UE to reselect to LTE or NR.

The issue is, the fake cell wants UE to attach on it and it never releases the RR connection. The hacked UE must keep the connection until leaving the fake cell coverage. In some extreme situations, this may get worse. For example, the network considers ARFCN 1800 is good for UE to conduct the CS call because it has a low payload, but all UE which received with this information may redirect to the fake 1800 one, this could lead to the lower loading of real one. Finally, the network shall redirect more UEs to the ARFCN 1800 and jump into a bad loop until humanity intervention (Fig. 5).

## 4 The Implementation of Fake Cell with OpenBTS

OpenBTS (Open Base Transceiver Station) is a software-based GSM access point, allowing standard GSM-compatible mobile phones to be used as SIP endpoints in Voice over IP (VoIP) networks. OpenBTS is an open-source software developed and maintained by Range Networks. The public release of OpenBTS is notable for being the first free-software implementation of the lower three layers of the industry-standard GSM protocol stack. It is written in C++ and released as free software under the terms of version 3 of the GNU Affero General Public License [1].

To implement the openBTS, the intruder requires a base operating system (Ubuntu Server 12.04 LTS). Here we use the openBTS UMTS architecture as an example but actually the system used the openBTS GSM architecture. From the perspective of safety, we use openBTS UMTS as an example to illustrate the principle of constructing the fake cell because this kind of method cannot be used in the redirection attack methodology as it does not support the ciphering (encryption) and USIM-based authentication. This will

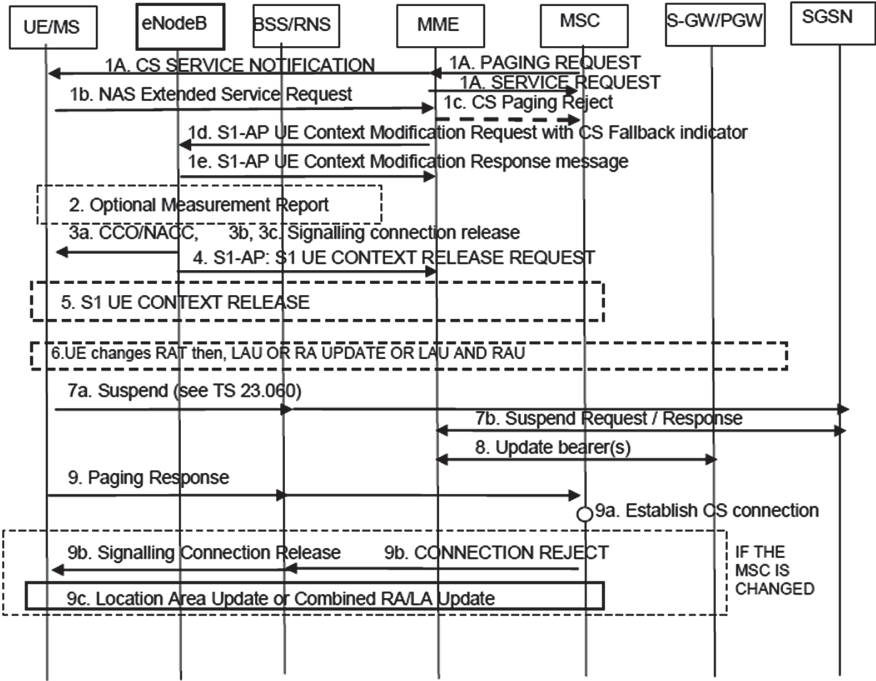


Fig. 5. CS Page in E-UTRAN, Call in GERAN/UTRAN without PS HO [5]

cause that the failure camping of the UE after its release connection from LTE. On the other hand, the GSM allows the USIM-based authentication method to be configured as the option “none”, which means that the MS could camp on the fake cell without using an authentication method, this will allow the UE camp on the fake GSM cell after redirection immediately [4].

OpenBTS-UMTS is a Linux-based application that uses a software radio to present a UMTS network to any standard 3G UMTS handset or modem. It builds upon the OpenBTS framework, where the MS or UE is treated as an IP endpoint at the edge of the network. It supports the original UMTS Release 99 (or Release 3) while it also provides the function or packet-switched services only (Tables 2 and 3).

The system used on this fake cell shall be more powerful as the OpenBTS-UMTS is a more computationally intensive application compared with Open-BTS (GSM). Its channel bandwidth is nearly 13x larger than the GSM channel. From the technology theory, WCDMA (Wide-band Code Division Multiple Access) uses 5 Mbps bandwidth with a 3.84 Mcps chip rate while the GSM uses 200 kHz with 100 Kbps. However, it is not precise to directly compare these two technologies from the aspect of connection speed because they are completely different from two air interfaces. In somehow, it shall be confirmed, the UMTS system is more computationally intensive than the openBTS GSM one.

**Table 2.** Ettus Devices that Capable with OPENBTS [1]

	Transport	Recommended RF	Frequency Accuracy	
B200	USB 3.0	Integrated	TCXO 2.0 ppm	GPSDO < 1 ppb
B210				
X300	1 or 10 Gb Ethernet	SBX, WBX, CBX	TCXO 2.5 ppm	
X310				
N200				
N210	1 Gb Ethernet			
USRP2			VCXO 20 ppm	NA

**Table 3.** OPENBTS Supported Frequency Bands with Different Devices [8]

UMTS Band Support			
	Frequency Range	UMTS Bands	Output Power
B200/B210	70 MHz–6 GHz	1–14, 19–21, 22, 25, 26, 32	up to 100 mW
WBX	50 MHz–2.2 GHz	1–14, 19–21, 25, 26, 32	
SBX	400 MHz–4.4 GHz	1–14, 19–21, 22, 25, 26, 32	
CBX	1.2 GHz–6 GHz	1–4, 7, 9–11, 21, 22, 25	

We shall now jump back to the SIMs/authentication problem, UMTS mandates mutual authentication between the UE and the NodeB. This is a major change from 2G/2.5G authentication, where only the BTS authenticates the MS. So, what does this mean? The subscriber registry will need to know the SIM’s  $K_i$  value to [1]:

- 1) *perform authentication and*
- 2) *enable integrity protection.*

Without proper authentication and integrity protection, the UE will not attach (or register) with OpenBTS-UMTS. For most users, this means you must provide the SIMs for the UEs on the network. The only way to use SIMs from another provider is to obtain the  $K_i$  through a roaming interface to the provider’s HLR/HSS [6].

This also means that some of the features that circumvented authentication in OpenBTS, like open registration, are not possible with OpenBTS-UMTS. USIMs (e.g. 3G SIMs) are not currently supported by the OpenBTS-UMTS implementation. They require different authentication algorithms than GSM SIMs; these algorithms are not supported in the public release of OpenBTS-UMTS.

## 5 Results and Conclusions

We did implement the openBTS GSM system based on the Ettus USRP B210 hardware. Under the situation of manual configuration, successfully made that up to 5 smartphones had been attached to the openBTS GSM system. We did try to sniff the SMS (short message services) in the system as all the data in this situation are transmitted with plaintext. Due to the software license issue, we were not able to log and decode the smartphone RRC\_RELEASE message because the log analysis software is usually designed by the modem system designer (fabless IC chip designer) and only authorize to the Tier 1 customer to use. There are some versions of the unauthorized logger tool software on the internet but we cannot find the test smartphone one, the Samsung Galaxy SII with a Texas Instrument processor.

This article proposed one of methodology to attack the smartphone with fake cell implementation. It also exposes the weakness of the current 4G and 5G systems. To the LTE and NR themselves, are highly protected and secure enough but the disadvantages exist in the cooperation among different network systems. Especially, the redirection from any systems to GSM.

With the development of mobile communication and the voice over LTE/NR solutions, the opportunities of redirecting UE to GSM is getting smaller and smaller. To make a conclusion, the current V2X system still faces some potential cybersecurity risks, and it shall be solved well in the not-too-distance future.

## References

1. Aaron, X.: Implementation of the GSM Test Network with OpenBTS (in Chinese). 9. 4. 2018. <https://blog.csdn.net/xrh003/article/details/79447468>. Accessed 17 Feb 2020
2. 3. W. Group: LTE Evolved Universal Terrestrial Radio Access (E-UTRA) Radio Resource Control (RRC) Protocol specification (3GPP TS 36.331 version 9.0.0 Release 9), in *Group, 3GPP Work* (2009)
3. Huawei Technologies Co., “Vo5G Technical White Paper. In: Huawei Technologies Co., Ltd. (2018)
4. 3. W. Group: Radio Resource Control (RRC) Protocol Specification (3GPP TS 38.331 version 15.8.0 Release 15), In: 3GPP 5G Work Group (2020)
5. 3. W. Group: Digital cellular telecommunications system (Phase 2 +) Universal Mobile Telecommunications System (UMTS) Circuit Switched (CS) fallback in Evolved Packet System (EPS) Stage 2 (3GPP TS 23.272 version 9.15.0 Release 9), in *Group, 3GPP Work* (2013)
6. Ghadialy, Z.: UMTS Security: A Primer, 06 11 2004. [https://www.3g4g.co.uk/Tutorial/ZG/zg\\_security.html](https://www.3g4g.co.uk/Tutorial/ZG/zg_security.html). Accessed 30 Mar 2020
7. 3. U. Team, Yang, Q., Huang, L., Zhang, W., Haoqi, S., Jun, L.: Radio security attack and defense revealed (in Chinese), Beijing: Publishing House of Electronics Industry (2016)
8. OpenBTS.org, “OpenBTS-UMTS,” 2 12 2014. [http://openbts.org/w/index.php?title=Main\\_Page](http://openbts.org/w/index.php?title=Main_Page). Accessed 15 Mar 2020
9. Buchanan, C., Ramachandran, V.: Kali Linux Wireless Penetration Testing Beginner’s Guide (in Chinese). Post & Telecom Press, Beijing (2018)





# Keeping Privacy Data Secure Under Factory Recovery

Wang Lianfang<sup>1</sup>, Wang Ye<sup>2</sup>, Zhao Gang<sup>3</sup>, Liu Lu<sup>1</sup>(✉), and Kuang Xiaohui<sup>3</sup>

<sup>1</sup> School of Computer Science and Technology,  
Beijing Institute of Technology, Beijing 100081, China  
{wanglianfang, liulu}@bit.edu.cn

<sup>2</sup> Exchange, Development and Service Center for Science and Technology Talents,  
The Ministry of Science and Technology (MoST), Beijing 100045, China  
wangye@sttc.net.cn

<sup>3</sup> National Key Laboratory of Science and Technology on Information System Security,  
Beijing 100101, China  
zhao-gang20@126.com, xiaohui\_kuang@163.com

**Abstract.** Nowadays, mobile devices are widely used. Mobile smartphones are not only a carrier for users to record, store, or transfer sensitive data in social network, but also a covert forensic tool for some security departments. In some extreme cases like war correspondent being about to be captured, the operation records of shooting and communication on the mobile phone will bring them disadvantages. Immediately performing a factory recovery is the smartest option. However, this operation will clear all user data, including those important case scene evidence. This paper proposes a flexible and reliable factory recovery mechanism to protect sensitive data under factory recovery. By analyzing the factory recovery process and mastering the Linux kernel execution process, to ensure that when users perform factory recovery operations, applications and data can be cleared normally, and the private data that users want to keep is protected. We evaluated the safety and usability of the method.

**Keywords:** Android device · File system · Data protection · Factory recovery

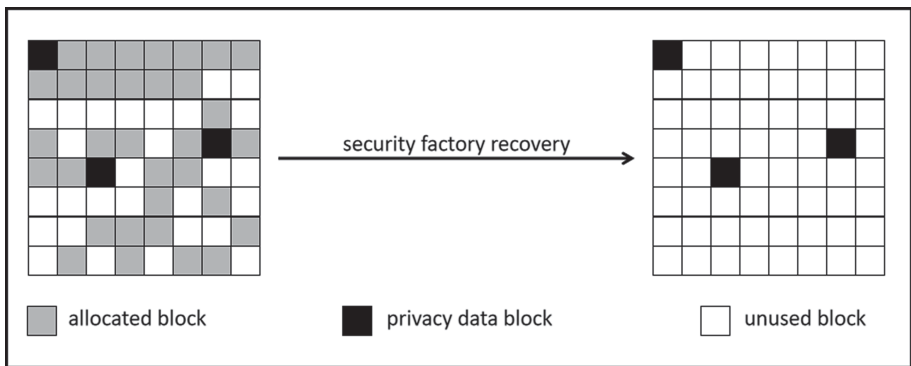
## 1 Introduction

Smartphones have become pervasive due to the availability of office applications, Internet, games, vehicle guidance using location-based services apart from conventional services such as voice calls, SMSes, and multimedia services. Android devices have gained huge market share due to the open architecture of Android and the popularity of its application programming interface (APIs) in the developer community [1]. A recently statistic published by Statista Research Department shows the market share of mobile operating systems in China from January 2013 to December 2018. In August 2019, Android held a share of 79.32% of the mobile operating system market in China [2].

The high degree of openness and convenience also brings unprecedented security threats to personal privacy data stored in mobile phones [3]. According to the 620,000

user-uploaded vulnerability detection reports collected by the 360 Perspective Mirror APP, as of August 2019, 99.97% of Android mobile phones have security vulnerabilities, which means that most mobile phones are facing the risk of data leakage [4]. With the continuous expansion of mobile phone functions, users' personal privacy data has become ubiquitous [5, 6]. How to continuously improve the security of data stored in mobile phone memory is the long-term efforts of Android parent company Google, and the data protection technology under the Android platform is also the research hotspots in the security field [7, 8]. Even if we use smart phones with highest security level, and carefully protect our private data to ensure that they are not maliciously stolen, there is still a huge security threat [9, 10]. All your data and usage behaviors are stored on phone's flash memory, which can be easily retrieved by forensic tools [11, 12]. So, before you replace or sell your phone, remember to do a factory recovery [13].

Smartphone's built-in factory recovery can basically meet the needs of ordinary users, but considering such users and scenarios: human rights workers secretly shoot a lot of precious live videos in conflict areas and store them in the hidden partition of their devices. Once they are faced with a captured emergency, they must immediately perform a factory recovery operation to prevent the special behavior and communication content on the mobile phone from threatening their personal safety [14] (Fig. 1).



**Fig. 1.** Keeping privacy data blocks in user-data partitions during factory recovery process.

However, after performing a factory recovery, not only the software that collect evidence, but also the evidence collected will be discarded. The main focus of this thesis is to explore how to completely protect the data stored in the hidden partition after performing a factory recovery operation.

The main contributions of this paper include:

1. A comprehensive analysis of the work done by Android operating system and Linux kernel after client performs a factory recovery operation.
2. To develop a file system analysis tool in recovery mode, which greatly improved the accuracy and convenience of partitioned data analysis.

3. Based on the analysis of first step, to modify the kernel source code, and develop a security factory recovery mechanism, and provide effective data security tools for users in specific scenarios.

## 2 Background

In the Android phone's settings options, you can find a command to reset the system [15]. When the user clicks the "reset" button, the phone will restart into factory mode and perform the actual erase operation in normal mode. After erasing the data partition, restart it to enter the normal operating system. In order to grasp what is going on inside the phone during this process, you need to understand some basic Android phone technologies, including: Android system partition, Linux file system, Android recovery, Linux kernel.

### 2.1 Android System Partition

The Android system divides the ROM into different partitions, and each partition function is independent of each other [16, 17].

**/boot** partition is responsible for mobile phone startup, and it includes the kernel (kernel) and virtual memory disk (ramdisk). Kernel is the most basic part of the operating system and is the core of an operating system. **/system** partition contains the complete Android operating system, including the Android GUI and all system applications on the device. **/recovery** partition is an important partition for system developers, because when your custom operating system fails to start, this partition can also provide you with a simple interactive interface to perform mobile partition erase, restart, and system upgrade and so on. **/data** partition is extremely important to users (almost mean everything to users) because it stores your daily contacts, messages you send and receive, photos and videos you take, system settings that match your own habits, and Download all applications installed. **/misc** partition is a very small partition, about 4 MB. Recovery uses this partition to save some information about the upgrade and responds to the situation where the device is powered off and restarted during the upgrade. **/cache** partition is a system cache area that temporarily stores application data (to save data here, a specific app permission is required), and OTA upgrade packages can also be stored here.

### 2.2 Linux File System

The Linux file system is a software organization responsible for managing and storing file information in the Linux operating system [18]. Ext4 is currently the most popular file system type in the Android file system, built on partitions such as `/data`, `/system` [19]. At the hardware layer, the smallest storage unit on the memory is a sector, which takes up 512 bytes. Ext4 forms a block with multiple sectors as the smallest logical storage unit. Generally, the default size is 4 K. To facilitate management, multiple blocks form a block group. Blocks are divided into two types: data blocks that store data and metadata blocks that describe file system information. An inode contains the metadata of a file, such as timestamps, user and group permissions, as well as pointers to data blocks. A superbblock

contains metadata of file system and a group descriptor stores metadata of a particular block group. In addition, each block group has a data block bitmap, which records the allocation of data blocks in this group.

### 2.3 Android Recovery

Recovery mode refers to a mode that can modify the data or system inside the Android machine (similar to Windows PE or DOS) [20]. As mentioned earlier, the Android system usually has `a/recovery` partition, which is also composed of the Android kernel and the ramdisk. In fact, when the system starts, it will parse the BCB control block command from the bootloader to determine which partition image to load.

### 2.4 Linux Kernel

Factory recovery is just a user program. What really interacts with the hardware and reads and writes the contents of the block is the Linux kernel function `ioctl`. Linux's running space is divided into kernel space and user space [21]. They run at different levels and are logically isolated from each other. Normally, user programs cannot access kernel data or call kernel functions [22]. To restore factory settings, a program must call a kernel function by using a system call. The user programming interface (API) is a function definition that explains how to obtain a given service.

## 3 Design

### 3.1 Analysis of the Process of Factory Recovery

The code of the Android application layer is written in the Java language [23].

When the user clicks the “reset” button at the application layer, a series of codes will be executed to complete the final erase operation as shown in Fig. 2.

#### User layer

1. The user clicks the “reset” button;
2. Sending a broadcast called `ACTION_FACTORY_RESET` (`android.intent.action.FACTORY_RESET`) in the click event of the button.

#### Framework Layer

1. The broadcast receiver `MasterClearReceiver` receives the broadcast parameters and calls the `rebootWipeUserData` method in `RecoverySystem.java`;
2. The method `rebootWipeUserData` passes the “-wipe\_data” command to the `bootCommand` method;
3. The `bootCommand` method calls the system service `RecoverySystemService.java` and passes the command parameters to the `rebootRecoveryWithCommand` method;
4. The `rebootRecoveryWithCommand` method calls the `setupOrClearBcb` method to write a command to erase data to the BCB. Then, restart the system through the `reboot` method of `PowerManager`;

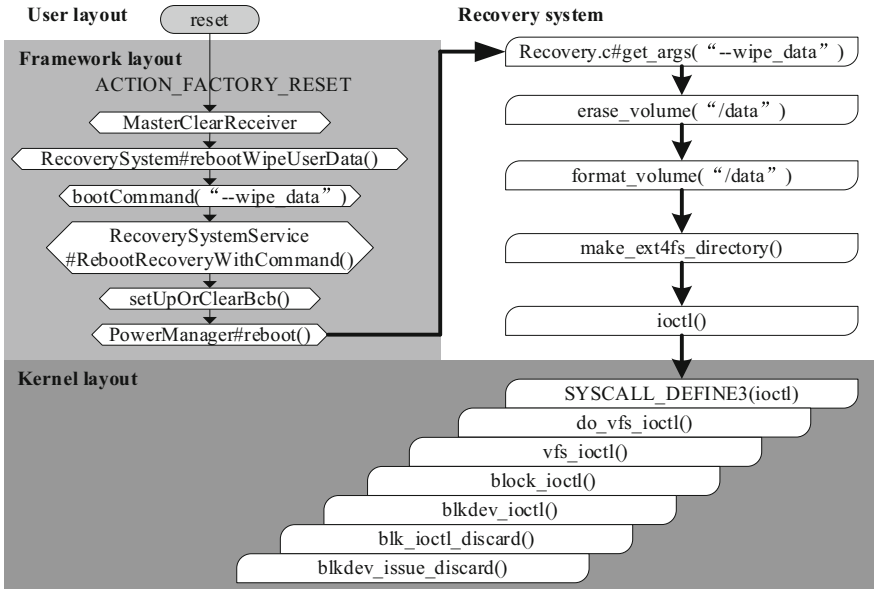


Fig. 2. Factory recovery process.

### Recovery Layer

1. The system reads the “boot-recovery” command of the BCB block to boot the system into recovery mode;
2. The main function in bootable/recovery/recovery.c calls the `get_args()` method to obtain the command parameter “-wipe\_data”;
3. To execute `erase_volume (“/ data”)`, while the function actually calls the `format_volume (“/ data”)` function;
4. `Format_volume` calls the `make_ext4fs_directory` function, which actually runs the `make_ext4fs` tool, which is used to make an ext4 file system image on the Android platform;
5. To analyze the `make_ext4fs` tool source code. The `write_ext4_image()` function writes the file system image to the partition file. The function executes the `ioctl` function.

### Kernel Layer

1. `SYSCALL_DEFINE3` takes the `ioctl()` function as a parameter, and then calls the `do_vfs_ioctl()` method;
2. The `do_vfs_ioctl()` method calls the `vfs_ioctl()` method;
3. The `vfs_ioctl()` method calls the `block_ioctl()` method;
4. The `block_ioctl()` method calls `blkdev_ioctl()` method;
5. The `blkdev_ioctl()` method calls `blk_ioctl_discard()` method;
6. The `blk_ioctl_discard()` method calls the `blkdev_issue_discard()` method.

### 3.2 Privacy Data Protection Scheme

The user wants to be able to protect one or more audio and video files specified by himself when performing the factory recovery operation. Each file has a unique inode number. According to the inode number, the actual block number where the file content is stored can be located. During the execution of the program, we tried to modify and add code in different functions to achieve data protection, but the simplest and most effective method is to implement it in the `blkdev_issue_discard()` function, because it uses the block as the minimum unit of operation, and iteratively wipes every block of the `/data` partition. In order to achieve our purpose, it is only necessary to determine whether the block is a block that needs to be protected before erasing. Otherwise, the block is skipped.

**Table 1.** The algorithm of security factory recovery.

<b>Algorithm</b> security discard
1: <code>blkdev_issue_discard</code> (sector, nr_sects)
2: <b>while</b> (nr_sects)
3: <b>if</b> (sector in privacy_sectors)
4:         sector ++.
5: <b>else</b>
6: <b>discard</b> (sector).
7:     nr_sects --.

In the `blkdev_issue_discard (* bdev, sector, nr_sects, gfp_mask, flags)` function, the parameter `sector` represents the start sector, and the parameter `nr_sects` represents the number of sectors to discard. The core algorithm of the function is to use `nr_sects` as the body of the while loop, and erase the parameter sector where it is located Sector, then increment the value of sector. The modified erasure algorithm can be expressed as Table 1, where `privacy_sector` is the sector where the privacy data is located.

## 4 Implementation

### 4.1 Data Analysis Tool

In order to evaluate the effectiveness of the data protection function before and after the factory recovery is performed, the `/data` partition metadata and data block content need to be fully analyzed [24]. Under the Android platform, there are a large number of softwares that implement similar functions, but their common disadvantage is that in the normal mode, after the `/data` partition is mounted, the data of some blocks is modified. Another way to analyze the `/data` partition is to dump the entire `/data` partition from the phone memory. This approach does guarantee the integrity of the data analysis, but an obvious disadvantage is that the time cost is too high. Only for the evaluation of the function of factory recovery, it is necessary to analyze the change of some blocks of the `/data` partition at least 3 times, which is unrealistic.

After fully considering the two factors of data integrity and time feasibility, this paper compiles and implements a file system analysis tool that runs in recovery mode. In recovery mode, after unmounting the partition, you can still access the contents of the/data partition and ensure the correctness of the content. This tool receives commands and parameters from the command line, and mainly implements the following main functions: taking the file inode number as a parameter and outputting the block number occupied by the file; using the block number as a parameter to view the contents of the data block; using the block group number as a parameter, checking the data block allocation and inode node application status in the block group; taking the block number as a parameter, filling the block with hexadecimal numbers.

### 4.2 Hidden Space and Privacy Data [25, 26]

Private data is stored in a hidden space to ensure sufficient security [27–29]. Wang’s paper used an effective method to obtain a large amount of hidden space on the/data partition. This method obtains large-capacity, discretely distributed hidden space by creating a host file to modify the block bitmap. The hidden space is composed of multiple sections, and the distribution can be represented by a two-dimensional array. The first parameter is the starting block number and the second parameter is the offset. The array is defined in the kernel space as a parameter for determining whether the block to be erased is in the hidden space. Hidden partitions can only write private data with custom programs, and ordinary users cannot read and write these data.

## 5 Evaluation

In this section the effectiveness and performance of security factory recovery will be evaluated. A Samsung S8 development mobile phone (Android 7.0, Kernel 4.4.13, Internal storage:64G) is used as the test environment.

### 5.1 Functional Evaluation

The specific test steps are as follows, and the results are shown in Fig. 3.

```
dreamlte:/data/local/tmp # ./blc -b 594124
[0000] ee ee ee ee ee ee ee ee ee ee ee ee ee ee ee |
[0016] ee ee ee ee ee ee ee ee ee ee ee ee ee ee ee |
[0032] ee ee ee ee ee ee ee ee ee ee ee ee ee ee ee |
[0048] ee ee ee ee ee ee ee ee ee ee ee ee ee ee ee |
```

Fig. 3. Data read from the hidden spaces.

- 1. Create a normal file normal\_data.txt with a size of 1G and all contents of 0xcc in the root directory of the/data partition.

2. Execute command `blc -w 238` to fill the blocks in the hidden spaces with number 238 (0xee).
3. Perform a security factory recovery.
4. Check whether the file `normal_data.txt` in the `/data` directory exists; To execute the command `blc -b n` to check whether the data in the hidden space is still 0xee.

The result shows that after security factory recovery, hidden space data is retained while other data is cleared.

### 5.2 Performance Evaluation

We use the hidden space size as a parameter to test the execution speed of the security factory recovery. Among them, the test time is counted from when the factory recovery button is pressed to the phone returning to normal state (Table 2).

**Table 2.** The speed of the security factory recovery.

	Size of hidden space(G)	Speed(s)
Traditional factory recovery	/	161.58
Security factory recovery	4	162.04
Security factory recovery	8	160.59
Security factory recovery	16	157.96
Security factory recovery	32	163.02

The results show that the code added in the kernel has a small effect on the speed of factory setting. This result is in line with what we expected, because we only made very few changes in the kernel.

## 6 Conclusion and Future Work

we analyze the execution flow of the internal code of an Android device when performing a factory recovery and then propose an innovative method of factory recovery based on the demand to protect private data. This method requires little modification of the linux kernel to make sure privacy security when normal factory recovery. After that, a JNI program running on recovery mode is developed to analyze file system. At last, the experiment results show that private data would not be cleared when factory recovery, nor would it lose data after the mobile phone was restarted.

Of course, our proposed method still has several limitations that make it impossible to apply to the actual scene:

1. It takes a long time to perform a factory recovery. The requirements for ordinary scenarios can be met, but in order to clear specific data in an emergency, the shorter the time, the better.



2. The hidden space is still easy to be detected by file system check software, although it has been concealed.

In the future, we will try to solve the above limitations and then apply the method to the actual scene. At the same time, we will consider encrypting data in hidden space to increase security.

**Acknowledgments.** This work was supported by the National Natural Science Foundation of China under Grant No. 61876019.

## References

1. Faruki, P., et al.: Android security: a survey of issues, malware penetration, and defenses
2. Statista. <https://www.statista.com/statistics/262176/market-share-held-by-mobile-operating-systems-in-china/>. Accessed 27 Nov 2019
3. Kim, J., Jung, I.Y.: Private data protection of android application (2018)
4. <https://zt.360.cn/1101061855.php?dtid=1101061451&did=210942656>. Accessed 10 Nov 2019
5. Li, Z., Yang, J., Cui, B.: Study on sensitive data protection based on SEAndroid. In: 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA) (2018)
6. Shahriar, H., et al.: Data protection labware for mobile security (2019)
7. Nguyen, N.H., et al.: No-key protocol for deniable encryption (2018)
8. Xue, Y., Tan, Y., Liang, C., Zhang, C.Y., Zheng, J.: An optimized data hiding scheme for Deflate codes. *Soft. Comput.* **22**(13), 4445–4455 (2017). <https://doi.org/10.1007/s00500-017-2651-2>
9. Yu, X.: A fault-tolerant and energy-efficient continuous data protection system. *J. Amb. Intell. Hum. Comput.* **10**, 1–10 (2018)
10. Xiao, Yu., Changyou, Z., Yuan, X., Hongfei, Z., Yuanzhang, L., Yu-An, T.: An extra-parity energy saving data layout for video surveillance. *Multimedia Tools Appl.* **77**(4), 4563–4583 (2017). <https://doi.org/10.1007/s11042-017-4540-1>
11. Yu, X., et al.: A high-performance hierarchical snapshot scheme for hybrid storage systems. *Chin. J. Electr.* **27**(1), 76–85 (2018)
12. Zhang, X., Tan, Y., Zhang, C., Xue, Y., Li, Y., Zheng, J.: A code protection scheme by process memory relocation for android devices. *Multimedia Tools Appl.* **77**(9), 11137–11157 (2017). <https://doi.org/10.1007/s11042-017-5363-9>
13. Schwamm, R.: Effectiveness of the Factory Reset on a Mobile Device (2014)
14. Zhu, D., Fan, Z., Na, P.: A dynamic credible factory reset mechanism of personal data in android device
15. Dong, H.K., Eom, Y.I.: iDiscard: enhanced Discard() scheme for flash storage devices. In: 2018 IEEE International Conference on Big Data and Smart Computing (BigComp) (2018)
16. Xue, Y., et al.: Isolating host environment by booting android from OTG devices. *Chin. J. Electr.* **27**(03), 617–624 (2018)
17. Zhu, H., Zhang, Q., Liu, L., Aourra, K., Tan, Y.: Research of optimizing the system partition in android system. In: Xhafa, F., Patnaik, S., Yu, Z. (eds.) *IISA 2016. AISC*, vol. 541, pp. 168–173. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-49568-2\\_23](https://doi.org/10.1007/978-3-319-49568-2_23)
18. Wang, H.S.O.C., et al.: An Analysis of the File System for Linux., Korea, p. 5 (2012)

19. Yudha, F., Prayudi, Y.: Block and inode based analysis on EXT4 File System **24**(1), 652–655 (2018)
20. Fang, D.R., et al.: Research on recovery method of deleted data for android system. **40**(10), 275–280 (2014)
21. Bovet, D., Cesati, M., Oram, A.: Understanding the Linux Kernel, 2n edn. (2002)
22. Yuan, P., et al.: Device-specific Linux kernel optimization for android smartphones (2018)
23. Park, J.H., et al.: An enhanced security framework for reliable Android operating system. Secur. Commun. Networks p. n/a-n/a
24. Talley, A.V.: Content analysis tools in android memory forensics (2014)
25. Liu, Z.L., Pun, C.M.: Reversible data-hiding in encrypted images by redundant space transfer. s 433–434, 188–203 (2018)
26. Abdel-Nabi, H., Al-Haj, A.: Efficient joint encryption and data hiding algorithm for medical images security. In: 2017 8th International Conference on Information and Communication Systems (ICICS) (2017)
27. Jiang, R., et al.: A high-capacity reversible data hiding method in encrypted images based on block shifting. In: 2017 2nd International Conference on Multimedia and Image Processing (ICMIP) (2017)
28. Abdulaziz, N.K.: Digital watermarking and data hiding in multimedia (2017)
29. Sharma, A., Sharma, N., Kumar, A.: A new algorithm to secure image steganographic file. In: 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence (Confluence) (2017)



# An Intelligent File Transfer Optimization for Poor Network Conditions

Ming Yan<sup>1,2</sup>, Bo Zhang<sup>3</sup>, Gang Zhao<sup>4</sup>, Xiaohui Kuang<sup>4</sup>, Lu Liu<sup>1</sup>,  
and Ruyun Zhang<sup>5</sup>✉

<sup>1</sup> School of Computer Science and Technology, Beijing Institute of Technology,  
Beijing 100081, China  
395600460@qq.com

<sup>2</sup> Institute of Artificial Intelligence and Blockchain,  
Guangzhou University, Guangzhou, China

<sup>3</sup> Information Center of Ministry of Human Resources and Social Security,  
Beijing 100090, China  
imagepool@126.com

<sup>4</sup> National Key Laboratory of Science and Technology on Information  
System Security, Beijing 100091, China  
zhao-gang20@126.com, xiaohui\_kuang@163.com

<sup>5</sup> Zhejiang Lab, Hangzhou, Zhejiang Province, China  
zhangry@zhejianglab.com

**Abstract.** File transfer is based on the reliable TCP protocol. However, when the network is of poor quality, TCP-based transmission will still perform less effectively due to some reasons. Existing approaches mainly optimize file transmission by modifying network strategies or optimize transmission mechanism. But when network conditions cannot be selected or modified, a suitable parameter adjustment method would be needed. The experiments in this paper firstly perform TCP-based file transmission tests on a poor network condition and find two problems. Then, multiple tests were performed on multiple platforms to detect the conditions under which these problems occurred. Next, to address the issue and improve the transmission performance, we propose an intelligent optimization scheme. By adjusting the transmission parameters and adding policies, the scheme equips the intermediate parameters with intelligent self-adaptation capabilities. We also test and evaluate the performance of the intelligent scheme. The result shows that the file transfer under the new scheme not only basically avoids the target problems, but also reduces the practical upload time under poor network condition from the perspective of decreasing the number of retransmissions and reducing the failure rate.

**Keywords:** File Transmission · TCP Optimization · Poor Network Conditions

---

Supported by the National Natural Science Foundation of China under Grant No. 61876019 and Zhejiang Lab (NO. 2020LE0AB02).

© Springer Nature Singapore Pte Ltd. 2020  
Y. Xiang et al. (Eds.): SocialSec 2020, CCIS 1298, pp. 234–244, 2020.  
[https://doi.org/10.1007/978-981-15-9031-3\\_21](https://doi.org/10.1007/978-981-15-9031-3_21)

## 1 Introduction

Mobile embedded devices works anywhere. However, these devices are not always in a high-quality network environment. There may be various reasons that make the network environment poor. Generally, file data is transmitted via the TCP protocol. Due to inherently limited network resources, various reasons can easily lead to low TCP transmission efficiency. This prompted some recent research efforts in studying Network resource allocation, network equipment parameter adjustment, etc. Nevertheless, there are still some critical issues which are closely related to practical implementations and have not been well addressed:

- i) The poor network caused by inherent factors, making the network environment cannot be optimized. For example, the device is in a location with poor wireless signal coverage. Under these circumstances, developers can not change the network transmission conditions by existing methods and device can only work on poor networks.
- ii) More retransmissions increase time consumption in the whole. More frequent file retransmission, poor network condition leads, makes the transmission time longer. Even worse, programs more likely misinterpret some other errors as transmission failure.
- iii) The network where the portable embedded device located changes in real time [1]. So, strategy needs to be dynamically adjusted.

Anyway, we cannot guarantee that the network environment will always be high-quality under any circumstances through optimization. This article proposes an optimization scheme for TCP transmission in poor environment, test it combine with FTP and evaluate the improvement.

The contributions of this article are as follows:

- (1) We tested the transmission between the same client and multiple service providers on multiple system platforms. The results show that some of the platforms have problems when sending files in poor network condition. And then we analyzed and evaluated the frequency and conditions of these errors.
- (2) For the two types of problems that arise, we adjusted the parameters and strategies during file transfer to build a new intelligent solution with adaptive capabilities to deal with such problems.
- (3) In addition. we tests and evaluates its improvement effects. Results show that the new intelligent scheme mostly avoids the problems and reduces the file transfer time by 16% as a whole.

The rest of the paper is organized as following. Section 2 describe related works. Section 3 briefly introduce the general transmission logic and its problems when it is transmitted in a poor network. Section 4 analyze the problems, then proposed some adjustment to improve transmission performance. Section 5 describes the environment and data during the entire experiment. Section 6 gives conclusions. And Sect. 7 makes discussion and looks forward.

## 2 Related Work

Many existing algorithms to optimize network conditions have been widely used in many fields, including congestion control, routing, traffic engineering, and load balancing. Although these ideas look promising on paper or in simulations, they may perform poorly in practice [2]. Some of scheme attach new strategies to optimize the transformation [3]. Take an instance, Q. Zhao propose a solution based on multi-path TCP and software-defined networking, which, when applied to mobile wireless heterogeneous networks, reduces the network handover delay and improves the total throughput for transmissions among various entities [4]. Some of researchers adjust the strategies of protocol to make TCP more efficient [5,6]. For example, Andreadis proposed algorithm is to integrate the jitter-based error estimation introduced with a cross-layer approach. The algorithm continuously monitors the wireless channel error status, by counting the number of inferred congestion and non-congestion loss events. According to this monitoring activity, the congestion control scheme takes an appropriate decision about TCP window size reduction and retransmissions [7]. But once developer can not modify the environment or based system, they are not suitable enough, obviously.

## 3 Transmission and Detection

### 3.1 Fundmantal Logic

Build the upload logic according to the general FTP mode, as shown in the Algorithm 1.

The key details are as follows:

- (1) At receivingRespondCode step, the 3th, set the waiting time for receiving the response code from the server to 10 s, considering the high latency.
- (2) When the process receives some special signals (such as SIGPIPE), exit and cause transmission of this turn failure, maybe the current network environment is in poor quality. Wait for 20 s.

---

#### Algorithm 1. TraditionalUpload

---

```

1: fd=connect(sock,ip,port)
2: read send data.
3: receiveRespondCode(con)
4: size ← getServerFileSize(filename)
5: if localFileSize!=size then
6:   reupload
7: end

```

---

### 3.2 Experimental Discovery

Under poor network condition, testing the file sending, we can find that there is not so much odd in the transmission under windows and Linux x86 systems except short in speed. But under Android kernel platforms, something occurred several times during the 20pm–2am, approximately. The phenomena follows:

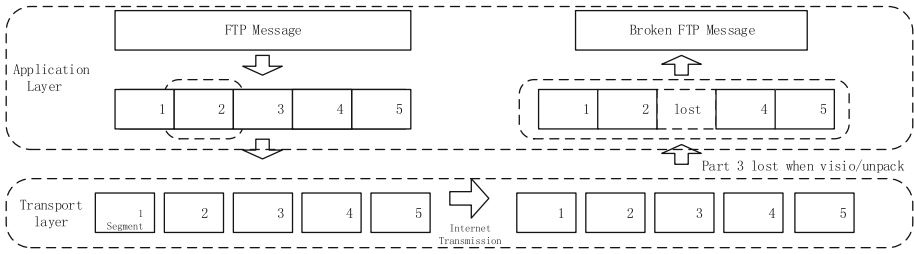
- (1) The file received by the server is smaller than the uploaded file size.
- (2) After the server returns a response code for a successful file upload and waits for 3s, use the SIZE command to verify that the file does not actually exist on the server.

## 4 Analysis and Optimization

### 4.1 Considerations

Take the environment and condition into account, reasons are as follows:

- (1) When the buffer size of the network devices is out of proper range while network latency is high, the efficiency of data transmission will be reduced [8,9]. Correspondingly, the same as the file sender like client [10]. Combine with poor network and large size send buffer, data sends delay appears [11,12]. According to all kinds of congestion control theory, poor network contributes to small TCP congestion window [13]. As saw from the former program, function send() returns after copying the data into the socket buffer which the data port corresponding to. After waiting for 3s, the progress sends SIZE quote to verify through command port. But, the socket bind with the data port, at same time, has not sent data of large amount to the server yet. So, the server cannot find the file. As for the Android kernel, the size of the socket sending buffer is larger than other platforms. Actually, the delay may change constantly while in an unstable network state. In order to make waiting time intelligently self-adapted, the process could achieve it by checking socket tx\_queue [14,15]. That is to say, wait till the tx\_queue emptied instead of wait for constant seconds.
- (2) Poor network conditions make the transmission adopt TCP's congestion control strategy. Smaller TCP's congestion window split a normal message packet into more segments as showed Fig. 1. Unstable latency disrupts the package's order. When receiving, server needs to divide and sort them which means unpack package. Not only the client sending more frequently, the server also needs to stick more times to turn the segment into logical message packets. Although the transmission based on reliable TCP, part of the data more likely be covered or lost by mistake at application layer because of higher frequency visio package, unpack package [16,17]. Actually it is not rare in actual application scenarios for these problems.



**Fig. 1.** Data lost when frequently copying to congestion window or from advertised window.

As showed in the Fig. 1, the FTP Message is divided into many packets by a small sliding window, and then sent by TCP/IP multiple times. At last, packets are received to the advertised window and spliced into a complete message. During the packet receiving and sticking, the third segment has not been correctly copied to the buffer by the application at the application layer, or may be covered. But at this time, the TCP of the transport layer has been transmitted over. The receiver will not ask to retransmit the packet, consequently, lead to the third segment lost. Thereby reduce the probability of the phenomena described above [18].

- (3) When the network delay is high, the time for the server with respond to the response code becomes longer. On the on hand, for high-latency networks, co-relation should sustain

$$\theta + \Delta < \varphi \tag{1}$$

when the transmission is successful. Where  $\theta$  refers to the time for the server to return the response code under normal circumstances, and  $\Delta$  is the delay for the response code in poor network. On the other hand, note that the total time to transmit the file is  $t$ . The time function send costs is  $\eta$ . The retransmission cost is  $t_r$ . And the mean of  $t$  sustains

$$E(t) = \eta + \Delta + \theta + t_r \tag{2}$$

And the  $t_r$  is

$$t_r = \frac{\theta + \tau + \Delta + \eta}{\mu} \tag{3}$$

Where  $\mu$  notes the probability of failure. The receiving waiting time set by the client is  $\varphi$ . And the waiting time after determining that the file transmission fails is  $\tau$ . For a network with a bad state,  $\Delta$  is constantly changing, and the results obtained in actual tests are generally  $E(\Delta)$ . If the probability is high that the transmission is successful, then waiting time  $\varphi$  need to sustain

$$E(\Delta) \ll \varphi - \theta \tag{4}$$

It is more effective to extend the waiting time when receiving the server response code than retransmit it [19].

- (4) Function `recv` or `read` are used in network programming to receive the messages and response codes returned by the server. Such system-call functions can be interrupted by unnecessary exceptions or system signal, such as `EINTR`. When the network condition is of poor quality, blocking time becomes longer. So, the probability of a program being interrupted by such an exception is larger. Once the receiving function returns an error for these reasons, retransmissions will increase the time cost [20]. Try to restart the system call multiple times to receive the response code is a good idea.

## 4.2 Methodology

According to ideas mentioned above, the transmission logic changed like Algorithm 2. Where the details of the change are as follows.

- (1) Change the send buffer size in the send data step from 64K to 16K.
- (2) Add the step `waitTillSocketTxQueueEmpty()` to wait until the socket sending queue is empty.
- (3) In the function `receiveCode`, IO multiplexing (like function `select`) is used to judge the read and write status of the socket. The time to wait for receiving the response code is adjusted from the original 10s to 20s.
- (4) Increase the times of receiving the response code. If the response code is not successfully received after 3 turns, regard it as failed. On the contrary, break loop and continue.

---

### Algorithm 2. ChangedUploadFunction

---

```

1: SetSocketBuffer(16*1024)
2: fd=connect(sock,ip,port)
3: read and send data.
4. WaitTillSocketTxQueueEmpty()
5: for i from 0 to 2 do
6:   select(fd)
7:   success ← receiveRespondCode(con)
8:   if success=true then break;
9: end
10:if success=false then reconnect
11: end
12: size ← getServerFileSize(filename)
13: if localFileSize!=size then
14:   reupload
15: end

```

---

## 5 Experiment Results and Discussion

The new solution is used to repair transmission problems, but from the perspective of reducing failure rate and decreasing the times of retransmissions, there may be some improvement. So it should be evaluated from both problem correction and performance improvement.



## 5.1 Environment

Tests is run at three platform:Android Kernel, Windows 10 and Ubuntu 16.04. The build tool of Android Kernel is NDK 3.8.1. The build tool of Windows 10 is visual studio 2015 4.8.03761. The build tool of Ubuntu 16.04 is gcc 5.4.0.

The Android kernel is a plurality of mobile phone kernel platforms based on linux 4.14.85 and linux 4.9.59.

Android kernel Tx\_buffer are as Table 1. Far more larger it is than x64 Ubuntu as we mentioned in Sect. 4.1.

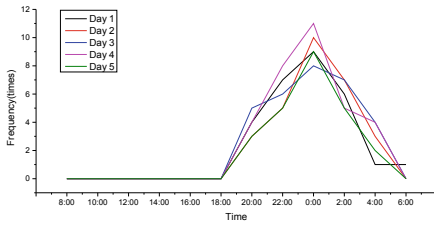
**Table 1.** Tx\_buffer size of multiple devices

Model	Min (KB)	Default (KB)	Max (KB)
OnePlus 7T	262144	524288	1048576
XiaoMi 8	262144	524288	4194304
SAMSUNG S10	524288	1048576	4194304
HUAWEI Mate10 pro	524288	1048576	4194304
x64 Ubuntu	4096	16384	4194304

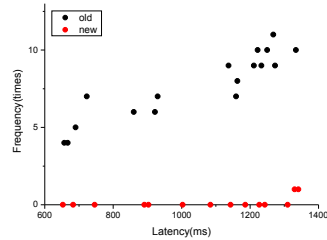
## 5.2 Practice Found

Under poor network condition, we have found that there is not so much odd in the transmission under windows and Linux computer systems except short in speed. But under Android kernel platforms, Something occurred several times during the 20pm–2am, approximately. The details of the phenomena follows:

- (1) The file received by the server is smaller than the uploaded file size. Taking the sending of 191227022854.rar for example, the sending log shows that this file is 588900 bytes in size and a total of 58890 bytes have been sent. The server returned the transmission successfully with response code 226. But after the first uploading, the server received 556 KB. And the second time retransmitted 543 KB. Then the third time retransmissions, 576 KB was received, which refers to the complete file data. That is to say, some data lost in the first two transmissions.
- (2) File transfer complete. But the server can not find the file. After sending the file completely, the server returns response code 226 with Transfer complete. This code and message mean that file transfer over and successfully. And then the client sends quote SIZE to verify the size of file in server. It returns the response code 550 with message that file not found, which means the server can not find the file received just now.



**Fig. 2.** Frequency of errors mentioned over time



**Fig. 3.** Frequency of errors - latency under old and new scheme.

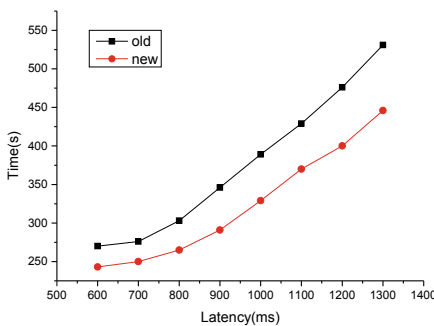
### 5.3 Frequency Test

In order to count the frequency of these two phenomena, the test uploads 240 files each time. Their size ranges from 400 KB to 700 KB. During the test, the latency is about 900 ms. The test results are in Fig. 2 frequencytime, shows that in a poor network, the two phenomena mentioned most likely occur between 20: 00–04: 00. In fact, when the network environment is unstable, such problems will make a greater impact on file transfers.

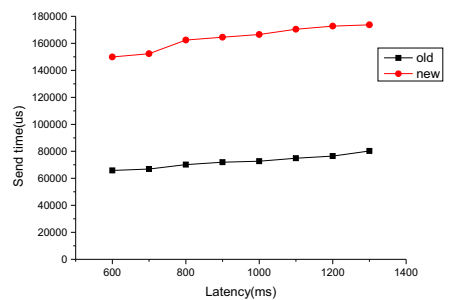
### 5.4 Effect Test

Since the latency is not so stable during the transmission, take it every 10s. And take the average into account. The file size is between 400 KB and 700 KB. Upload at 22: 00–02: 00, 120 files each time. The result is as Fig. 3.

It can be seen from the data in the Fig. 3 that the intelligent scheme almostly avoids problems mentioned at Sect. 3.2. Actually, when testing in new scheme, errors only occurred twice under the worst network condition of the whole test. And it is data loss error. In addition, we can avoid it by reduce the sending rate more.



**Fig. 4.** Time consumption of uploading - latency under old and new scheme.



**Fig. 5.** Time consumption of single file data sending - latency under old and new scheme

## 5.5 Optimization Test

This chapter tests the improvement of the intelligent scheme. 60 files are sent each turn. File size is between 400 KB and 700 KB. When sending the data, send is called every time combine with 3 ms wait. Before sending SIZE quote, wait 3 s. If the transmission fails, wait 5 s till the next retransmission. Network connection broken makes process quit. And every time it happens, wait 20 s for the file retransmission. Considering that latency varies any time, regard latency  $x$  in  $[x/100 * 100, x/100 * 100 + 100)$  as  $x/100 * 100$ . And then take the average of all the time consumption, whose latency in this interval. Under the constant consumption, the whole consumption is as follows in Fig. 4.

As shown by the data in the Fig. 4, the new scheme has reduced the whole upload time for about 16%. And then, record the time consumption of single file data sending, which refers to step6–step12 in Algorithm II. Then we tested the time consumption of the single file data sending, as Fig. 5 shows Fig. 5 records the time consumption of data copied from program buffer to socket buffer, which is the time consumption of function send. Actually, the send rate of the new scheme is much slower. But Fig. 4 shows it costs less time. As shown in the formula (2), when  $E(t)$  gets less and  $\eta$  becomes larger,  $\Delta$  and  $\tau$  is regarded as constant valve,  $\mu$  get larger in Fig. 5, we can judge that  $\theta$  get less. In addition, the quantity of retransmission get less. Then  $t_r$  becomes less, which meet our prediction.

## 6 Conclusion

This paper tests the TCP transmission from multiple platforms to service providers by FTP in poor network. The results show that even in the same logic, although there is no problem with the transmission on windows and Linux platform, but two special errors occur on the Android kernel platform when uploading by the native program. We conducted a series of transmission tests to find out the relationship of the frequency and time. And then we proposed an intelligent optimization scheme. Experiments result shows that the new scheme not only mostly avoids the above errors, but also reduces the transmission time for about 16%. And it improves the FTP transmission in poor network condition as a whole.

**Acknowledgment.** This work was supported by the National Natural Science Foundation of China under Grant No. 61876019 and Zhejiang Lab (NO. 2020LE0AB02).

## References

1. Wang, Z., Zeng, X., Liu, X., Xu, M., Wen, Y., Chen, L.: TCP congestion control algorithm for heterogeneous internet. *J. Network Comput. Appl.* **68**, 56–64 (2016). <https://doi.org/10.1016/j.jnca.2016.03.018>. <http://www.sciencedirect.com/science/article/pii/S1084804516300327>

2. Dukkupati, N., Cheng, Y., Vahdat, A.: Research impacting the practice of congestion control. *SIGCOMM Comput. Commun. Rev.* **46**(3) (2018). <https://doi.org/10.1145/3243157.3243171>
3. Showail, A., Shihada, B.: Battling latency in modern wireless networks. *IEEE Access* **6**, 26131–26143 (2018). <https://doi.org/10.1109/ACCESS.2018.2836439>
4. Zhao, Q., Du, P., Gerla, M., Brown, A.J., Kim, J.H.: Software defined multi-path TCP solution for mobile wireless tactical networks. In: *MILCOM 2018–2018 IEEE Military Communications Conference (MILCOM)*, pp. 1–9 (2018)
5. Kudo, T., Taketa, T., Hiranaka, Y.: Proposal of cross-layer bandwidth assignment with buffer size indication for TCP flow control. *Electron. Commun. Jpn.* **102**(9), 27–37 (2019). <https://doi.org/10.1002/ecj.12203>, <https://onlinelibrary.wiley.com/doi/abs/10.1002/ecj.12203>
6. Dong, M., Li, Q., Zarchy, D., Godfrey, P.B., Schapira, M.: PCC: re-architecting congestion control for consistent high performance. In: *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pp. 395–408. USENIX Association, Oakland, CA, May 2015, <https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/dong>
7. Andreadis, A., Rizzuto, S., Zambon, R.: A cross-layer jitter-based TCP for wireless networks. *EURASIP J. Wirel. Commun. Network.* **2016**(1), 1–11 (2016)
8. Showail, A., Jamshaid, K., Shihada, B.: Buffer sizing in wireless networks: challenges, solutions, and opportunities. *IEEE Commun. Mag.* **54**(4), 130–137 (2016). <https://doi.org/10.1109/MCOM.2016.7452277>
9. Nalini, M., Priyadarsini, U.: To improve the performance of wireless networks for resizing the buffer. In: *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, pp. 1–5 April 2019. <https://doi.org/10.1109/ICIICT1.2019.8741406>
10. Miao, R., Li, B., Liu, H.H., Zhang, M.: Buffer sizing with HPCC (2019)
11. Chauffournier, L., Ali-Eldin, A., Sharma, P., Shenoy, P., Towsley, D.: Performance evaluation of multi-path tcp for data center and cloud workloads. In: *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering. ICPE 2019*, Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3297663.3310295>
12. Im, Y., et al.: I sent it: Where does slow data go to wait? In: *Proceedings of the Fourteenth EuroSys Conference 2019. EuroSys 2019*, Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3302424.3303961>
13. Carletto, A.Q., Santisteban, J.: Shallow window reduction for congestion control under TCP. In: *2019 UNSA International Symposium on Communications (UNSA ISCOMM)*, pp. 1–7, March 2019. <https://doi.org/10.1109/UNSAISC.2019.8712828>
14. Abdelmoniem, A.M., Bensaou, B.: Hysteresis-based active queue management for TCP traffic in data centers. In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1621–1629, April 2019. <https://doi.org/10.1109/INFOCOM.2019.8737369>
15. kumar Shukla, S., Ghosal, D., Farrens, M.: Tuning network i/o processing to achieve performance and energy objectives of latency critical workloads. 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. 1499–1508. IEEE (2019)

16. Shukla, S. K., Ghosal, D., Farrens, M.: Tuning network i/o processing to achieve performance and energy objectives of latency critical workloads. 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. 1499–1508, August 2019. <https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00207>
17. Garg, S., Sharma, P., Singh, V.: A novel approach for efficient bandwidth utilization in transport layer protocols. In: Bhateja, V., Tavares, J.M.R.S., Rani, B.P., Prasad, V.K., Raju, K.S. (eds.) Proceedings of the Second International Conference on Computational Intelligence and Informatics. AISC, vol. 712, pp. 467–479. Springer, Singapore (2018). [https://doi.org/10.1007/978-981-10-8228-3\\_43](https://doi.org/10.1007/978-981-10-8228-3_43)
18. Kato, T., Diwakar, A., Yamamoto, R., Ohzahata, S., Suzuki, N.: How insufficient send socket buffer affects MPTCP performance over paths with different delay. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S. (eds.) Trends and Advances in Information Systems and Technologies, pp. 614–624. Springer International Publishing, Cham (2018). [https://doi.org/10.1007/978-3-319-77712-2\\_57](https://doi.org/10.1007/978-3-319-77712-2_57)
19. Avranas, A., Kountouris, M., Ciblat, P.: Energy-latency tradeoff in ultra-reliable low-latency communication with retransmissions. *IEEE J. Sel. Areas Commun.* **36**(11), 2475–2485 (2018). <https://doi.org/10.1109/JSAC.2018.2874143>
20. Sato, Y., Koga, H., Ikenaga, T.: TCP using adaptive FEC to improve throughput performance in high-latency environments. *IEICE Trans. Commun.* **advpub** (2018). <https://doi.org/10.1587/transcom.2018EBP3091>



# A VirtualXposed-Based Inline Hooking Framework for Android Native Methods

Shuo Feng<sup>1,4</sup>, Yu-an Tan<sup>1</sup>, Gang Zhao<sup>2</sup>, Xiaohui Kuang<sup>2</sup>, Xiao Yu<sup>3</sup>,  
and Juan Wang<sup>1</sup>(✉)

<sup>1</sup> School of Computer Science and Technology, Beijing Institute of Technology,  
Beijing 100081, China

981588349@qq.com, {tan2008,wangjuan99}@bit.edu.cn

<sup>2</sup> National Key Laboratory of Science and Technology on Information  
System Security, Beijing 100093, China

zhao-gang20@126.com, xiaohui\_kuang@163.com

<sup>3</sup> Department of Computer Science and Technology,  
Shandong University of Technology, Zibo 266590, China

yuxiao8907118@163.com

<sup>4</sup> Institute of Artificial Intelligence and Blockchain, Guangzhou University,  
Guangzhou 510006, China

**Abstract.** Hooking is an important technique for monitoring application performance and adding features to applications. Various hooking frameworks are developed to intercept events and process their own specific events. The hooking tools for Java methods are varied, however, the native hook has few methods. Besides, the commonly used Android hook frameworks cannot meet the requirement of hooking the native methods in shared libraries on non-root devices. Even some approaches are able to hook these methods, it has limitations or is complicated to implement. In the paper, a feasible hooking approach for Android native methods is proposed and implemented, which doesn't need any modifications to both Android framework and app's code. In this approach, the method's reference address is modified and control flow is redirected. Beyond that, we combine this approach with VirtualXposed which aims to run it without root privileges. This hooking framework can be used to enforce security policies and monitor sensitive methods in shared objects. The evaluation of the scheme demonstrates its capability to perform hook operation without a significant runtime performance overhead on real devices and it is compatible and functional for the native hook.

**Keywords:** Native hook · Call encryption · ELF file format · VirtualXposed

# 1 Introduction

Android is the most popular smartphone operating system over the world [1]. However, due to the open-source character and its popularity, there are a number of different threats to the Android OS users' privacy, such as malware [2], spyware, the negligence of benign application, or the lack of fail-safe defaults in Android SDK [3]. Moreover, Android apps' static analysis becomes more and more difficult nowadays because both benign and malicious developers use various protection techniques, such as code obfuscation [4] to prevent the apps from reversing. Hook technique can be used by developers to deploy new security extensions on their device or to analyze apps dynamically [5] which is able to avoid modification of the original code.

We should notice that if we modify the target method's instructions in shared libraries directly, and repackage the app, we need to re-sign the app, which breaks Android's signature-based same origin policy and furthermore causes legal concerns [6]. Currently, there are some common native hooking methods: PLT Hook and Inline Hook, which have their own advantages and disadvantages. PLT Hook directly modifies the GOT table in the ELF file to jump to user-defined hook function code when calling external functions of the shared library [7]. However, internal custom functions, so as not in the PLT table, cannot be hooked. The basic principle of Inline Hook is to insert jump instructions in a code segment to direct the program execution process to the functional code required by the user [8]. However, there may be different compatibility and stability issues for processors of different architectures, processor instruction sets, compiler optimization options, and operating system versions. And it may be difficult to analyze and locate the problems.

Due to the limitations of the current native hooking methods, this paper will introduce a new hooking method. This method can hook both external methods and inner user-defined functions, so it has a wide range. The idea of this method is tampering the reference address in the relocation section which saves the offsets of the methods defined in the .text section. The sections which have the methods' reference contain .got which stores the entry address of external invocation, .data.rel.ro which has the vtable of C++ class, and other relocation section. This hooking method is achieved by absolute address calculation and rewriting this address to the new function's address to redirect control flows to the patch code. This hooking method does not require register calculation and assembly code modification, so it has low technical difficulty. Since this method can hook C++ inner functions, using it can make up for the lack of PLT method.

## 2 Background

### 2.1 ELF File Structure

Shared libraries in Android are relocatable Executable and Linkable Format (ELF) files that map to the address space of the process when loaded. The main parts of ELF format are the ELF file header, section header table (SHT),

and program header table (PHT). At the beginning of the ELF file, there is a fixed-length file header which includes the starting position and length of SHT and PHT in the current ELF file. ELF organizes and manages information in sections. ELF uses SHT to record basic information for all sections. It mainly includes the type of section, the offset in the file, the size, the relative address of virtual memory after loading into memory, the alignment of bytes in memory and so on.

For an ELF file that is dynamically linked to other shared object files, to call the shared object functions, it is an instruction to jump to some stub function in the procedure linkage table (PLT). In order to retrieve the real address of the target function, the stub function then performs a memory load on the entry in the global offset table (GOT). That is, the GOT table contains an array of all dynamically linked external functions' pointers that are referenced by their code. In the period of dynamic linking, the GOT table is filled with the right function pointers depends on the maps between GOT entry and the function in shared object files, in the control of the metadata stored in the ELF file. The PLT hook exploits this level of indirection introduced by dynamic linking. It goes through every loaded ELF file and overwrites GOT entries with the pointer to user-defined methods, which is equivalent to performing dynamic linking another time, but substituting the method's pointer.

## 2.2 Dynamic Dispatch in C++

In C++ dynamic dispatch rules, when a virtual method is called, the real implementation depends on the runtime type of the calling object, which allows the subclass to override the parent class's method implementation. Virtual Method Tables (vtable) are commonly used in dynamic dispatch, which contains the pointers of the virtual methods in a class. The compiler assigns an identifier to each virtual method. The items in the vtable represent the corresponding method of the class.

To implement call for a virtual method, the following steps are performed: first, the vtable pointer located at the starting position of the object is loaded. Then, search the item in the vtable depending on the index of the called method. Last, the method implementation found in the vtable is invoked.

## 3 Design and Implementation

The goal of this framework is to enable developers to hook the native methods of Android apps without modifying both Android system and app. Therefore, the design is oriented towards modifying the target native methods' runtime address after the SO file is loaded into memory space and the relocation period. This framework contains two components. The first component is the core hooking engine and the patch code written in C which will be compiled to SO libraries. The other one is the Java side that is used for loading the former SO libraries and performing the hijack operation. The core engine aims to modify the content



in reference address which points to the target method to the address of patch code. Moreover, it stores the original methods' entry address in ELF. The patch methods should have the same parameters form with the target ones.

Suppose that you want to intercept calls to a native method in a shared library. For example, it contains vulnerable code, or it can actually deal with sensitive data. You have to define your own C method and override the target method by using the native hook method described in this paper. All calls to the target method will be intercepted and then go to your C method. Then, the patch code receives the target method's arguments as its own parameters. Internally, this framework can invoke the original implementation of the target method by the method's address stored before the hooking phase and get the return value of it. This hooking framework supports loading and running patch code from the shared library. Because the hook core engine and the patch code both written in C, the hook program's execution is more efficient.

This hooking framework is based on the virtual environment and uses VirtualXposed tool [9]. Normally if we want to hook other processes, the root privilege is needed to modify the app's virtual memory or inject the hooking library in the running app or the Zygote [10] master process [11]. The VXP is used to load and enable the SO libraries which work as the core hooking engine and contain the patch code into the current process. Since both the target process and the hook code are running in the same virtual environment, the hooking library can modify the target process's address space without the root privilege.

Now, we explain this hook method design in figures. Figure 1 shows the app's memory layout without hooking. The libvoipCodec.so file defines the target native methods. The method has data cross reference in the relocation section. SendDataToChannel method is a virtual method, so its reference is in the vtable. Instead, Fig. 2 represents the app's memory layout while hooking is enabled. First, the hooking library is loaded inside the virtual environment which also has the target app. After this, the hooking method uses its internal functions to calculate the absolute address of the reference items which point to the target methods. And then, it can hijack the methods to implement hooking operations.

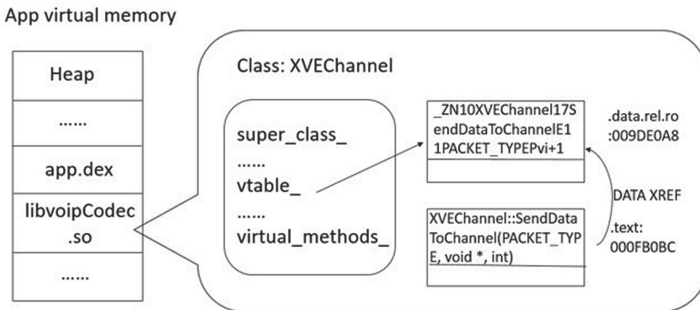


Fig. 1. Application's memory layout without hooking a method.

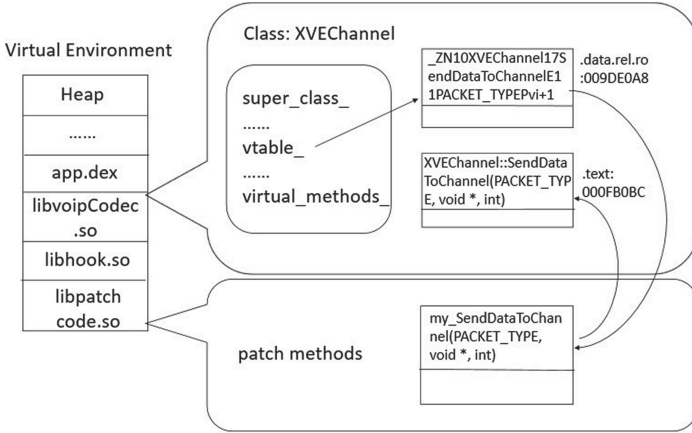


Fig. 2. Application’s memory layout with hooking a method.

The target method’s absolute address after the program is loaded into memory can be calculated by adding the base address of the ELF files and the target method’s reference offset address. The IDA pro tool can be used to check the relative addresses when analyzing the ELF file. The base addresses can only be obtained at runtime. We know that the mmap area is used to map dynamically linked libraries used by the executable file. Therefore, “/proc/<pid>/maps” file is accessed to get the mapping information of the specified process’s memory space, which includes the filename and the base address of the shared libraries.

To guarantee reliable hooking, this method first verifies the offset address of the target method which is defined in the ELF’s .text section with the address stored in the relocation section which has been modified to absolute address after the relocation period. After the binary and operation are executed between those addresses and 0xffe, the result should be the same and then the target absolute address can be modified.

To register the hook points, the native method “start” is invoked, in which we should provide the target method’s information which contains the SO’s filename that the method belongs to, the original method’s offset address, the reference offset address in ELF, and the patch method. And then, in order to extract the shared libraries’ information of the target process which is running in the VXP’s virtual environment from memory, the native method “refresh” is called to read the maps information. Depending on the registration information, the hook operation is then performed.

## 4 Evaluation

### 4.1 Functional Evaluation

For the functional evaluation of our native hooking method, we do the experiments on different apps to enhance their security. First, voice call encryption

function is achieved on two popular chatting apps: WeChat and Line, using this method. After analyzing the SO files related to VOIP using the IDA tool, we get the appropriate methods as hook points. Another experiment is done based on the Canon Camera Connect application. This application transmits the pictures from Camera to the phone. We achieve the picture encryption when writing the received data to the phone storage space by hooking the file manipulation methods. Table 1 shows the target methods of the three apps.

**Table 1.** Target methods of three apps.

Application	Hook method	Reference section	Whether can be hooked
WeChat	SendDataToChannel	.data.rel.ro	✓
	XVE_ReceiveRTPPacket	.data.rel.ro	✓
Line	Srtp_cipher_encrypt	.got	✓
	Srtp_cipher_decrypt	.got	✓
Canon Camera Connect	fopen	.got	✓
	fwrite	.got	✓

Table 1 shows that these target methods in three apps all can be hooked successfully using the method proposed in this paper to achieve the data protection purpose. Many underlying methods of Android applications are defined or used in shared libraries, which may be used by hackers to disclosure users' private data. Therefore, this native hooking method is able to enforce security policies and has wide usage.

## 4.2 Compatibility

To gain greater assurance in this method's compatibility, we did the same experiments on different Android devices, as shown in Table 2.

**Table 2.** Experimental results of Samsung and Huawei in different Android versions.

Phone	Samsung G960	HUAWEI Mate10pro	Samsung G973F
Operating System	Android OS 8.0	Android OS 8.0	Android OS 9.0
CPU	Exynos 9810	Kirin 970 2417MHz	Exynos 9820
CPU Architecture	Armeabiv-v7a	Armeabiv-v7a	Armeabiv-v7a
Whether Root	No	No	No
Compatible with VXP	Yes	Yes	Yes
Read Virtual Memory Mapping Information	Readable	Readable	Readable
Read Virtual Memory Space	Readable	Readable	Readable
Rewrite Virtual Memory Space	Writable	Writable	Writable
Native Hook Result	Success	Success	Success

The previous section conducted experiments on the Samsung G973F and the Android version is 9.0. The same experiments are achieved on the HUAWEI Mate10 Pro of Android 8.0 and the Samsung G960 of Android 8.0. The HUAWEI Mate10 Pro is equipped with different processors from Samsung. And the two Samsung phones have different Android versions. However, when the same experiment is done on other phones, the experimental result is the same.

The core of this native hooking method is to calculate the target method's reference address at runtime and rewrite it, so as long as the virtual memory which contains the target ELF can be read and written, this hooking method can be used. The VXP supports the Android version from 5.0 to 10.0. Therefore, we can use this app to run our hooking method without root permission on these different Android versions, and this hooking method is compatible.

## 5 Related Work

A family of native hooking methods is achieved by modifying the instruction in the entry of the function to the jump code. Frida [12], a dynamic instrumentation toolkit, can hook both Java and native methods. A server program is needed to install on phone to communicate with the script on the PC. Therefore, it is not suitable for practical production. Cydia [8] also supports hooking on both Java and Native. Frida and Cydia are not supported on non-root Android devices.

Aurasium [7] builds a reference monitor into application binaries and rewrites function pointers in a module's global offset table. Clearly, such approaches are not suitable for hooking the inner methods in shared libraries. Also note that when the instrumented application is repackaged, the package signature is broken. Mulliner et al. proposed PatchDroid [13] which is a system to distribute and apply third-party patches for security vulnerabilities for both Dalvikbyte-code and native code in Android. It patches for native code by performing inline hooking and hooking the global offset table and then injecting the shared library. Users can only install it on already rooted devices.

Several approaches aim to achieve hooking on ART Runtime. You et al. proposed TaintMan [14], an ART-compatible DTA framework that can be deployed on unmodified and non-root Android devices by instrumenting taint enforcement code into the target application and the system class libraries to track data flow and control flow. TaintART [15] proposed a multilevel information-flow tracking system. ART java method hook is more at the virtual machine level in Android and used to intercept Java methods.

App sandboxing is an important means of sandbox unmodified apps in non-rooted devices on stock Android. Michael et al. proposed Boxify [6], an approach based on application virtualization and process-based privilege separation to encapsulate unmodified apps in an isolated execution environment within the context of another. Bianchi et al., proposed NJAS [16], similar to Boxify, an approach to sandbox arbitrary Android applications by means of system call interposition.

## 6 Conclusion and Future Work

In this paper, we present a practical framework for hooking native methods in shared libraries in Android. This framework supports the method hooking without any modifications to both Android system and app's code by reference modification and control flows redirection and is thus easy to deploy. The final form of this method is shared object files that contain the core engine and the patch methods. We combine this method with VXP and install the target app inside the VXP's virtual environment, and the hooking operation can be performed without root privilege on any real devices. This framework can be used to analyze Android malware and implement security policies. Evaluation result indicates that there is a considerable amount of sensitive methods can be hooked and therefore this framework has wide usage. It also demonstrates that it can perform hook operation without a significant runtime performance overhead on real devices and it is compatible with different versions of Android devices and applications. Obviously, this native hooking method has its limitations and corner cases. The main limitation is due to only native methods which have reference can be hooked. We leave other native methods hooking for future work.

## References

1. Gasparis, I., Qian, Z., Song, C., Krishnamurthy, S.: Detecting android root exploits by learning from root providers. In: 26th USENIX Security Symposium, p. 1129–1144 (2017)
2. Yerima, S.Y., Sezer, S., Mcwilliams, G.: Analysis of Bayesian classification-based approaches for android malware detection. *IET Inf. Secur.* **8**(1), 25–36 (2014)
3. Sbirlea, D., Burke, M.G., Guarnieri, S., Pistoia, M., Sarkar, V.: Automatic detection of inter-application permission leaks in android applications. *IBM J. Res. Dev.* **57**(6), 10:1–10:12 (2013)
4. Rastogi, V., Chen, Y., Jiang, X.: Droidchameleon: evaluating android anti-malware against transformation attacks. In: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, ASIA CCS 2013, pp. 329–334. Association for Computing Machinery, New York (2013)
5. Spreitzenbarth, M., Freiling, F., Ehtler, F., Schreck, T., Hoffmann, J.: Mobile-sandbox: having a deeper look into android applications. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing. SAC 2013, pp. 1808–1815. Association for Computing Machinery, New York (2013)
6. Backes, M., Bugiel, S., Hammer, C., Schranz, O., Von Styp-Rekowsky, P.: Boxify: full-fledged app sandboxing for stock android. In: Proceedings of the 24th USENIX Conference on Security Symposium. SEC 2015, pp. 691–706. USENIX Association, USA (2015)
7. Xu, R., Saïdi, H., Anderson, R.: Aurasium: practical policy enforcement for android applications. In: Proceedings of the 21st USENIX Conference on Security Symposium. Security 2012, p. 27. USENIX Association, USA (2012)
8. Cydia substrate for android. <http://www.cydiasubstrate.com>. Accessed 2020 10 July
9. Virtualxposed. <https://github.com/android-hacker/VirtualXposed>. Accessed 10 July 2020

10. Lee, B., Lu, L., Wang, T., Kim, T., Lee, W.: From zygote to morula: Fortifying weakened ASLR on android. In: 2014 IEEE Symposium on Security and Privacy, pp. 424–439 (2014)
11. Goldberg, I., Wagner, D., Thomas, R., Brewer, E.: A secure environment for untrusted helper applications confining the wily hacker. In: Proceedings of the 6th Conference on USENIX Security Symposium, Focusing on Applications of Cryptography. SSYM 1996, vol. 6. p. 1. USENIX Association, USA (1996)
12. Frida.re. <https://frida.re>. Accessed 10 July 2020
13. Mulliner, C., Oberheide, J., Robertson, W., Kirda, E.: Patchdroid: scalable third-party security patches for android devices. In: Proceedings of the 29th Annual Computer Security Applications Conference. ACSAC 2013, pp. 259–268. Association for Computing Machinery, New York (2013)
14. You, W., Liang, B., Shi, W., Wang, P., Zhang, X.: TAINTMAN: an art-compatible dynamic taint analysis framework on unmodified and non-rooted android devices. *IEEE Trans. Depend. Secure Comput.* **17**(1), 209–222 (2020)
15. Sun, M., Wei, T., Lui, J.: Taintart: a practical multi-level information-flow tracking system for android runtime. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. CCS 2016, pp. 331–342. Association for Computing Machinery, New York (2016)
16. Bianchi, A., Fratantonio, Y., Kruegel, C., Vigna, G.: NJAS: sandboxing unmodified applications in non-rooted devices running stock android. In: Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices. SPSM 2015, pp. 27–38. Association for Computing Machinery, New York (2015)



# An Android Data Protection Scheme for System-as-Root Architectures

Kai Yang<sup>1</sup>, Ling Pang<sup>1</sup>(✉), Bo Zhang<sup>3</sup>, Gang Zhao<sup>2</sup>, and Xiaohui Kuang<sup>2</sup>

<sup>1</sup> School of Computer Science and Technology,

Beijing Institute of Technology, Beijing 100081, China

1406996437@qq.com, pangling\_nkl@163.com

<sup>2</sup> National Key Laboratory of Science and Technology on Information System Security,  
Beijing 100101, China

zhao-gang20@126.com, xiaohui\_kuang@163.com

<sup>3</sup> Information Center of Ministry of Human Resources and Social Security,  
Beijing 100716, China

imagepool@126.com

**Abstract.** At present, most of the new Android devices on the market use the system-as-root architecture. Therefore, the security of personal privacy data under this architecture has become a very concerned issue for the majority of people. As for the problem, this paper proposes to build a safe ecosystem by customizing TF card. When users deal with privacy data, our scheme creates a safe mode based on TF card, and store privacy data in the safe mode. In this way, users can protect their privacy data in the safe modes, even if the mobile is stolen by other people. In order to illustrate the security of the safe mode, this paper does some experiments to evaluate the performance overhead between the normal mode and the safe mode. The experiments show that the performance overhead of this scheme is reasonable and can effectively reduce the risk of sensitive information leakage.

**Keywords:** Data protection · Android · Micro SD card · System-as-root architectures

## 1 Introduction

As the most popular smartphone platform in the world [1], Android is vulnerable to hackers due to its vulnerability and complexity [2]. Connecting insecure WiFi hotspot [3], scanning malicious QR code [4], clicking malicious network links [5], and other issues have brought serious challenges to privacy data protection [6]. Nowadays, most Android devices use the system-as-root structure, people pay more attention to the security of this new architecture than before.

In order to deal with these threats under the architecture, we analyzed the boot process of Android and the structural characteristics of the TF card. It is found that the TF card can be divided, and then the data partition and cache partition can be reset to the TF card by the way of partition reset, so as to realize the safe mode.

The contribution of this paper is as follows:

- This paper realizes the data protection scheme based on TF card which has the functions of stable and fast switching in two modes, performance optimization in safe mode, data isolation in two environments.
- This paper discusses and realizes the bypass method of Android 9.0 verification mechanism.

The rest of this article is organized as follows: Sect. 2 introduces the related works. Section 3 introduces the design scheme of the safe mode. Sections 4 and 5 shows the implementation details and evaluates the performance of the scheme. Section 6 introduces the limitations and future work. Section 7 is the final conclusion.

## 2 Related Works

In this section, we will introduce the current schemes for data protection [7].

Android introduces FDE to automatically encrypt the data in the disk and decrypt the data when reading it, so as to protect the private data [8]. Meng Fanjiao separates some storage space of Android device as encrypted data partition, and applies dm-crypt encryption technology to automatically mount encrypted data partition when Android system starts [9]. In encrypted data partition, users can store their privacy data and encrypted data partition for data storage.

Trustshadow allows unmodified applications to run in the security area to isolate applications from untrusted REE kernels [10]. The TEE kernel does not process system calls from protected applications. Instead, it redirects all system calls to the REE kernel for processing.

Xue Yuan proposes a method of multi boot Android operating system based on OTG devices to meet the needs of users in different scenarios [11]. This method uses system domain isolation to ensure the security of privacy data on different Android operating systems [12].

However, there are some defects in these researches [13]. Most of the security solutions are implemented in the framework layer or application layer [14], while our solution is protected by TF card, and data isolation can be realized on hardware.

## 3 Design

In this section, we analyze the difficulties in the process of implementation, introduce the design ideas, and give the feasibility analysis.

### 3.1 System-as-Root Partition Layout

Since May 2018, all devices equipped with the new operating system version have system-as-root partition layout. Consider with this new architecture, this paper plans to create a secure space in TF card. When users need to process their privacy data, they can enter the safe mode to operate; when users exit, the mobile will return to the normal mode. Therefore, it is necessary to learn and analyze the environment switching under different modes and TF card space partition.



### 3.2 Mount Process of Partition After Booting

Usually, after the Android device is powered on, the bootloader is executed to boot the hardware initialization, configure the corresponding running environment, and prepare for the start of the operating system. After that, the Android system loads the ramdisk and kernel, enters the entrance of the kernel, and finishes the initialization of the remaining hardware, includes calling `set_task_stack_end_magic` (&init task) in `start_kernel` function to set the first process named `zygote` process of the system. This process runs the `init` program in ramdisk. The `init` program starts the process or service according to the rules of `init.rc` file, and completes the work of creating mount point, mounting file system, starting attribute service, etc.

In the mobile phone with new architecture, by checking the `init.rc` file in the root directory, we can find the related service of mounting file system and starting the core system.

We can see that `init.rc` will start the `late-init` trigger to trigger `fs`. Corresponding to the `init. { $device }. rc` file, we found the trigger `fs` which called the `mount_all` command: `mount_all/vendor/etc/fstab`. Then we find `{ "mount_all", { 1,kMax, { false, do_mount_all } } }` code in `system/core/init/builtins.c`. The command `mount_all` corresponds to the `do_mount_all` function which calls the `mount_fstab` function, and the `mount_fstab` function finally calls the `fs_mgr_read_fstab` function to parse the partition file and completes the mounting of the file system. This means the `mount_all` command mounts the corresponding file system in the `fstab` file.

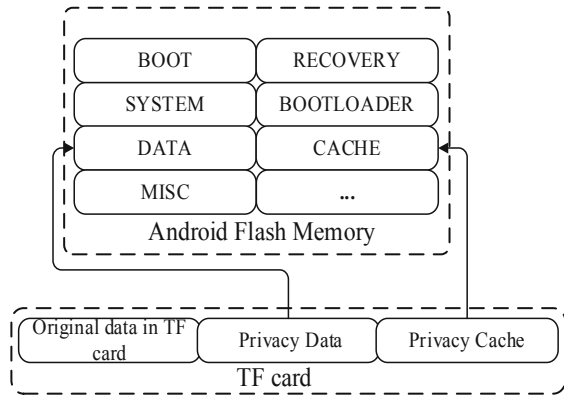
### 3.3 Customization of TF Card

After experiments, we find that the partitions related to data protection are data partition and cache partition. Data partition mainly includes user's information, application program and data. Cache partition saves the most frequently accessed data. In order to mount these partitions to the customized TF card, we need to manually partition the TF card and initialize the file system configuration.

For this reason, this paper considers customizing three partitions on the TF card: external expansion card partition, privacy data partition and privacy cache partition. They are respectively used to store data in the original TF card, user privacy data and cache data generated in private mode. The external expansion card partition is used as the external expansion card to store the data of the original TF card in order not to affect the normal use of the TF card (Fig. 1).

This paper decides to use the partition tool under Linux to realize the partition operation of disk. The size of the external expansion card partition and privacy data partition can be customized according to the actual needs of users, while the privacy cache partition should meet the minimum capacity of the mobile. According to the partition formats seen on the normal mode, we know that the file formats of data partition and cache partition are `ext4`, while the file format of external expansion card is `exfat`. Therefore, the `mkfs` command is needed to initialize the newly divided partitions.

After the partition is completed, we can check the information through the first 512 bytes of MBR partition table of TF card with WinHex tool.



**Fig. 1.** Design of customized TF Card.

## 4 Implementation

Based on the Android data protection method we proposed, this section will implement the data protection scheme. This scheme can realize the physical isolation between the modes and avoid the privacy data being stolen by attackers.

### 4.1 Bypassing Checksum

AVB is the start-up verification of Android devices to ensure the integrity of devices, which is usually started from the firmware. In AVB 2.0, the Android system adds a keystore (vbmeta.img) to protect the integrity of boot, system, vendor, dtbo and recovery partitions. The procedure is that bootloader verifies the signature in vbmeta by using the public key compiled in advance.

Because our scheme will modify kernel in the boot partition and fstab files in vendor partition, it is necessary to modify vbmeta.img to bypass the process of integrity verification of the partition. Vbmeta.img needs to be used in the bootloader stage when ROM code is verified, so we need to unlock OEM key first.

The vbmeta partition is divided into three parts: the first part of the header is 512 K. Through the header, we can know the signature of the algorithm, the size of hash, the size of signature, the location of the Authentication data block and other information. The second part is the Authentication data block, which size is given by header. It mainly contains hash data and signature data for verification. The hash data is the hash calculation of the header and Auxiliary data block, while the signature data is the signature after the hash data is padding. The third part is the Auxiliary data block, it contains a chain-type partition. The chained partition descriptor is used to authorize it to contain the name of the partition where the authorization is located, AVB\_Descriptor\_tag, and the public key named AvbRSAPublicKey trusted by the signature on this particular partition.

In order to bypass AVB checksum, there are two ways: one is to use avbtools to verify the AVB checksum, so as to generate a new vbmeta to replace the original vbmeta. The other is to modify the image to invalidate the integrity verification. We adopt the

second one, which is realized by modifying some bytes of the header. Therefore, we set the other parts of the header to 0 except the magic header and release string. Since the relevant messages of Authentication data block and Auxiliary data block are set to 0, we only need to keep the first 512 bytes of header information, so that we can bypass the integrity verification function.

### 4.2 Switching the Modes

According to Sect. 3.2, we need to change the mount point on the fstab file to the path of the corresponding partition of the customized TF card, and then rewrite the modified fstab file to the path of/vendor/etc. The data and cache partition will be replaced with the safe space of the TF card after the mobile restarts (Fig. 2).

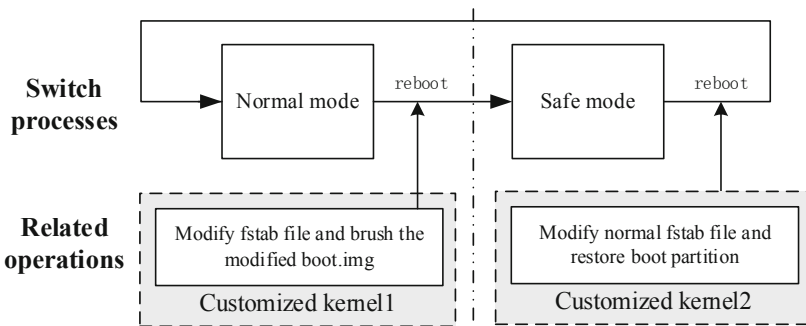


Fig. 2. Switching process between two modes.

Considering the high permission required for the mutual switch between the modes and the open-source characteristics of the kernel, this scheme decides to implement the switch operation by customizing the kernel.

In order to achieve the security target, we monitor the renaming file system in the kernel. If the renamed name is the specified password we set, the operation will be triggered, which can prevent the user from misoperation and prevent the attacker from entering the safe mode. The specific process is that when the user renames a file, the `vfs_rename` function will judge the renaming operation of the file. If the file name is the password to enter the safe mode, the above operations will be performed.

In `/fs/readwrite.c`, the main body of the read/write operation is implemented through the `vfs_read` and `vfs_write` function. It is worth noting that the data in user space should be copied to the kernel space by using the `copy_from_user` function, and then operate the data. The modified data should also be copied to the user space by using the `copy_to_user` function.

## 5 Evaluation

In order to have a intuitive understanding of the functionality and security of this scheme, we will test in this section and evaluate the experimental results. The parameters needed

for the experiment are Samsung Galaxy S10 (Android version is 9.0.0) and Lexar Micro SD Card (which speed is 633x and space is 256G).

### 5.1 Evaluation of Data Isolation Performance

In order to check the isolation result, we use the ADB tool to debug, and then use the mount command to check the mount path.

Through experiments, we find that two block devices named `/dev/block/mmcblk0p2` and `/dev/block/mmcblk0p3` under Android are attached to the data and cache partition, and these two block devices are corresponding to the privacy data partition and privacy cache partition we divided on TF card. This shows that our scheme has been successfully verified on the device.

In order to intuitively observe the effect of data isolation, we do the following experiments, respectively put recordings, pictures, apps, etc. in different modes. In addition, we need to modify the desktop background under safe mode to distinguish different modes.

**Table 1.** The experimental results of our proposed method.

	Stored in normal mode		Stored in safe mode	
	Normal mode	Safe mode	Normal mode	Safe mode
Recordings	Y	N	N	Y
Pictures	Y	N	N	Y
Videos	Y	N	N	Y
Apps	Y	N	N	Y
Messages	Y	N	N	Y
Contracts	Y	N	N	Y

As shown in Table 1, Y represents existence and N represents not existence. The experiment results are consistent with expectations, which verifies the security of our scheme.

### 5.2 Performance Evaluation

In order to evaluate the performance of our scheme, we select the official testing software named LuDaShi for testing.

As shown in Table 2, the CPU, GPU, memory, flash memory and other parameters are almost the same between the two modes, but there is a lack of storage performance in the safe mode. This is due to the performance problem of the micro SD card we selected: the read/write speed of 1x is about 0.15mb/s, so 633x is about 94 MB/s. According to the evaluation results, we found that the read speed of the mobile is about 10 times that of the TF card, while the write speed is about 2 times that of the TF card. Therefore, in the safe mode, we should not read or write too large files. In order to check the influence of our scheme on the read/write speed, dd command is used to test the read/write speed of the system.

**Table 2.** Comparison of various properties with LuDaShi software.

Parameter	Normal mode	Safe mode
CPU performance	117280	98870
GPU performance	124442	123796
RAM performance	41020	40820
Flash memory performance	31334	3233
Reading speed	847 MB/S	46 MB/S
Writing speed	178 MB/S	72 MB/S
Database	4802	1140

As shown in Table 3, there is no significant difference in read/write speed between the two modes when processing files smaller than 1G.

**Table 3.** Comparison of read/write speed between normal mode and safe mode.

	Read speed		Write speed	
	Normal mode	Safe mode	Normal mode	Safe mode
64 M	0.07 s	1.28 s	0.38 s	1.53 s
128 M	0.17 s	2.06 s	0.69 s	2.7 s
256 M	0.28 s	4.45 s	1.44 s	4.67 s
512 M	0.71 s	12.33 s	2.81 s	8.80 s
1G	1.93 s	22.06 s	8.01 s	15.10 s

### 5.3 Overhead Evaluation

Considering the overhead of the system in the safe mode, we choose the parameter of boot time to research:

As shown in Fig. 3, we can see that there is a certain difference in boot time between the two modes, but the difference is not significant, which should be tolerable by users.

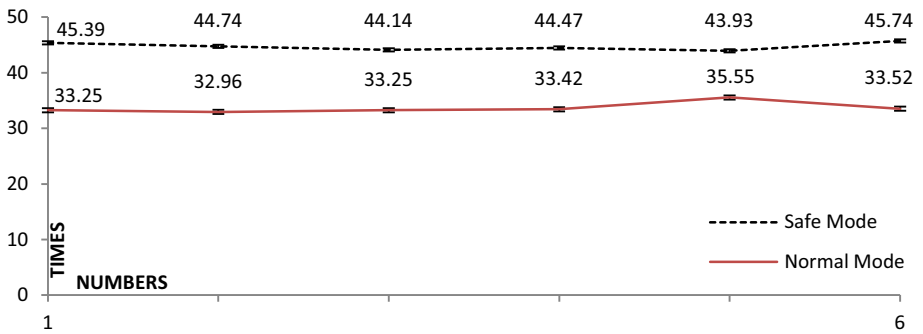


Fig. 3. Comparison of start-up time in two modes.

## 6 Limitation and Future Work

From the experiment in the previous section, we can see that our scheme can achieve the target that physical isolation of data between modes, so as to achieve the purpose of data protection. But there are some defects and deficiencies in our design.

First of all, we use the customized kernel solution for modes switching, our scheme has poor flexibility of the switching, and the upgrade of small version may lead to the failure of the switching. Secondly, our scheme has not been tested comprehensively in the actual attack cases, so we cannot get the robustness of our scheme.

In the next step, we decide to complete the mode switching without modifying the kernel to solve the problem of the design. At the same time, we will test all kinds of attack methods in the future, detect the scheme for all kinds of attack methods, and improve the scheme according to the experiments.

## 7 Conclusions

This paper introduces a safe area based on TF card on mobile devices. Using TF card to realize two isolated data spaces and deploy the safe area on TF card. The scheme can freely and safely switch between safe space and normal space, so as to achieve the purpose of data protection. Experimental results show that this method effectively reduces the risk of sensitive information leakage, and the cost of performance is reasonable.

**Acknowledgment.** This work is supported by National Natural Science Foundation of China under Grant No. 61876019.

## References

1. Bahis, K.: Mobile internet connection status in 2018. *GSM Assoc.* **2**(5), 1–63 (2018)
2. StatCounter. <https://gs.statcounter.com/os-market-share,last>. Accessed 1 Jan 2020
3. Liang, C., Tan, Y., Zhang, X., Wang, X., Zheng, J., Zhang, Q.: Building packet length covert channel over mobile VoIP traffics. *J. Network Comput. Appl.* 144–153 (2018)
4. Liang, C., Wang, X., Zhang, X., Zhang, Y., Sharif, K., Tan, Y.: A payload-dependent packet rearranging covert channel for mobile VoIP traffic. *Inf. Sci.* **465**, 162–173 (2018)
5. Gu, J., Li, C., et al.: Combination attack of android apps analysis scheme based on privacy leak. In: *Proceedings of 2016 4th IEEE International Conference on Cloud Computing and Intelligence Systems. CCIS 2016*, pp. 62–66 (2016)
6. Fanjiao, M., et al.: A high efficiency encryption scheme of dual data partitions for android devices. In: *Proceedings - 2017 IEEE International Conference on Computational Science, CSE and EUC 2017*, vol. 1, pp. 823–828 (2017)
7. Anthony, S., Feng, L.: Android smartphone third party advertising library data leak analysis. In: *Proceedings - 11th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2014*, pp. 749–754 (2015)
8. Tan, Y., Zhang, X., Sharif, K., Liang, C., Zhang, Q., Li, Y.: Covert timing channels for iot over mobile networks. *IEEE Wirel. Commun.* **25**, 38–44 (2018)
9. Chen, L., Wang, X., Zhang, X., Zhang, Y., Sharif, K., Tan, Y.: A payload-dependent packet rearranging covert channel for mobile VoIP traffic. *Inf. Sci.* 162–173 (2018)
10. Maier, D., Protsenko, M., et al.: A game of Droid and Mouse: the threat of split-personality malware on Android. *Comput. Secur.* **54**, 2–15 (2015)
11. Wu, Q., Zhao, C., Guo, Y.: *Android Security Mechanism Analysis and App Practice*, 2nd edn. The China Machine Press, Beijing (2013)
12. Maier, D., Protsenko, M., et al.: A game of Droid and Mouse: the threat of split-personality malware on Android. *Comput. Secur.* **54**, 2–15 (2015)
13. Guan, Z., Si, G., Zhang, X., Wu, L., Guizani, N., Du, X., Ma, Y.: Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities. *IEEE Commun. Mag.* **56**(7), 82–88 (2018)
14. Guan, Z., Zhang, Y., Zhu, L., Wu, L., Yu, S.: Effect: an efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid. *Sci. China Inf. Sci.* **62**, 1–14 (2019)



# A Two-Fitness Resource Scheduling Strategy Based on Improved Particle Swarm Optimization

Xueming Qiao<sup>1</sup>, Meng Chen<sup>2</sup>, Xiangkun Zhang<sup>1</sup>, Weiyi Zhu<sup>3</sup>, Yanhong Liu<sup>1</sup>,  
Zhixin Huo<sup>4</sup>, Ruiqi Sun<sup>4</sup>, and Dongjie Zhu<sup>4</sup>(✉)

<sup>1</sup> State Grid Weihai Power Supply Company, Weihai, China

<sup>2</sup> State Grid of China Technology College, Jinan, China

<sup>3</sup> State Grid Shandong Electric Power Company, Jinan, China

<sup>4</sup> School of Computer Science and Technology, Harbin Institute of Technology,  
Weihai, China

zhudongjie@hit.edu.cn

**Abstract.** The performance optimization of cloud platform for big data processing is a research hotspot, among which resource scheduling is the most important. Through the analysis of the internal resource scheduling mechanism of Cloud-Stack, the two-level scheduling of resources plays an important role in task optimal span, load balance and other aspects. In this paper, aiming at optimizing IaaS service performance and taking CloudStack platform as the research object, a dual fitness resource scheduling strategy based on improved particle swarm optimization is proposed. First of all, PSO algorithm with high precision and fast convergence speed is used to optimize the two-level resource scheduling, which can shorten the scheduling time when the scheduling requirements are met. Secondly, aiming at the problem of “prematurity” of particle swarm optimization (PSO), this paper USES simulated annealing algorithm to optimize the traditional PSO. Finally, aiming at the two pole resource scheduling, this paper proposes the virtual machine deployment algorithm based on improved particle swarm and the dual fitness task scheduling algorithm based on Improved Particle Swarm respectively, and carries out simulation in CloudSim simulation tool. The simulation results show that the algorithm proposed in this paper can effectively improve the optimal span and optimize the load balance.

**Keywords:** Cloud computing · Resource scheduling · Improved particle swarm · IaaS · CLOUDSTACK

## 1 Introduction

As a widely used architecture, the data access efficiency of cloud platform is the key factor to ensure the operation efficiency of the platform. However, the increasingly complex logical business and application scenarios make great changes in data access characteristics. Traditional NAS and other storage architectures and access policies have



been unable to meet the needs of data access, resulting in uneven load and high latency. Therefore, relevant researchers have optimized access in data prefetching [1, 2], grouping [3] and other technical aspects. As one of the main architectures of cloud platform, spark includes scheduling service, indexing service, caching service, statistics service, sorting service, etc. the effectiveness of resource scheduling becomes the key factor of its performance optimization effect. Resource scheduling algorithm is the mainstream method in computer network technology, which has been widely used in the field of IOT and data center optimization [4]. For example, Qian Li [5] proposed an optimization model for SLA based resource scheduling, and provided a solution using random integer programming technology. LV Lianggan [6] proposed a trust model and a trust driven resource load balancing scheduling algorithm for the resource load degree and scheduling time span of the scheduling itself, and implemented the algorithm in MATLAB simulation, and then designed a prototype system to test the algorithm in practice, which provides a data reference for the experiment and improvement of the algorithm.

For the cloud platform, appropriate load balancing and task scheduling strategies enable each node to be in the best processing capacity, and the computing capacity can also be customized according to the processing requirements to achieve “on demand requirements”. For example, Xin Lu et al. [7] improve the capability of cloud computing on-demand service by studying the dynamic scheduling of cloud resources and propose a load adaptive scheduling model based on ant colony algorithm. By monitoring the performance of virtual machine, real-time scheduling resources can meet the changing load demand. Bee colony algorithm is also an effective intelligent algorithm. Yao Jing et al. [8] introduced the bee colony algorithm into the study of load balancing, and proposed the bee colony algorithm based on load distribution and the improved adaptive bee colony algorithm, which solved the problem of server throughput well, and pointed out that the former’s system stability and the latter’s system scalability are advantages.

For two-level resource scheduling, Cheng et al. [9] established a two-level resource scheduling model for cloud computing and proposed a minimum energy consumption resource scheduling algorithm based on genetic algorithm. The algorithm simulated the total completion time and total energy consumption of tasks, pointed out the feasibility of the algorithm, and showed the ability of genetic algorithm in resource scheduling. Li et al. [10] proposed a dual fitness genetic algorithm based on the genetic algorithm, which takes the shortest execution time of the total task and the shortest execution time of the task tie as the standard to schedule. For the two-level resource scheduling algorithm, this paper proposes the virtual machine deployment algorithm based on particle swarm and the dual fitness task scheduling algorithm based on particle swarm, and then through the CloudSim simulation tool to verify the results, test the effectiveness of the algorithm.

This paper studies the internal structure of CloudStack open-source cloud platform, analyzes the resource scheduling and task scheduling algorithms used by the platform, hoping to improve or introduce the latest algorithm theory and technology to replace the original method, so as to improve the operation efficiency of the open-source cloud platform, so as to provide strong support for the promotion and use of CloudStack.

## 2 Related Work

In recent years, the task scheduling and resource allocation of cloud platform has been a key research issue for researchers. In terms of resource allocation, Li et al. [11] proposed two online dynamic resource allocation algorithms to deal with preemptive tasks in IAAs cloud system. The algorithm dynamically allocates tasks according to the update of actual task operation information. Besides ant colony and colony algorithm, more and more researchers used the particle swarm optimization (PSO) algorithm, Arlindo etc. [12] in order to keep the group of diversity, and prevent the premature convergence is proposed with predator-prey particle swarm optimizer optimization algorithm, the algorithm was analyzed and the other a variety of standard particle swarm algorithm, the result of the algorithm is implemented in CloudSim simulation tools, and the results show that the improved task completion time reduced to a certain extent, for in this paper, the particle swarm algorithm for resource scheduling research laid the foundation. Then, the model of resource scheduling is put forward, which abstracts the problem into mathematical model, and promotes the development of resource scheduling algorithm. For example, Shan Hongbo et al. [14] combined the genetic algorithm with the simulated annealing algorithm, introduced the simulated annealing operation after the genetic mutation stage, and improved the premature convergence of the genetic algorithm; Li et al. [15] proposed a task scheduling based on the optimized ant colony algorithm, so as to minimize the optimal span and average span of task time. But there is little analysis of open source cloud platforms other than algorithms. Peng [16] analyzed the resource structure and virtual machine deployment mechanism of CloudStack, and used the CloudSim simulation tool to simulate a virtual machine deployment scheme based on users' personalized needs. Based on the CloudStack cloud management platform, she developed a billing system. This paper provides a powerful help for the study of CloudStack internal scheduling algorithm and resource management strategy.

## 3 Design

### 3.1 Virtual Machine Deployment Algorithm Based on Improved Particle Swarm Optimization

**Virtual Machine Deployment Design Based on Improved Particle Swarm Optimization.** There are generally two levels of resource scheduling in cloud computing: one is to schedule tasks to virtual machines in the virtual resource layer; the other is to deploy virtual machines to hosts in the physical resource layer, as shown in Fig. 1(a). In the figure, “Vm1 => Hostn” means to deploy the first virtual machine to the nth physical machine. It can be seen from the analysis that the computing capacity, memory capacity and storage capacity of the cloud platform server are far beyond the use requirements of ordinary users, so virtual machines are used to meet the daily needs of users. However, most of the popular open-source cloud platform virtual machine deployment methods are to simplify the deployment process without considering the problem of load balancing, which leads to the waste of cloud platform resources. Considering the differences of physical machine configuration and virtual machine resource requirements, this paper proposes an improved particle swarm optimization algorithm

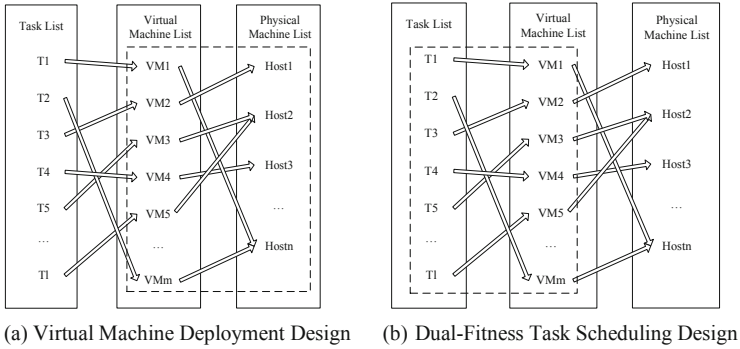


Fig. 1. The overall design.

to calculate the virtual machine deployment strategy, and uses the particle swarm optimization algorithm optimized by simulated annealing algorithm to calculate the better virtual machine deployment strategy. Particle swarm optimization algorithm has the advantages of fast convergence speed and wide search range. When all virtual machine deployment requests come, the deployment strategy is calculated quickly. At the same time, simulated annealing algorithm is introduced to avoid the result falling into the local optimal solution, which makes the physical machine achieve the effect of resource load balancing.

**Fitness Function.** The improved particle swarm optimization method needs different fitness functions for different problems. Each particle needs to recalculate its fitness after each position movement. First, the local optimal solution recorded by the particle itself is changed through the fitness, and then the global optimal solution is changed. The fitness function is not unique. It needs specific analysis for specific problems, and the fitness function is very important for the effectiveness of particle swarm optimization.

Based on the analysis of the factors that affect the deployment of virtual machine, this paper puts forward the fitness function to judge the load imbalance according to the factors such as computing power, memory and bandwidth of virtual machine. Formula (1) to formula (4) is as follows:

$$F_1(x) = \sqrt{\frac{1}{n} \sum_{i=1}^n (c_i - c_{avg})^2} \tag{1}$$

$$F_2(x) = \sqrt{\frac{1}{n} \sum_{i=1}^n (m_i - m_{avg})^2} \tag{2}$$

$$F_3(x) = \sqrt{\frac{1}{n} \sum_{i=1}^n (bw_i - bw_{avg})^2} \tag{3}$$

Where  $c_i$  is the utilization ratio of computing power of the  $i^{\text{th}}$  physical machine,  $c_{avg}$  means the average utilization ratio of computing power of all physical machines;

$m_i$  represents the memory utilization ratio of the  $i^{\text{th}}$  physical machine,  $m_{avg}$  means the average utilization ratio of memory of all physical machines;  $bw_i$  represents the bandwidth utilization ratio of the  $i^{\text{th}}$  physical machine,  $bw_{avg}$  means the average utilization ratio of bandwidth of all physical machines.

$$F(x) = \frac{1}{n} \sum_{i=1}^n \alpha_i F_i(x) \quad (4)$$

Where  $\alpha_i$  represents the influencing factor of the  $i^{\text{th}}$  fitness. The load unevenness of CPU, memory and bandwidth is calculated through the formula (4), and then the weighted average value of the three kinds of unevenness is calculated. This value is not only used as the overall load unevenness and fitness function value, but also as the judgment standard of the particle.

The load balance degree of resources is to allocate virtual machines to physical machines and check whether the resources consumed by each physical machine are balanced. If the resource utilization rate of one physical machine is too high and that of the other physical machine is too low, the operation efficiency of the former host will be reduced, and the latter host will be idle. Because there are differences in the performance of physical machines, this paper takes the resource occupancy rate as the calculation standard to compare the resource load balance of physical machines, and the common method to compare the differences is statistical standard deviation. The standard deviation indicates that the occupancy rate of each resource is relatively close, indicating that the allocation mode is balanced. On the contrary, it is unbalanced. Therefore, it can be considered that the fitness function,  $F_1(x)$ ,  $F_2(x)$ ,  $F_3(x)$  can be used as a standard to evaluate the load balance of CPU, memory and bandwidth, and the result obtained can be used as the load imbalance of these three factors.

**Algorithm Flow and Complexity.** Using particle swarm optimization algorithm to deploy the virtual machine, each particle in the particle swarm represents a deployment scheme, updating the optimal deployment scheme through the movement of a large number of particles. The difference between the improved particle swarm optimization and the traditional particle swarm optimization is that there is a certain probability to choose the particles with poor adaptability in the choice of new particles.

Steps of Algorithm:

The whole population is initialized according to the set of virtual machine and host, and the initial value and velocity value of each particle are randomly assigned. The local optimal solution is defined as the initial position, and the global optimal solution can be obtained by comparing the optimal solutions of all particles.

Calculate the speed and position of the updated particles and calculate the particle fitness according to the fitness function, that is, the load imbalance corresponding to the particles.

The local optimal solution of particles is recalculated and simulated annealing algorithm is introduced to screen the non-optimal local optimal solution and update the global optimal solution.

After a certain number of updates or when the change of the optimal solution is less than a certain threshold value, the execution is terminated, otherwise, the execution continues in step 2.

Output the optimal solution of the population, which is the final virtual machine deployment mode.

Analyze the time complexity of the algorithm, initialize  $m$   $n$ -dimensional particles, the time complexity is  $O(MN)$ ; update the speed and position of particles, the time complexity is  $O(MN)$ ; calculate the complexity of particle fitness function generally is  $O(n)$ ; update the optimal solution of particles and population, the time complexity is  $O(m)$ ; judge the termination condition, the time complexity is 1. Suppose that the algorithm goes through  $K$  iterations and the time complexity is  $O(KMN)$ , that is to say, the time complexity of the algorithm is  $O(KMN)$ .

### 3.2 Dual-Fitness Task Scheduling Algorithm Based on Improved Particle Swarm Optimization

This section will study the task scheduling algorithm in the two-level resource scheduling, as shown in Fig. 1 b), where “ $T5 \Rightarrow vm3$ ” indicates that the fifth task runs on the third virtual machine. After all tasks are assigned to the virtual machine according to a certain task scheduling strategy, the virtual machines work in parallel. The total running time of all tasks depends on the longest running time of all virtual machines. A good task scheduling strategy can make this time the shortest.

The improved particle swarm optimization algorithm is also suitable for task scheduling. The definition of particle in particle swarm optimization algorithm can correspond well with the solution of task scheduling. In order to optimize the traditional algorithm, which only guarantees the shortest task completion time but does not consider load balancing, this paper proposes that the overall load balancing effect among virtual machines should be considered in the algorithm, so that when the task scheduling strategy is used to allocate virtual machines, not only the overall task completion time is short, but also the load among virtual machines can be balanced, thus ensuring the cloud The stability of the platform improves the operation efficiency of the task.

#### **Task Scheduling Algorithm Based on Improved Particle Swarm Optimization.**

Based on the above, a task scheduling algorithm is proposed. The purpose of task scheduling is to calculate the optimal allocation mode from task to virtual machine according to the instruction length and space occupied by each task and the performance of virtual machine. Through the analysis of the internal task scheduling strategy of CloudStack, we know that it only uses the sequential allocation algorithm, which is fair to the virtual machine, and each virtual machine can receive the task allocation equally. However, due to the different configuration of virtual machines, the seemingly equal way will lead to the unbalanced load between virtual machines. If a large number of tasks are assigned to virtual machines with small computing power, the running time of the overall task will increase, and the load of the whole cloud platform will also be unbalanced. In this case, this paper proposes a task scheduling algorithm based on improved particle swarm optimization, which can calculate the optimal allocation of tasks to virtual machines, bind virtual machines to each task before task scheduling, predict the total running time

and load balance degree of tasks according to task information and virtual machine performance, and find out the optimal scheduling strategy.

**Fitness Function.** The difference between this algorithm and the algorithm used in virtual machine deployment strategy lies in the definition of fitness function. The task scheduling algorithm uses two fitness: one is the total execution time of the task, the other is the load balance of the virtual machine.

Initialize task scheduling particle swarm, each particle represents a task scheduling method, update particle position and speed according to particle swarm algorithm, and use double fitness function to judge when calculating particle fitness function:

$$F_1(x) = \max_{r=1}^l \sum_{j=1}^k Time(r, j) \quad (5)$$

According to the formula (5), where  $l$  represents the number of virtual machines,  $K$  represents all  $k$  tasks on virtual machine  $R$ , and time  $(R, J)$  represents the running time of the  $j^{\text{th}}$  task running on virtual machine  $R$ . Because tasks are executed serially on virtual machines and parallel among virtual machines, the total execution time of tasks is to find the maximum running time of a virtual machine.

$$F_2(x) = \sqrt{\frac{1}{l} \sum_{i=1}^l (m_i - m_{avg})^2} \quad (6)$$

According to the formula (6), where  $l$  is the number of virtual machines,  $m_i$  is the total memory usage of all tasks in the  $i^{\text{th}}$  virtual machine, and  $m_{avg}$  is the average memory usage of all virtual machines. There is no memory usage here, because the task is assigned to the virtual machine, and the task is executed serially, only one task is executing at a time, so the memory usage is meaningless, and it is likely that the total memory of all tasks on the same virtual machine is greater than the memory of the virtual machine. Therefore, the total amount of memory allocated to all tasks on each virtual machine is taken as the standard when calculating the load balancing.

The final fitness function of particles is defined as the formula (7):

$$F(x) = 1 / (\alpha / F_1(x) + \beta / F_2(x)) \quad (7)$$

Among them, the parameters  $\alpha$  and  $\beta$  respectively represent the influence factors of fitness, and different values can be set to make one of them more decisive. The quality of particles is determined by these two fitness, which not only ensures the shortest task completion time, but also ensures the load balance of virtual machine.

**Algorithm Flow and Complexity.** In the task scheduling algorithm, the position of particles represents the way of task scheduling. The fitness function is calculated by task execution time and load imbalance, so as to find the best way of task scheduling, which makes the execution time of the total task shortest and the load imbalance of task allocation on the virtual machine minimum.

Algorithm steps:

- 1) The particles are initialized with task set and virtual machine list, and the initial velocity and position of particles are generated randomly. The local optimal solution of particles is the initial position.
- 2) Through the formula, update the speed and position of particles, calculate the particle fitness according to the fitness function.
- 3) The local optimal solution of particles is recalculated and simulated annealing algorithm is introduced to screen the non optimal local optimal solution to update the global optimal solution.
- 4) Determine whether the end condition is reached at this time, otherwise skip back to step 2 to continue.
- 5) End algorithm. Output the optimal solution of the population, that is, the optimal way of task scheduling.

The steps of the algorithm are similar to the virtual machine deployment, so the time complexity is similar. Assuming that the number of tasks is  $m$ , the number of virtual machines is  $n$ , and the algorithm iterates  $t$  times in total, the time complexity of the algorithm is:  $O(TMN)$ .

## 4 Simulation of Resource Scheduling Algorithm

In order to verify the effectiveness of the above algorithm, this paper uses CloudSim simulation tool to simulate, on the basis of other algorithms, write improved algorithm to simulate resource scheduling, and compare the specific data to analyze and verify the algorithm in this paper. Simulation experiment is divided into two parts: first, simulation virtual machine deployment algorithm. In CloudSim, a virtual data center containing a certain number of physical machines and a certain number of virtual machines with different configurations are created. The deployment strategy calculated by the improved particle swarm optimization algorithm proposed in this paper is used to allocate the virtual machines, and the parameters of the physical machines after deployment are detected. The load balancing degree of all physical machines is calculated to verify the algorithm effect. Second, the experimental simulation task scheduling algorithm, simulation randomly generated a certain number and instruction length to the size of the task, using improved particle swarm algorithm proposed in this paper to calculate the task scheduling strategy task to run to the corresponding virtual machine, get the simulation results, through the completion of all tasks time length and the degree of load balance of all the virtual machine to verify the effect of the algorithm.

### 4.1 Simulation and Implementation

In order to verify the quality of the algorithm, this paper decided to use CloudSim simulation tools to verify the algorithm. At the same time, the algorithm in this paper was compared with the current commonly used algorithm to analyze the effectiveness of the algorithm.

### Algorithm Parameters.

#### 1. Virtual Machine Deployment Algorithm

First, set the parameters of the virtual machine and host, as shown in Table 1 and Table 2.

**Table 1.** List of virtual machines

Number of CPUs/unit	Computing Power/MIPS	RAM/GB	Network Bandwidth/Mbps
1	200	1	100
1	200	1	100
1	200	1	200
1	200	2	100
1	200	2	200
1	300	1	200
1	300	2	100
1	300	2	200
1	400	2	100

**Table 2.** List of physical machines

Number of CPUs/unit	Computing Power/MIPS	RAM/GB	Network Bandwidth/Mbps
8	1000	4	1000
8	1000	8	1000
4	2000	16	1000
4	2000	8	1000

The virtual machine and physical machine are initialized according to the Settings in the table, and the proxy provided by the CloudSim framework is submitted to the data center, where the Vm Allocation Policy PSO class is created to complete the deployment from the virtual machine to the physical machine. Determine the length of particles in this experiment is 9, and the number in each position of the particles represents the host to which the virtual machine is allocated. Decode the particles in one step to find the mapping from the virtual machine to the host, so as to calculate the allocation of resources and load balance on the host.

By formula, whether  $F(x)$  of all relevant factors a weighted average of the load imbalance degree, in front of the can by changing factors in the practical research of impact factor for the increase or decrease of a certain factor, the importance of this experiment is not specific to emphasize a factors, all factors are equally important, impact



factor 1. The results calculated by the formula are taken as the fitness of particles, and the load imbalance of the deployment mode can be obtained by judging the fitness of particle  $x$ , so as to determine whether the deployment mode is the optimal deployment mode. The algorithm ends when the specified number of iterations is exceeded.

In this experiment, the number of iterations is set to 1000. When the number of iterations exceeds the threshold, the global optimal solution is taken out, and the solution is used as the optimal deployment mode from virtual machine to physical machine.

## 2. task scheduling algorithm.

The initialization of the number of virtual machines USES a certain number in the experiment, so that the number of tasks changes without affecting the performance of the virtual machine, which is very useful for the comparison of results obtained by multiple tasks. The creation of cloud tasks is shown in Table 3. The task instruction length (MI) is shown in the table, which represents the number of millions of machine language instructions. Based on the execution speed of the virtual machine, the running time of each task can be calculated. The task size represents how much memory a task consumes and is used to calculate load balancing.

**Table 3.** Task List

Task command length/MI	Task size/MB	Task output size/MB
10000	300	100
20000	200	100
15000	200	100
...	...	...
20000	300	100

First, according to the task size and the computing power of the virtual machine, the ETC matrix of task execution time is obtained. ETC [I, J] represents the time required for the  $i^{\text{th}}$  task to run on the  $j^{\text{th}}$  virtual machine. The time of task operation can be obtained by the task instruction length/computing power.

When calculating the final fitness function value, it is necessary to determine the influence factors of the two fitness values. In this experiment, the total running time of the task is more important than the load balancing of the memory. Therefore, according to the formula, set  $\alpha$  to 4 and  $\beta$  to 1 in the actual simulation, which not only ensures the influence of the load balancing, but also improves the importance of the total running time of the task.

The end condition of the algorithm is designed to reach a certain number of iterations. The maximum number of iterations is set to 1000, and the number of iterations exceeds the threshold. At this time, the global optimal solution is taken out, and the solution is taken as the optimal mode of task scheduling.

## 4.2 Result Analysis

### Analysis of Virtual Machine Deployment Algorithm results.

#### 1. Load Balancing Effect.

In order to verify the effectiveness of the algorithm, the first matching algorithm of CLOUDSIM, the single fitness particle swarm optimization algorithm of each factor and the improved particle swarm optimization algorithm (SAPSO) proposed in this paper are simulated in the experiment, and the equalization degree of each factor obtained by the four algorithms are compared and analyzed which includes FM, CPU-SF, Memory-SF, SAPSO.

Through the comparison of the four algorithms, firstly, the imbalance degree  $F$  of each result is obtained through the simulation experiment, and  $y = 1 - F$  is defined as the load balance degree of the result. The higher the load balance degree is, the better the result can make the system achieve the load balance. The resulting load balance is shown in Fig. 2.

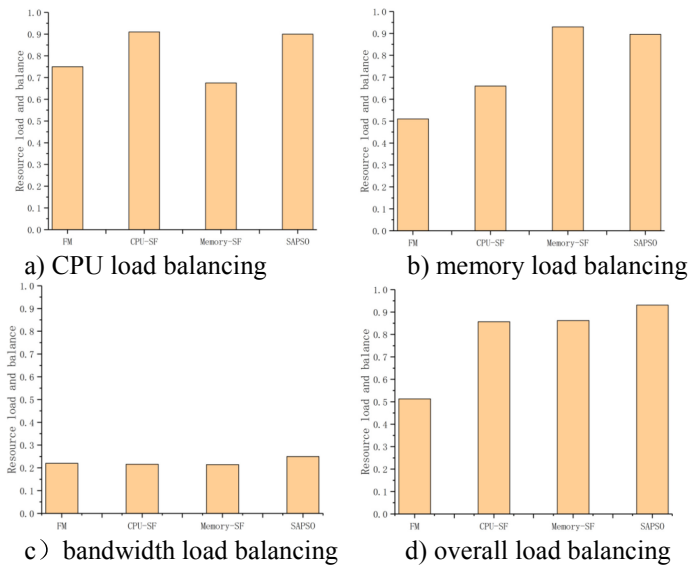


Fig. 2. Comparison of load balancing

From the image analysis, it can be seen that the overall load balancing degree of virtual machine deployment algorithm is significantly improved after using particle swarm optimization algorithm; the impact of physical machine load balancing is determined by many factors, and a single factor change can only affect the corresponding load balancing degree. The algorithm proposed in this paper uses CPU, memory and bandwidth as the criteria to calculate the load balancing degree, and the comprehensive effect is the best,

and the unilateral load balancing degree is also very good, which proves the effectiveness of the virtual machine deployment algorithm SAPSO proposed in this paper.

2. Change of Simulated Annealing Strategy and Number of Virtual Machines

In order to solve the “precocity” problem of particle swarm optimization, this paper introduces simulated annealing algorithm to improve the original particle swarm optimization.  $F$  is the calculated fitness, and  $y = 1 - F$  is defined as the load balance of the system, as shown in Fig. 3(a). The number of virtual machines and physical machines in the above experiment is determined, but this experiment only changes the number of virtual machines and physical machines, and observes the change of the overall load balance of the system. In the experiment, the traditional particle swarm optimization algorithm is compared with the improved particle swarm optimization algorithm proposed in this paper. The load balancing degree is defined as  $y = 1 - F$ , and  $f$  represents the fitness. The results are shown in Fig. 3(b).

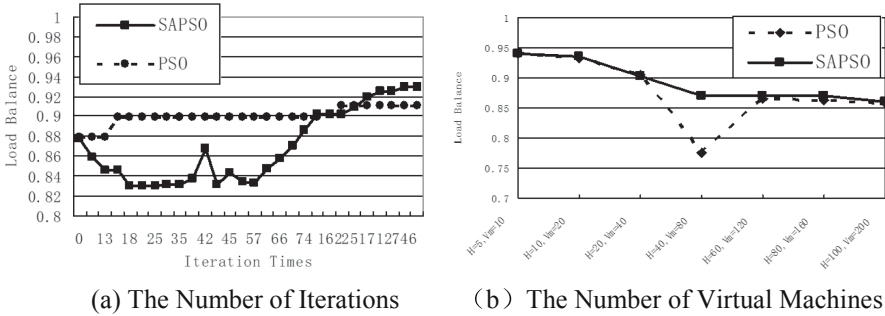


Fig. 3. Change of simulated annealing strategy and number of virtual machines.

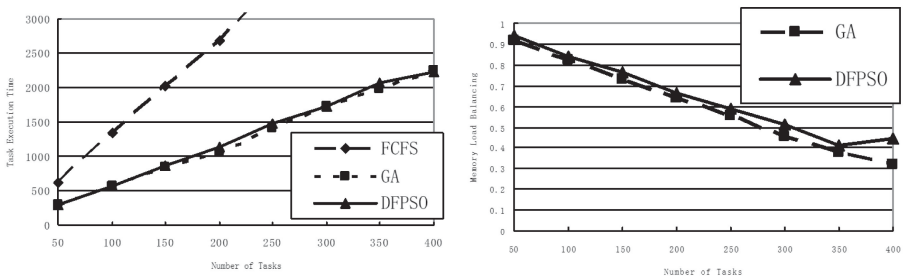
It is known from Fig. 3(a) that traditional particle swarm optimization algorithm will find the first global optimal solution and may not find the best global optimal solution. SAPSO, with the strategy of simulated annealing, adjusts its position all the time, searches from different directions, and finally finds the optimal solution which is closer to the global optimal solution than the traditional particle swarm optimization algorithm. As shown in Fig. 3(b), as the number of physical machines and virtual machines increases, the difficulty of distribution increases, and the load balance degree of the system decreases gradually. By contrast, when there are many virtual machines, the improved particle swarm optimization algorithm is better than the traditional one. From the above experiments, it is concluded that the SAPSO proposed in this paper can optimize the deployment strategy of virtual machine, and the effect of load balancing is more obvious, which proves the feasibility of this algorithm in virtual machine deployment.

**Result Analysis of Task Scheduling Algorithm.**

3. The Number of Tasks and Memory Load Balancing

In the experiment, there were a total of 400 tasks with a gradient of 50, randomly generating the instruction length between 10000–30000mi and the task length between 200–400mb. Task scheduling was carried out through the three algorithms of FCFS algorithm, greedy algorithm (GA), and particle swarm optimization algorithm (DFPSO) proposed in this paper, and the running time and load imbalance of memory were recorded and compared and analyzed after the completion of scheduling for each algorithm. The comparison results of the three algorithms are shown in Fig. 4(a). As can be seen from the figure, with the increase of the number of tasks, both the greedy algorithm and the algorithm proposed in this paper have much less execution time than the FCFS algorithm, and sometimes the task scheduling method obtained by the greedy algorithm is better.

FCFS does not consider load balancing, greedy algorithm simply realizes load balancing; the algorithm proposed in this paper considers both task execution time and load balancing. The fitness function value changes in the range of 1000-10000, so  $y = 1 - f/10000$  is defined as the load balance degree. The load balance degree of the greedy algorithm and the algorithm proposed in this paper is shown in Fig. 4(b). It can be seen that although the algorithm proposed in this paper is slightly worse than greedy algorithm in task execution time, it is obviously better than greedy algorithm in memory load balancing. It can also be seen from the figure that with the increase of the number of tasks, the load balancing degree decreases gradually, because in this paper, the standard deviation of the memory usage occupied by tasks is taken as the standard deviation. When the number of tasks is large, the slight deviation of memory will cause the standard deviation to increase a lot, resulting in the decrease of the calculated load balancing degree. Therefore, when the number of tasks is large, it does not mean that the load balance of the system must be less time difference than the number of tasks. The curve in the figure only has the effect of comparing with the load balance of greedy algorithm, which shows that the algorithm in this paper has more advantages than greedy algorithm in this respect when the number of tasks is the same.



**Fig. 4.** The result of tasks and memory load balancing.

Through the simulation experiment, we can see that the proposed dual fitness task scheduling algorithm based on particle swarm optimization is much better than CloudSim's default FCFS algorithm. Compared with greedy algorithm, the task execution time is sometimes less than greedy algorithm, but the load balance is better than greedy algorithm. The algorithm in this paper sacrifices the execution time of the task a little, and has a great optimization in load balancing, which is necessary. Therefore, in the task scheduling algorithm, the dual fitness task scheduling algorithm based on particle swarm optimization is effective and has a certain value.

## 5 Conclusion

Researchers optimize the cloud platform based on spark big data processing, which generally includes scheduling service, indexing service, caching service, statistics service, sorting service, etc., of which resource scheduling is the most important. By analyzing the internal resource scheduling mechanism of CloudStack, this paper puts forward the improvement of resource scheduling, and provides solutions for the load balance and the optimal span of tasks in the cloud platform. The main results of this paper are as follows: the two-level resource scheduling mechanism in the cloud platform is analyzed, and the particle swarm optimization algorithm is introduced to optimize the two-level scheduling algorithm. The simulated annealing algorithm is used to optimize the particle swarm optimization algorithm, and different fitness functions are formulated according to the actual problems. For the virtual machine deployment mechanism, an improved virtual machine deployment algorithm based on particle swarm optimization algorithm is proposed, using the load imbalance as the fitness function; for the task scheduling mechanism, a dual fitness task scheduling algorithm based on particle swarm optimization algorithm is proposed, using the optimal task span and load imbalance as the fitness function. Through the simulation of CloudSIM simulation tool, the results show that the algorithm proposed in this paper can effectively improve the optimal span and optimize the load balance.

**Acknowledgement.** This work is supported by State Grid Shandong Electric Power Company Science and Technology Project Funding under Grant no. 520613180002, 62061318C002, the Fundamental Research Funds for the Central Universities (Grant No. HIT. NSRIF.201714), Weihai Science and Technology Development Program (2016DX GJMS15), Key Research and Development Program in Shandong Provincial (2017GGX90103) and State Grid Shandong Electric Power Company Science and Technology Project Funding under Grant no. 520613200001.

## References

1. Zhu, D., Du, H., Sun, Y.: Massive files prefetching model based on LSMT neural network with cache transaction strategy. *Comput. Mat. Continua* **63**, 979–993 (2020)
2. Zhu, D., Du, H., Wang, Y.: An IoT-oriented real-time storage mechanism for massive small files based on Swift. *Int. J. Embed. Syst.* **12**(1), 72–80 (2020)
3. Zhu, D., Du, H., Cao, N., Qiao, X., Liu, Y.: SP-TSRM: a data grouping strategy in distributed storage system. In: Vaidya, J., Li, J. (eds.) *ICA3PP 2018*. LNCS, vol. 11334, pp. 524–531. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-05051-1\\_36](https://doi.org/10.1007/978-3-030-05051-1_36)

4. Zhu, D., Du, H., Sun, Y.: Research on path planning model based on short-term traffic flow prediction in intelligent transportation system. *Sensors* **18**(12), 4275 (2018)
5. Li, Q.: An optimal model and solution for resource scheduling in cloud computing. In: Proceedings of the 2012 International Conference on Computer Application and System Modeling, pp. 575–578. International Society of Computer Science and Electronic Technology, Shenyang (2012)
6. Lu, L.: Research on resource load balancing and scheduling algorithm in cloud computing environment. Master's thesis of Xinjiang University, Urumqi (2010)
7. Xin L, Gu Z, A load-adapative cloud resource scheduling model based on ant colony algorithm. In: Cloud Computing and Intelligence Systems, pp. 296–300 (2011)
8. Yao, J., He. J.: Load balancing mechanism of cloud computing based on adaptive swarm algorithm. *Computer Application*, September 2012
9. Cheng, C., Pan, Y., Zhang, D.: An energy-saving resource scheduling algorithm in cloud environment. In: Systems Engineering and Electronic Technology, November 2013
10. Li, J., Peng, J.: Task scheduling algorithm based on improved genetic algorithm in cloud computing environment. *Comput. Appl.* **31**, 184–186 (2011)
11. Li, J.Y., Mei Kang, Q., Zhong, M.: Online optimization for scheduling preemptable tasks on IaaS cloud systems. *J. Parallel Distrib. Comput.* **72**(2), 666–677 (2012)
12. Silva, A., Neves, A., Costa, E.: An empirical comparison of particle swarm and predator prey optimisation. In: O'Neill, M., Sutcliffe, Richard F.E., Ryan, C., Eaton, M., Griffith, Niall J.L. (eds.) AICS 2002. LNCS (LNAI), vol. 2464, pp. 103–110. Springer, Heidelberg (2002). [https://doi.org/10.1007/3-540-45750-X\\_13](https://doi.org/10.1007/3-540-45750-X_13)
13. Liu, P.: Research on task scheduling and resource allocation strategies in cloud computing environment. Shanghai: Master's thesis of east China normal University (2013)
14. Shan, H., Li, S., Gong, D.: Genetic simulated annealing algorithm-based assembly sequence planning. In: Technology and Innovation Conference, pp. 1537–1579 (2006)
15. Li, J., Peng, J., Cao, X.: A task scheduling algorithm based on improved ant colony optimization in cloud computing environment. *Energy Procedia* **13**, 6833–6840 (2011)
16. Hong, P.: Research and application of key technologies based on CloudStack cloud management platform. East China Univ. Sci. Technol., Shanghai (2013)



# Privacy-Preserving Nonlinear SVM Classifier Training Based on Blockchain

Nan Jia<sup>1</sup>, Shaojing Fu<sup>1,2(✉)</sup>, and Ming Xu<sup>1</sup>

<sup>1</sup> College of Computer, National University of Defense Technology, Changsha, China  
shaojing1984@163.com

<sup>2</sup> State Key Laboratory of Cryptology, Beijing, China

**Abstract.** The SVM classifier has been a significant and prevailing technique in machine learning applications. Training a high-quality SVM classifier usually requires a huge amount of data, which makes collaborative training by multiple parties become an inevitable trend. However, it causes privacy risks when sharing sensitive data with others. There are some existing methods to solve this problem. These methods mainly contain computation-intensive cryptographic techniques which are inefficient and not suitable for practical use. Therefore, it is important to realize efficient SVM classifier training while protecting privacy. In this paper, we propose a novel privacy-preserving SVM classifier training scheme based on blockchain. We establish a blockchain-based SVM classifier training mechanism which realizes collaboratively training while protecting privacy. We adopt the additive secret sharing technique to design several computation protocols, which are much more efficient than the schemes which contain complex cryptographic primitives. We conduct a thorough analysis of the security properties of our scheme. Experiments over a real dataset show that our scheme achieves high accuracy and practical efficiency.

**Keywords:** Privacy-preserving training · Nonlinear SVM classifier · Secret sharing · Blockchain

## 1 Introduction

Machine learning has been extensively involved in almost every aspect of human life and greatly changed our living styles. Classification, as an important machine learning technique to produce predictive models, plays an important role in various fields such as medical diagnosis [1], image recognition [2], recommendation system [3], and so on. Support Vector Machine (SVM) [4] is one of the most powerful machine learning methods which has been widely used for classification or prediction. For instance, the healthcare institutions collect the clinical records of patients and train SVM classifiers to establish a decision support system which can make accurate medical diagnosis. The image classification is another significant application of SVM which can help the e-commerce companies conduct effective commodity recommendation. In order to obtain an SVM

model with high accuracy, the training process usually requires a huge amount of data. However, it is difficult for a single party (e.g., a healthcare institution or a company) to collect plentiful and diversified data for training. In addition, the large volume of data and complex computation during the training process often exceed the storage and computation capacity of the resource-limited single parties. Therefore, there has been a surge in demand for collaborative training by a group of parties. On the one hand, the merged dataset from multiple sources has obvious advantages on data volume and variety to train a high-quality classifier. On the other hand, it greatly reduces the storage and computation overheads on the resource-constrained entities. However, it causes nonnegligible privacy risks when sharing the data with other untrusted parties [5,6]. For instance, the healthcare institution which wants to train an SVM classifier for medical diagnosis by utilizing the clinical records of the patients cannot directly share the records with others. Because the medical histories contain lots of sensitive information of the individuals [7]. The privacy of data could be invaded once they share them with other participants. Besides, the classification model is also private and valuable, the participants who make contributions to collaborative training are also unpleasant to reveal any information about the model. Therefore, it is a crucial problem to perform collaborative training while protecting privacy.

To address this issue, many cryptographic techniques have been adopted to achieve privacy-preserving classifier training, such as homomorphic encryption [8,9], differential privacy [10,11], and so on. However, the homomorphic encryption technique is usually involved with computationally expensive cryptographic primitives, which result in heavy computation cost. The differential privacy technique cannot achieve high accuracy because it protects data privacy by adding immeasurable noises to the parameters of the model. Besides, in the above privacy-preserving training schemes, the data owners completely lose control of their data when outsourcing the data to the untrusted parties. The data ownership has not been well guaranteed, which is also a potential threat of data confidentiality. Recently, Shen et al. [12] proposed a privacy-preserving SVM training scheme over blockchain-based encrypted IoT data. However, they adopted the computing-intensive Paillier homomorphic encryption technique, which causes huge computation overheads.

In this paper, we propose a new privacy-preserving nonlinear SVM classifier training scheme based on blockchain. Specifically, we establish a blockchain-based data sharing and computation outsourcing mechanism which allows participants to collaboratively train the model while protecting privacy. We adopt the additive secret sharing technique based on two-party computation to design the computation protocols, which can achieve secure SVM training with minimal computation overheads.

The main contribution of this paper can be summarized as follows:

- 1) To train a high-quality nonlinear SVM classifier while protecting privacy, we propose a privacy-preserving training scheme based on blockchain. We utilize the blockchain technique to design a decentralized scheme for



collaborative training while ensuring the invariance and ownership of training data.

- 2) We adopt the additive secret sharing techniques and design a series of arithmetic primitives to realize efficient collaborative training while protecting privacy of both the data and the model.
- 3) The through analysis of security strength and the experiments over real-world datasets show that our scheme is secure and achieves high training efficiency and practical accuracy.

The rest of this paper is organized as follows: Sect. 2 introduces the system model and threat model of our scheme. In Sect. 3, we present the details of our privacy-preserving training scheme. Security analysis and performance evaluation are presented in Sect. 4 and Sect. 5, respectively. We conclude the paper in Sect. 6.

## 2 Problem Statement

### 2.1 System Model

In this paper, we focus on designing a scheme for privacy-preserving and efficient nonlinear SVM classifier training. There are three entities in our framework: the blockchain, the data providers, and the servers, as shown in Fig. 1. The role of each entities is described as follows:

- Blockchain: The blockchain in our scheme serves as a distributed and immutable ledger. Each block of the blockchain stores a group of transactions of the training requests, the delivery of training data, and so on.
- Data providers: The data providers take charge of encrypting the original datasets before sending them to the servers and generating random values in the secure computation process.
- Servers: The servers are two computing parties which are selected from the parties in the decentralized network. They are incented to conduct the training tasks, like the miners in BitCoin.

In the blockchain-based decentralized system, the servers should first register on the blockchain and join the network. When the data providers want to conduct collaborative training, they should first generate a transaction that contains the requests and pays. After receiving the transaction, the blockchain selects two servers as computing parties and sends the addresses and the public encryption keys of the servers to the data providers. Then, each provider encrypts the training data with the respective public keys and sends the encrypted data to the servers. On receiving the encrypted data from all the participants, the servers perform the secure computation protocols and conduct the SVM classifier training. After finishing the training process, the servers send the results back to the data providers.

### 2.2 Threat Model

In this paper, we assume that the servers are non-colluding and there are secure channels among the nodes in the decentralized network. We consider the servers to be *honest but curious*. It means that the servers would execute the designed task honestly, but they are curious to infer sensitive information from the encrypted data and the interactive messages. The privacy threats mainly come from two aspects: **1) the encrypted original datasets**. The original datasets may contain lots of sensitive information about the data providers. The adversaries can still infer valuable information about the local data through the training process; **2) the training results**. The results of training, i.e., the SVM model, can be used for some commercial benefit. Thus, the results could be embezzled by the computing parties or adversaries.

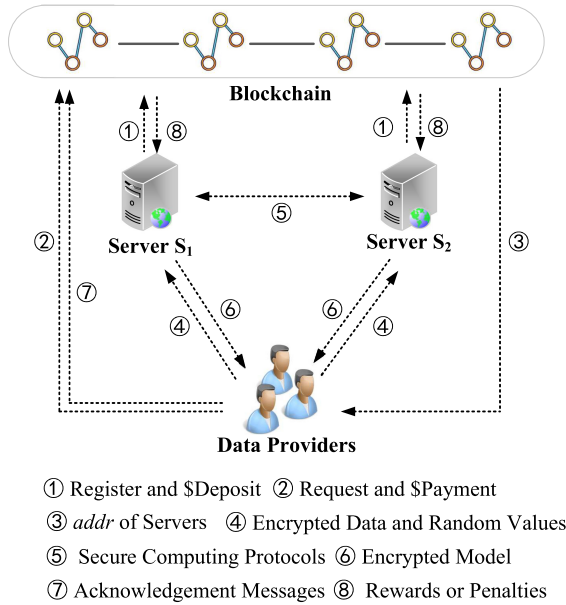


Fig. 1. System model of proposed scheme.

### 3 Privacy-Preserving SVM Classifier Training

In this section, we mainly describe the privacy-preserving SVM classifier training scheme. We first present the secure computation protocols which are based on additive secret sharing. Then we present the design of our proposed scheme. Given a value  $a$ , it will be randomly split into two shares, which are outsourced to two servers  $S_1$  and  $S_2$  respectively. Here, we denote  $\langle a \rangle_1$  and  $\langle a \rangle_2$  as the two shares stored on  $S_1$  and  $S_2$ , respectively.

### 3.1 Secure Addition/Subtraction Protocol

Given two values  $a$  and  $b$ , the addition/subtraction protocol is to jointly compute  $a \pm b$ . It is obvious that the computation can be executed by  $\mathcal{S}_1$  and  $\mathcal{S}_2$  independently since  $(\langle a \rangle_1 \pm \langle b \rangle_1) + (\langle a \rangle_2 \pm \langle b \rangle_2) = (\langle a \rangle_1 + \langle a \rangle_2) \pm (\langle b \rangle_1 + \langle b \rangle_2) = a \pm b$ . Note that there is no interaction between the two servers during the computation.

### 3.2 Secure Multiplication Protocol

The multiplication protocol is to calculate the product of two given values  $a$  and  $b$ . We adopt the *Beaver's pre-computed multiplication triplets* [13] technique to realize multiplication protocol. The steps of our secure multiplication protocol  $\text{SecMul}(\cdot)$  are given as follows:

To obtain  $c = a \times b$ , the algorithm utilize a pre-generated triplet  $(u, v, w)$ , where  $u$  and  $v$  are randomly generated and  $w = u \times v$ . The shares of  $u, v, w$  is  $u_i, v_i, w_i$  ( $i = 1, 2$ ), which are stored in  $\mathcal{S}_i$  respectively. The servers  $\mathcal{S}_i$  then calculate  $\langle e \rangle_i = \langle a \rangle_i - \langle u \rangle_i$  and  $\langle f \rangle_i = \langle b \rangle_i - \langle v \rangle_i$  locally. After that, they send  $\langle e \rangle_i$  and  $\langle f \rangle_i$  to each other and reconstruct  $e$  and  $f$ . Finally,  $\mathcal{S}_i$  compute and output the shared results as  $\langle c \rangle_i = f \cdot \langle a \rangle_i + e \cdot \langle b \rangle_i + \langle w \rangle_i + (i - 1) \cdot e \cdot f$ .

Thus, the product  $c$  can be reconstructed simply by adding the respective results of  $\mathcal{S}_1$  and  $\mathcal{S}_2$  as  $c = \langle c \rangle_1 + \langle c \rangle_2$ .

### 3.3 Secure Comparison Protocol

Given a value  $a$  and  $b$ , the secure comparison protocol  $\text{SecComp}(\cdot)$  is used to judge whether  $a < b$ . Specifically, the function outputs 1 if and only if  $a < b$ ; and outputs 0 otherwise. We adopt the bit-decomposition method in [14] and follow the comparison protocol proposed by Huang et al. [15] which is based on additively secret sharing.

We first transform the real-number shares into integers. Specifically, we multiply the numbers by  $10^p$  and truncate the remain decimal parts. Then we utilize the two's complement representation, where the most significant bit (MSB) of a number indicates whether it is a positive or negative. For a  $l$ -bit signed number  $c$ , denote its binary complement form can be denoted as  $c^{(l-1)}c^{(l-2)} \dots c^{(0)}$ . Correspondingly,  $c$  can be reconstructed as:

$$c = -c^{l-1} \cdot 2^{l-1} + \sum_{j=0}^{l-2} c^{(j)} \cdot 2^j. \quad (1)$$

Suppose that the  $l$ -bit shares of  $c$  are  $c_1$  and  $c_2$ , the protocol performs bitwise operations over  $c_1$  and  $c_2$  to compute the sign of  $c$ . Thus, the protocol can compare  $a$  and  $b$  by computing the sign of  $a - b$ .

### 3.4 The Proposed Scheme

In this subsection, we present the framework of our privacy-preserving nonlinear SVM classifier training scheme. The details are as follows:

**Algorithm 1.** Secure Gaussian kernel function

---

**Input:**  $\mathcal{S}_1$ : the shared vectors  $\langle \vec{x}_a \rangle_1, \langle \vec{x}_b \rangle_1, \sigma$ ;  
 $\mathcal{S}_2$ : the shared vectors  $\langle \vec{x}_a \rangle_2, \langle \vec{x}_b \rangle_2, \sigma$ .  
**Output:**  $\mathcal{S}_1$ : the shared Gaussian kernel result  $\langle r \rangle_1$ ;  
 $\mathcal{S}_2$ : the shared Gaussian kernel result  $\langle r \rangle_2$ .

- 1:  $\mathcal{S}_i$  initialize  $\langle s \rangle_i = 0$ .
- 2: **for**  $k$  from 1 to  $\text{len}(\langle \vec{x}_a \rangle_i)$  **do**
- 3:    $\mathcal{S}_i$  locally compute  $\langle z \rangle_i \leftarrow \langle x_a[k] \rangle_i - \langle x_b[k] \rangle_i$ .
- 4:    $\mathcal{S}_i$  compute  $\langle g \rangle_i \leftarrow \text{SecMul}(\langle z \rangle_i, \langle z \rangle_i)$ .
- 5:    $\mathcal{S}_i$  locally compute  $\langle s \rangle_i \leftarrow \langle s \rangle_i + \langle g \rangle_i$ .
- 6: **end for**
- 7:  $\mathcal{S}_i$  locally compute  $\langle f \rangle_i \leftarrow -\frac{1}{2\sigma^2} \cdot \langle s \rangle_i$ .
- 8:  $\mathcal{S}_i$  locally compute  $e^{\langle f \rangle_i}$  and randomly split  $e^{\langle f \rangle_i}$  into  $\langle e^{\langle f \rangle_i} \rangle_1$  and  $\langle e^{\langle f \rangle_i} \rangle_2$ .
- 9:  $\mathcal{S}_i$  send  $\langle e^{\langle f \rangle_i} \rangle_{i-1}$  to  $\mathcal{S}_{i-1}$ .
- 10:  $\mathcal{S}_i$  compute  $\langle r \rangle_i \leftarrow \text{SecMul}(\langle e^{\langle f \rangle_1} \rangle_i, \langle e^{\langle f \rangle_2} \rangle_i)$ .

---

**System Initialization.** Supposed that there are  $n$  data providers  $DP_j (1 \leq j \leq n)$  who own some training data respectively and want to collaboratively train an SVM classifier with a kernel function. The data providers first reach a consensus on the training protocols, parameters, and payments. Then, they send the transactions of request to the blockchain and thereafter receive the addresses and public keys of the computing servers.

After that, the data providers first encrypt the training data by randomly splitting each element into two shares. Then they encrypt the shares with the corresponding public key  $pk_i$  of the two computing servers and obtain the encrypted training datasets  $\langle D_j \rangle_i$ . Finally, the data providers send the encrypted datasets to the two servers respectively and then publish the transactions on the blockchain.

**Training.** After receiving all the encrypted datasets from the data providers, the servers decrypt the shares by utilizing their corresponding private key  $sk_i$  and perform the training protocol. In our SVM model, we choose the Gaussian kernel function to achieve nonlinear separation. The function can be interactively calculated by the two servers. The steps of calculating Gaussian kernel is shown in Algorithm 1.

To train the SVM classifier, we adopt the gradient descent (GD) as the optimization method, which is utilized in [12]. Compared with another optimization algorithm that is also frequently-used in plaintext tasks, i.e., the sequential minimal optimization (SMO), GD contains less complex computation. Thus it is regarded to be more suitable for the training in the encrypted domain. By introducing a hinge loss, the optimization problem of the SVM is converted to:

**Algorithm 2.** Secure nonlinear SVM classifier training

**Input:**  $\mathcal{S}_i$ : the split dataset  $\langle D \rangle_i = \{(\langle \vec{x}_1 \rangle_i, \langle y_1 \rangle_i), (\langle \vec{x}_2 \rangle_i, \langle y_2 \rangle_i), \dots, (\langle \vec{x}_m \rangle_i, \langle y_m \rangle_i)\}$ , learning rate  $\lambda$ , max iterations  $T$ , precision  $\varepsilon$ .

**Output:**  $\mathcal{S}_i$ :  $\langle \alpha^* \rangle_i, \langle b^* \rangle_i$ .

```

1:  $\mathcal{S}_i$  initialize  $\langle \alpha^1 \rangle_i, \langle b^1 \rangle_i, \langle loss \rangle_i$ .
2: for  $p$  from 1 to  $m$  do
3:   for  $q$  from 1 to  $m$  do
4:      $\mathcal{S}_i$  compute  $\langle K[p][q] \rangle_i \leftarrow \text{SecKer}(\langle \vec{x}_p \rangle_i, \langle \vec{x}_q \rangle_i)$ .
5:   end for
6: end for
7: while  $loss > \varepsilon$  or  $t < T$  do
8:    $\mathcal{S}_i$  initialize  $\Delta_\alpha, \Delta_b$ .
9:    $\mathcal{S}_i$  compute  $\langle loss \rangle_i \leftarrow \text{SecMul}(\text{SecMul}(\langle \alpha^t \rangle_i, \langle K \rangle_i), \langle \alpha^t \rangle_i^T)$ .
10:  for  $p$  from 1 to  $m$  do
11:    for  $q$  from 1 to  $m$  do
12:       $\mathcal{S}_i$  compute  $\langle g^p \rangle_i \leftarrow \text{SecMul}(\langle \alpha^t [q] \rangle_i, \langle K[p][q] \rangle_i)$ .
13:       $\mathcal{S}_i$  locally compute  $\langle s^p \rangle_i \leftarrow \langle s^p \rangle_i + \langle g^p \rangle_i$ .
14:    end for
15:     $\mathcal{S}_i$  locally compute  $\langle f^p \rangle_i \leftarrow \langle s^p \rangle_i + \langle b^p \rangle_i$ .
16:     $\mathcal{S}_i$  compute  $\langle f^p \rangle_i \leftarrow \text{SecMul}(\langle y_p \rangle_i, \langle f^p \rangle_i)$ .
17:     $\mathcal{S}_i$  compute  $\text{SecComp}(\langle f^p \rangle_i, 1)$ .
18:    if  $\langle f^p \rangle_i < 1$  then
19:       $\mathcal{S}_i$  compute  $\langle \Delta_\alpha \rangle_i \leftarrow \langle \Delta_\alpha \rangle_i - C \cdot \text{SecMul}(\langle y_p \rangle_i \cdot K[p])$ .
20:       $\mathcal{S}_i$  compute  $\langle \Delta_b \rangle_i \leftarrow \langle \Delta_b \rangle_i - C \cdot \langle y_p \rangle_i$ .
21:    end if
22:  end for
23:   $\mathcal{S}_i$  update  $\langle \alpha^t \rangle_i \leftarrow \langle \alpha^t \rangle_i - \lambda \cdot \langle \Delta_\alpha \rangle_i$ .
24:   $\mathcal{S}_i$  update  $\langle b^t \rangle_i \leftarrow \langle b^t \rangle_i - \lambda \cdot \langle \Delta_b \rangle_i$ .
25:   $\mathcal{S}_i$  compute  $\text{SecComp}(\langle loss \rangle_i, \varepsilon)$ .
26:   $t = t + 1$ .
27: end while
28:  $\mathcal{S}_i$  return  $\langle \alpha^* \rangle_i, \langle b^* \rangle_i$ .

```

$$\min \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^m \max(0, 1 - y_i (\sum_{j=1}^m \alpha_j K(\vec{x}_i, \vec{x}_j) + b)). \quad (2)$$

The protocol firsts executes  $\text{SecComp}(\cdot)$  to compare  $y_i (\sum_{i=1}^m \alpha_i K(\vec{x}_i, \vec{x}) + b)$  and 1. If  $y_i (\sum_{i=1}^m \alpha_i K(\vec{x}_i, \vec{x}) + b) < 1$ , the servers update  $\alpha$  and  $b$  by calculating the derivatives of the margin and the hinge loss. The steps of privacy-preserving training are shown in Algorithm 2. The dataset for training and the SVM model is well protected during the training process. The servers and other adversaries cannot infer any information except the respective shares obtained in each step.

## 4 Security Analysis

In this section, we present the security strength of our proposed scheme under the honest but curious model. We analyze the security of our scheme based on the universal composability (UC) framework [16], which is regarded to guarantee strong security properties. Due to the secret sharing based protocols, the addition and subtraction operations which are computed locally on the servers can be easily simulated. We prove the security of other computing protocols in our scheme.

**Theorem 1.** The protocol  $\text{SecMul}(\cdot)$  is secure under the *honest but curious* model.

*Proof.* The view of  $\mathcal{S}_1$  is  $\text{view}_1 = (a_1, b_1, u_1, v_1, w_1, e_2, f_2)$ . It is obvious that  $a_1, b_1$  are randomly split from  $a$  and  $b$  and  $u_1, v_1, w_1$  are uniformly random values.  $e_2$  and  $f_2$  are also random values because they are generated as  $e_2 = a_2 - u_2$ ,  $f_2 = b_2 - v_2$ . The output of  $\mathcal{S}_1$  is  $\text{view}_1 = f \cdot a_1 + e \cdot b_1 + w_1$ , which is also uniformly random. Note that both the input and output of  $\mathcal{S}_1$  are random values, so they can be perfectly simulated by  $\mathcal{S}$ . The view of adversary  $\mathcal{A}$  and its real view are computationally indistinguishable. Similarly, the input and output of  $\mathcal{S}_2$  can also be perfectly simulated.

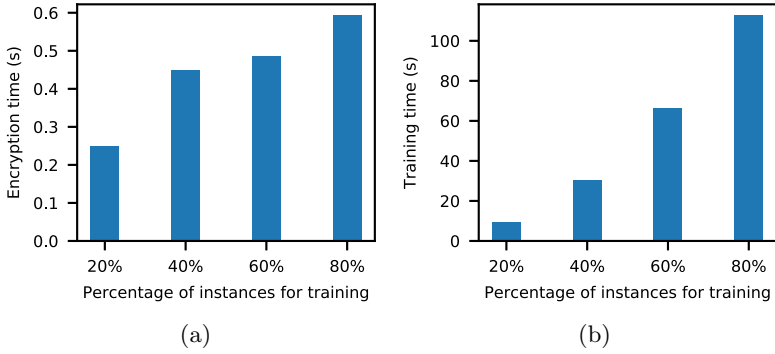
**Theorem 2.** The protocol  $\text{SecComp}(\cdot)$  is secure under the *honest but curious* model.

*Proof.* For the comparison protocol  $\text{SecComp}(\cdot)$ , the  $\text{view}_1$  and  $\text{view}_2$  of  $\mathcal{S}_1$  and  $\mathcal{S}_2$  are  $\text{view}_1 = (a_1, u_1, v_2)$ ,  $\text{view}_2 = (a_2, u_2, v_1, v_2^{(l-1)}, v_2^{(l-2)} \dots v_2^{(0)})$ . The values are random and simulatable. The bitwise addition can be deployed by secure addition and secure multiplication, which has been proved to be simulatable. Therefore, it can be proved that the comparison protocol can be simulated by a simulator  $\mathcal{S}$ .

The privacy-preserving training protocol is composed of the above computing protocols, which are proved to be secure. Thus, our privacy-preserving nonlinear SVM classifier training scheme is secure under the honest but curious model.

## 5 Performance Evaluation

In this section, we evaluate the performance of our scheme by conducting experiments over a real-word dataset. We use a real-world dataset of the UCI machine learning repository, i.e, Breast Cancer Wisconsin Database (BCWD) [17]. BCWD contains 699 instances and each instance contains nine attributes. The instances in the dataset are labeled as benign or malignant. We implement the experiments on a PC with a 32-core Intel i7 CPU @ 1.80 GHz and 16 GB RAM. The algorithms are programmed with Python 2.7. Specifically, we investigate the performance through the accuracy and the efficiency.



**Fig. 2.** Comparison of the computation time with different datasets.

**Table 1.** Comparison of the accuracy.

	Our scheme	SVM	Shen’s scheme
Precision	93.3%	93.3%	90.35%
Recall	1	1	96.19%

**Table 2.** Comparison of the efficiency.

	Shen’s scheme	Our scheme
Server side	953 s	146 s
Data provider side	2233 s	1.7 s

### 5.1 Accuracy

The precision rate and the recall rate are two key parameters to evaluate the accuracy of a classifier. We calculate the two parameters and compare our scheme with the SVM classifier implemented over plaintext. We also compare our scheme with the privacy-preserving training scheme proposed by Shen et al. The results are shown in Table 1. We can see that our scheme can achieve the same accuracy with the SVM classifier over plaintext and higher accuracy than Shen et al.’s scheme. It demonstrates that the cryptographic methods in our scheme do not influence the classification functionality. Our scheme can maintain high accuracy while protecting privacy.

### 5.2 Efficiency

In this subsection, we evaluate the efficiency of our scheme. Specifically, we investigate the time consumption both on the data providers and the servers by utilizing cross-validation. We compare the efficiency of our scheme with that of Shen et al.’s scheme. It is shown in Table 2 that our scheme achieves much

better efficiency performance. It is because that our scheme does not involve any computationally-expensive cryptography techniques. We can see that the time consumption on the data provider side in our scheme is just 1.4s, while in Shen et al.'s scheme it takes the data provider more than 2000s. The time of computation on the server side is also much less than Shen et al.'s scheme.

We also evaluate the time consumption with different percentage of instances for training and testing, as shown in Fig. 2. Specifically, we randomly take several percentages of instances for training and the others for testing. We can see that the time consumption is positively correlated to the percentage of instances for training because that more instances for training means a larger number of computation both on the data provider side and the server side. Overall, the efficiency of our scheme is acceptable for practical use. The experiment results show that our scheme is much more efficient than Shen et al.'s scheme and achieves better overall performance for practical utilization.

## 6 Conclusion

In this paper, we proposed a new privacy-preserving nonlinear SVM classifier training scheme for multiple data providers. We utilize the blockchain technique to design a decentralized framework for data sharing and training while ensuring the invariance of datasets. We adopt the additive secret sharing based on secure two-party computation and design a suite of secure computing protocols to conduct the training process with no information leakage. Our training scheme is proved to be secure through comprehensive analysis. Experiments over real datasets demonstrate that our scheme can achieve high accuracy and efficiency for practical applications.

## References

1. Hua, J., et al.: Cinema: efficient and privacy-preserving online medical primary diagnosis with skyline query. *IEEE Internet Things J.* **6**(2), 1450–1461 (2018)
2. Ma, Z., Liu, Y., Liu, X., Ma, J., Ren, K.: Lightweight privacy-preserving ensemble classification for face recognition. *IEEE Internet Things J.* **6**(3), 5778–5790 (2019). Kindly provide the volume number and page range for Refs. [5 and 15], if applicable
3. Ghazanfar, M., Prugel-Bennett, A.: An improved switching hybrid recommender system using Naive Bayes classifier and collaborative filtering (2010)
4. Bennett, K.P., Demiriz, A.: Semi-supervised support vector machines. In: *Advances in Neural Information processing systems*, pp. 368–374 (1999)
5. Li, J., et al.: Searchable symmetric encryption with forward search privacy. *IEEE Trans. Dependable Secur. Comput.* (2019)
6. Liu, Z., Li, B., Huang, Y., Li, J., Xiang, Y., Pedrycz, W.: NewMCOS: towards a practical multi-cloud oblivious storage scheme. *IEEE Trans. Knowl. Data Eng.* **32**, 714–727 (2019)
7. Accountability Act: Health insurance portability and accountability act of 1996. Public Law **104**, 191 (1996)



8. Bost, R., Popa, R.A., Tu, S., Goldwasser, S.: Machine learning classification over encrypted data. In: NDSS, vol. 4324, p. 4325 (2015)
9. Zhu, H., Liu, X., Rongxing, L., Li, H.: Efficient and privacy-preserving online medical prediagnosis framework using nonlinear SVM. *IEEE J. Biomed. Health Inform.* **21**(3), 838–850 (2016)
10. Chaudhuri, K., Monteleoni, C., Sarwate, A.D.: Differentially private empirical risk minimization. *J. Mach. Learn. Res.* **12**, 1069–1109 (2011)
11. Abadi, M., et al.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318 (2016)
12. Shen, M., Tang, X., Zhu, L., Xiaojiang, D., Guizani, M.: Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities. *IEEE Internet Things J.* **6**(5), 7702–7712 (2019)
13. Beaver, D.: Efficient multiparty protocols using circuit randomization. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 420–432. Springer, Heidelberg (1992). [https://doi.org/10.1007/3-540-46766-1\\_34](https://doi.org/10.1007/3-540-46766-1_34)
14. Damgård, I., Fitzgi, M., Kiltz, E., Nielsen, J.B., Toft, T.: Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 285–304. Springer, Heidelberg (2006). [https://doi.org/10.1007/11681878\\_15](https://doi.org/10.1007/11681878_15)
15. Huang, K., Liu, X., Fu, S., Guo, D., Xu, M.: A lightweight privacy-preserving CNN feature extraction framework for mobile sensing. *IEEE Trans. Dependable Secure Comput.* (2019)
16. Canetti, R., Cohen, A., Lindell, Y.: A simpler variant of universally composable security for standard multiparty computation. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 3–22. Springer, Heidelberg (2015). [https://doi.org/10.1007/978-3-662-48000-7\\_1](https://doi.org/10.1007/978-3-662-48000-7_1)
17. Asuncion, A., Newman, D.: UCI machine learning repository (2007)



# A Novel Defending Scheme for Graph-Based Classification Against Graph Structure Manipulating Attack

Quanyu Zhao<sup>(✉)</sup>, Fengqian Zhang, and Yuan Zhang

State Key Laboratory for Novel Software Technology,  
Nanjing University, Nanjing 210023, China  
quanyu.zhao1105@gmail.com

**Abstract.** Graph-based classification methods are widely used in social network. Wang et al. [1] proposed an attack for the collective classification method by manipulating the structure of the graph. In this paper, we propose a novel defense scheme for this attack by repairing the graph structure via deleting the edge for the key nodes. Our experiments demonstrate that our scheme is very effective in resisting such attacks and enables the classification algorithm to achieve pre-attack accuracy and precision.

**Keywords:** Graphbased classification algorithm · Graph structure · Social network · Effective

## 1 Introduction

Graph-based classification algorithms are widely used in various problems, for instance detection of malware [2], malicious users in social software [3–5], and false comments [6, 7], attribute inference [8, 9] et al.. By training on the positive and negative nodes set, Graph classification algorithm classified the untagged nodes into positive or negative nodes. The tags of positive and negative differ considerably in different questions. Some works [10–13] demonstrate that Graph-based classification algorithms are highly effective in detecting the malicious users who may attack the negative nodes.

Currently, it exists two types of graph-based classification algorithms, including graph neural network [14–16] and collective classification [5, 6, 14]. Graph neural networks [15–18] trains on the graph structure and obtains feature vectors for each node, and classifies the nodes by the the feature vectors. Collective classification defines a priori reputation score and weight for each node in the training set, computes a priori reputation score by using the weight graph, and classifies each node according the priori reputation score. Some novel and widely using collective classification algorithms exist at present, including Randow walk (RW) algorithms [5, 19], circular confidence propagation algorithms (LBP) algorithms [1, 2], linear circular confidence propagation algorithm (LinLBP) [1], joint weight learning and propagation (JWP) algorithm [7].

The RW algorithm calculates the posteriori reputation scores by random wandering, scores for the node iteratively, assigns the scores to the node’s neighbors by the weight. After iterations, we can categorize the nodes by scoring. The JWP algorithm obtains the weights and propagates the scores simultaneously. The posteriori reputation score can be calculated by iterating to balance. Finally, the nodes are classified by a posteriori reputation scoring. In LBP algorithm, it distributes the same weight to each side, models the graph as a pair of Markov random airports, employs the standard LBP algorithm to score each node. However, it exists a limitation that the Convergence and incompressibility for the circular graph can not be satisfied simultaneously. A modified LinLBP algorithm solves the limitation.

With the widely using of the graph classification algorithms, there are always some attacking algorithms since the attacker can evade detection easily. Some existing attacking algorithms include clustering attacking algorithm [20], neural networks attacking algorithms [21–23], and collective classification attacking algorithm [1]. Clustering attacking and collective classification attacking algorithms always modify the structure of the graph. Changing the structure of the graph means inserting a false edge between two nodes or removing an existing edge. This method will prompt the classification algorithms misclassifying the target node. Some researches [24–27] based on non-diagrammatic data in adversarial machine learning were proposed. Utilizing the graph-based adversarial machine learning, Wang et al. [1] proposed an attacking algorithm by modifying the structure of the diagram. This attacking algorithm invalidates the classification algorithm by inserting some edges or deleting the certain specific edges in the graph. Most of the defense algorithms against the generic attacks are based on non-graphic data. Those algorithms will fail to defense Wang’s attacking algorithms.

In this paper, we propose a defense algorithm against the attacking based on the graph. Our algorithm modifies the graph structure by deleting the key nodes randomly to “repair” the diagram. We delete the edges among the key nodes to resist the attack, this method also has less impact on the original classification algorithm. The main contributions of our work are listed as follows:

- In this paper, we analyze the attack which works on the Graph-based classification via manipulating the graph structure.
- We implement a replication of the attack algorithm and propose a defence model against the attack via deleting the key nodes randomly.
- We design some experiments based on multiple real-life social network data sets, the results demonstrate that our model is very effective in resisting the attack and has small impact on the original classification algorithm simultaneously.

The rest of the paper is organized as follows. Some background and problem setup are presented in Sect. 2. Section 3 analyze the attack. We put forward the our defense scheme in the Sect. 4. The summarize and the future work were introduced in Sect. 5.

## 2 Background and Problem Setup

### 2.1 Notations

Throughout our works, we utilize  $G = (V, E)^1$  to denote an undirected graph,  $L$  denotes the training dataset, including the positive set  $L_P$  and negative set  $L_N$ . Other parameters include:

- $V$  and  $E$  denotes the nodes set and the edges set,  $u \in V$  means the node  $u$  belonging to  $V$ .  $(u, v) \in E$  denotes an edge between  $u$  and  $v$ .
- $q_u$  denotes the prior reputation score for a node  $u$ . If  $u \in L_P$ , it denotes  $q_u = \theta$ , and  $u \in L_N$  denotes  $q_u = -\theta$ , otherwise,  $q_u = 0$ .
- $\theta$  denotes a set  $(0, 1]$ ,  $\omega$  is the weight parameter for the nodes which belongs to a set  $(0, 0.5]$ .
- $A$  and  $W$  denotes an adjacency matrix of the graph and a  $|V| \times |V|$  matrix.
- $p$  and  $q$  denotes the column vector of prior reputation scores and posterior reputation scores of all nodes.

### 2.2 Reviews of the Linearized Loopy Belief Propagation

We retrospect the Linearized Loopy Belief Propagation (LBP). In LBP, the posterior reputation scores are computed by Eq. 1.

$$p = q + A \odot W_P \quad (1)$$

where the  $W_P$  denotes the weight matrix for the node  $p$ ,  $\odot$  denotes the multiplication of two matrixes.  $A \odot W_P$  denotes the weight matrix of the graph. We calculate the iteration of the posterior reputation scores by Eq. 2.

$$p^t = q + A \odot W_P^{t-1} \quad (2)$$

where  $t$  denotes the  $t$ th iteration. Iff the posterior reputation scores  $p_u < 0$  and it is converge, the node  $u$  is a negative node.

The LBP system consists of three components, including the parameters, training set, and the graph. The parameters contain the weight parameter  $W$  and  $\theta$ . We combine the components into an array as {parameters, training set, Graph}, where the adversary may know the parameters or not, which are denoted as Yes or No. The training set can be trained or not for the adversary, we denote it as Yes or No. The adversary knows the graph complete or partial. The different arrays demonstrate the differ ability of the adversary. For example, {Yes, Yes, Complete} denotes that the adversary knows the parameters and has ability to train on the set, they know the structure of the graph in the social network. This situation maximizes the efficiency of the attack, and it is the most threatening situation. The array {No, Yes, Complete} means that the attacker obtains no information about the parameters, the attacker should choose a random value within the parameter range to replace these two parameters. The array {Yes, No, Complete} shows the attacker does not equipped with ability of

training set, the attacker should take a part of the original graph as a training set. The attack knows his target nodes will be regarded as the positive nodes, then, the attacker always utilizes these nodes as positive nodes, and attacks some other nodes that may be negative.  $\{Yes, Yes, Partial\}$  means the attacker knows at least one of his target nodes and the relationship between the edges and these nodes. We assume that the attacker knows a part of negative nodes, then the attacker collects more negative nodes by the crawler in this social network.  $\{No, No, Partial\}$  means the attacker executes the attack after using the above three alternatives in turn.

Assumption that the positive and negative nodes denote the malicious and general users. The adversary has some target nodes which will be classified as positive nodes generally. The adversary will attack the LinLBP classification algorithm and misclassify these target nodes as negative nodes. We assume  $FNR$  denotes the rate of the misclassified nodes, it is calculated by Eq. 3.

$$FNR = \frac{N_f}{N_t} \times 100\% \quad (3)$$

Where  $N_f$  and  $N_t$  denotes the number of the misclassification nodes and the target nodes.  $FNR$  is an evaluation criterion for the efficiency of the attacking.

The attack algorithm proposed by Wang et al. [1] changes the structure of the graph by deleting or inserting an edge. The cost of deleting or insertion an edge in differ nodes are different. The cost of inserting an edge between the positive nodes and negative nodes is much higher than the cost of inserting the edges among the positive nodes. The easiest and most effective way to attack the LinLBP is to sever the relationship among the negative nodes and build many relationships with the general nodes. In the real social network, we will limit the number of relationships for a single node, and limit a maximum of  $K$  sides inserted or deleted for the nodes by the attacker.

Wang et al. [1] utilizes a binary variable to represent the situation of the edge, when the situation of the edge changes,  $B_{uv}$  will be recorded as 1, otherwise  $B_{uv} = 0$ . Then, they turned the attack algorithm into an optimization problem. The objective function of this optimization problem contains three restrictions, 1) calculate the cost; 2)  $FNR = 1$ ; 3) the maximum number of inserted or deleted edges per nodes is  $K$ . The goals for the attacker are to optimize the problem listing as follows:

$$\begin{aligned} \min F(B) = & \sum_{u \in \tau, v \in V - \tau} B_{uv} C_{uv} + \sum_{u, v \in \tau, u < v} B_{uv} C_{uv} + \lambda \sum_{u \in \tau} p_u, \\ \text{s.t.} \quad & FNR = 1 \\ & B_{uv} \in [0, 1], u \in \tau, v \in V \\ & \sum_v \overline{B_{uv}} \leq K, u \in V \end{aligned} \quad (4)$$

Where  $C_{uv}$  denotes the cost of deleting or inserting an edge between node  $u$  and  $v$ .  $\tau$  represents the target set and  $\lambda$  is the Lagrange factor. In order to simplify

calculation, we shrink a binary variable  $B_{uv}$  to a continuous variable  $\overline{B_{uv}} \in [0, 1]$ . Then, we calculate the posterior reputation scores by Eq. 5.

$$p = q + |A - \overline{B}| \odot W_P \quad (5)$$

Where  $|A - \overline{B}|$  is the adjacency matrix after the attack. Utilizing the gradient descent method is an effective method to solve the optimization problem. We reduce the computational difficulty by replacing the final convergent posteriori reputation score with the posteriori reputation score in the iterative process.

### 3 Analysis of the Wang's Attack

Now, we start to re-produce the attack algorithm. We divide the points into positive nodes and negative nodes randomly and uniformly. The LinLBP algorithm will help us to complete this work. However, the attacker will choose the positive nodes as the target nodes in two ways, random selecting or connected component selecting, denoted as Rand and CC. The difference between them is the efficiency of the attacking. Rand attacking is randomness and CC attacking is certain. In CC attacking, if a node is misclassified as a negative node, all of the connective component are misclassified as negative nodes easily, this method will increase the value of the FNR.

After selecting the target nodes, the attacker defines the cost for all edges, no matter the edge exists in the graph or not. There are three methods for defining the cost. The first method is to define the same cost for all edges, it is simple but not meet the definition of the cost in real social networks. The second one defines the edges with different cost meeting the requirements of the social network, but we quit it for the complexity. The third one divides the edges into some regions, and defines the same regions with the same cost.

Thereafter, the attacker calculates the attack matrix, modifies the original graph by using the two-valued attack matrix to obtain a new graph. Running the LinLBP algorithm on the new graph, we classify the nodes by using the posteriori reputation score. We pay more attention to the target nodes, whether these nodes were misclassified and calculate the value of the FNR.

We employ two social network graphs as the test sets, including the data set collection for Facebook (4039 nodes, 88234 edges) and Enron (33696 nodes, 180811 edges). We obtain the relevant graphs from the Stanford Large Network. Due to the complex of the Enron dataset, we choose them as the testing dataset, and use the Facebook dataset for secondary validation.

After pricing the cost of the edges in Enron dataset by using the third method, we select 100 target nodes as the attack nodes using the method of Rang and CC, respectively. Assume  $\theta = 0.5$ ,  $\omega = 0.01$ ,  $\lambda = 1000$ , and  $K = 20$ . We select 100 positive nodes and 100 negative nodes as our training set and operate some experiments. FNR is equal to 0.99 and 1 in the Rand attacking and CC attacking on the condition of other parameters are constant. Moreover, the value of FNR increase with the increase of the  $K$  value. Next, we analyze the different arrays of the attackers.

{No, Yes, Complete}: The attackers choose  $\theta \in (0, 1)$  randomly. Our experiments demonstrate that FNR remains constant, it means  $\theta$  does not affect the efficiency of the attacking. The attackers choose  $\omega \in (0, 0.5]$  randomly, the result of experiments show that the value of FNR gets closer to 0 with the condition of the  $\omega$  gets closer to 0.5. The attackers always increase the value of FNR by keeping test and obtaining a suitable  $\omega$ . Then, constructing the defense scheme in this respect is impossibility.

{Yes, No, Complete}: Many experiments show that the value of FNR will increase with the increasing of the training test, and the attackers always gain a suitable training test making his attack more efficient. Then, we can not resist the attack in this regard.

{Yes, Yes, Partial}: When the attackers does not know the full graph, we control the attackers obtain the scale of the graph, expands the graph by using the breadth of the preferred iterative method. The results of tests show that the value of FNR will become larger with the increase of the proportion of the obtained graph. When the proportion reaches about 25%, the value of FNR reaches the same effect as the full graph.

Comparing the changes before and after attacking, we find two specific phenomena for the attacker's target nodes. First, the edges, which are among the target nodes or among the target nodes and some high posteriori reputation score positive nodes, will be deleted after attacking. Second, there does not exist edges among the target nodes and the negative nodes. If the edges exist at the beginning, this nodes would be classified as negative nodes by the LinLBP algorithm with a high probability. After attacking, the target nodes prefer to establish relationships among these nodes or nodes with lower a posteriori reputation scores.

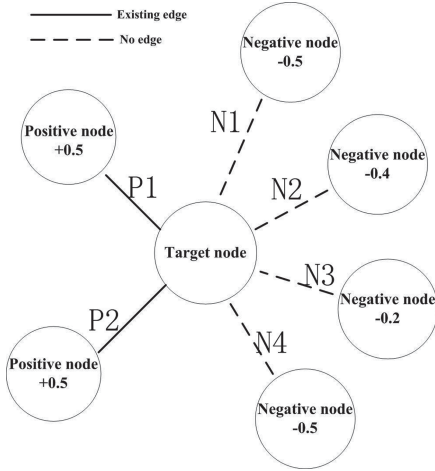
## 4 A Novel Defending Scheme for Graph-Based Classification Against Graph Structure Manipulating Attack

### 4.1 Defence Scheme Design

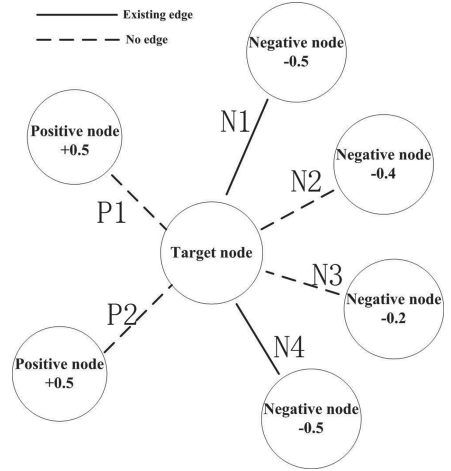
In the LinLBP classification criterion, a node is classified as a negative node if its posteriori reputation score is less than 0, otherwise it is classified as a positive node. Then, the attackers tend to disconnect from nodes with higher posteriori reputation scores and establish relationships with lower posteriori reputation scores nodes. Combining the above analysis, we propose a novel defense scenario.

As shown in Fig. 1, the edges  $P_1, P_2$  were established between the target node and two posteriori reputation score +0.5 nodes after executing the LinLBP classification algorithm. Two nodes are classified as positive nodes immediately, the target nodes connected with these nodes are classified as positive nodes. Target nodes choose to create  $N_1, N_4$  edges while cutting off  $P_1, P_2$  edges, thereby lowering its own posteriori reputation scores, as shown in Fig. 2.

Taking the cost of edges into consideration, target nodes will be tracked by calculating the cost of edges. The attackers delete the edges among the positive



**Fig. 1.** Target nodes before the attack



**Fig. 2.** Target nodes after the attack

nodes, add the edges among the low posteriori reputation score negative nodes, or build the edges with the nodes which are always treated as negative nodes. Therefore, users in the LinLBP algorithm obtain the original diagram, training sets and parameters, etc. It can “recover” the construction of the graph by deleting edges among the key nodes or adding edges to the key nodes randomly. This method changes the posteriori reputation scores for the target nodes, then, the target nodes will not be misclassified as negative nodes. The value of FNR will reduce and our scheme defends the attack.

The Enron data set includes ten thousand nodes, and each node may connect with hundred nodes. It means there are hundreds edges connecting with this nodes, adding one or several edges for this node randomly may have no effect for repairing the graph, but it has significant effect on the using of LinLBP classification algorithm. That situation determines that deleting edges randomly will be more effective in repairing the graph. We delete several or dozens of the edges for the nodes, the deleting edges may contain the new created edges. Even though, the nodes may be “silos” (no other nodes are connected to them) and the nodes can not be classified properly. It is possible to reduce the value of the FNR value and ensure the LinLBP classification algorithm by deleting an appropriate number and proper edges.

There are two methods to delete the edge randomly. One is training, we select the negative nodes as the key nodes, and delete the edge among them randomly. The other is TopN, we sort the nodes according to the posteriori reputation scores after running the LinLBP algorithm, and delete these edges among  $N$  nodes with the smaller posteriori reputation scores randomly.



## 4.2 Performance Testing for Training

We assume the number of the positive and negative nodes are 32492 and 34091, so the rate of them is 0.482 and 0.518 before the deleting edges. We choose the negative nodes as the key nodes by using the method of CC, and deleting the edges among them randomly. We utilize the new graphs to run the LinLBP classification algorithm and reclassify for the same training set and parameters. We set up 10 experimental groups, and the results are listed in Table 1, and we paint the results in Fig. 3.

**Table 1.** Training CC for the deleting edges randomly

Group	Deleting edges(n)	After the deleting		
		FNR	Positive	Negative
1	5	1	0.486	0.514
		100	32731	34662
2	10	1	0.49	0.51
		100	33010	34380
3	15	0.99	0.494	0.506
		99	33272	34114
4	20	0.92	0.499	0.501
		92	33606	33751
5	25	0.65	0.502	0.498
		65	33820	33489
6	30	0.46	0.506	0.494
		46	34097	33211
7	35	0.21	0.509	0.491
		21	34242	33008
8	40	0.04	0.51	0.49
		4	34273	32965
9	45	0.06	0.512	0.488
		6	34427	32808
10	50	0.05	0.514	0.486
		5	34545	32670

We compare our results with those before the experiments. The results demonstrate that the value of FNR will decrease with the increase of the deleting edges. This results show the efficiency of the attacking have decreased. Table 1 also shows that the proportion of positive nodes is going up and the proportion of the negative nodes is going down. The reason of this situation is that the connection edges of the negative nodes are deleted and the posteriori reputation score of those nodes increases, some nodes are classified as the positive

nodes. We discuss how to reduce the impact in classification phase hereinafter. The results indicate that the number of nodes is going down, the reason of this situation is that the “silo” nodes generated after the deleting edges can not be able to perform the LinLBP classification algorithm. We will utilize the anterior classification result to tag those nodes. Figure 3 shows that the value of the FNR will decrease linearly and reach steady case eventually with the increasing number of the deleting edges. The experiments show that our scheme has significant impact on the efficiency of the attacking, and has less impact on the LinLBP classification algorithm.

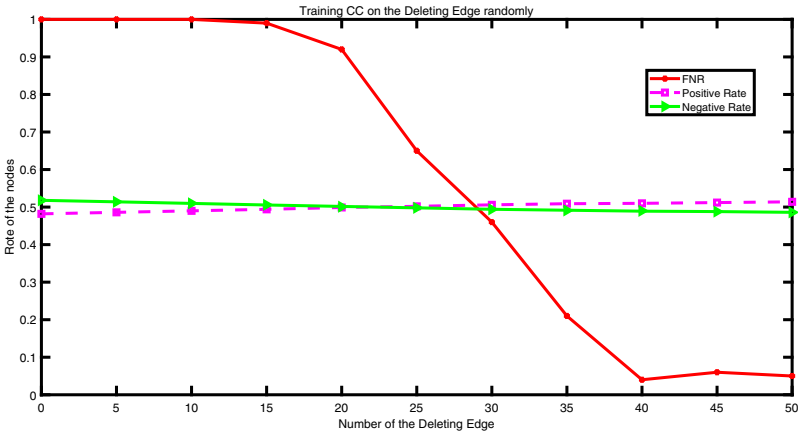


Fig. 3. Training CC for the deleting edges randomly

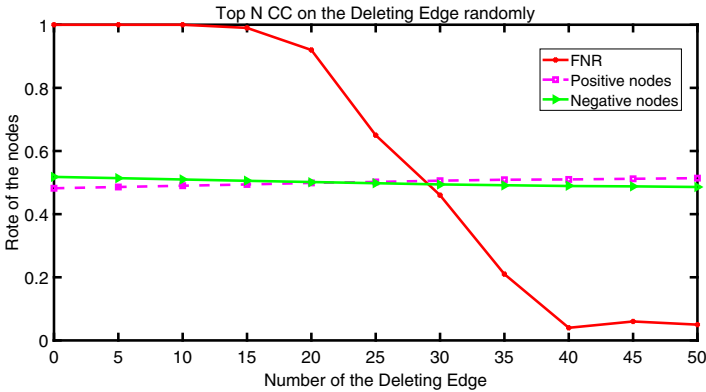
### 4.3 Performance Testing for Top N

Assume the number of the positive and negative nodes is 32482 and 34894, so the rate of them is 0.482 and 0.518. We sort the nodes according to the posteriori reputation scores after running the LinLBP algorithm, select  $N$  nodes with smaller posteriori reputation scores, delete the edges among these nodes randomly. We utilize the new graphs to run the LinLBP classification algorithm and reclassify for the same training set and parameters. Assume  $N = 300$ . We delete the edges among these 300 nodes including the target nodes, negative nodes and some small posteriori reputation scores nodes. All of these nodes are connected with the negative nodes with a close relationship in the training set. The results are listed in Table 2, and we utilize the results to plot the Fig. 4.

Comparing with the method of training, when the rate of positive nodes and the negative nodes are the same, under the condition of the same value for FNR, TopN deletes fewer edges to reach the same goals. The results of the experiments show that the value of FNR will decrease with the increase of the deleting edges. It also demonstrates the efficiency of the attacking have decreased. Table 2 also shows that the proportion of positive nodes is going up and the proportion of

**Table 2.** Top N CC for the deleting edges randomly

Group	Deleting edges(n)	After the deleting		
		FNR	Positive	Negative
1	5	1	0.491	0.509
		100	33063	34313
2	10	1	0.496	0.504
		100	33420	33937
3	15	0.99	0.499	0.501
		99	33604	33678
4	20	0.92	0.504	0.496
		92	33881	33324
5	25	0.65	0.506	0.494
		65	34009	33138
6	30	0.46	0.51	0.49
		46	34217	33878
7	35	0.21	0.51	0.49
		21	34226	32857



**Fig. 4.** Top N CC for the deleting edges randomly

the negative nodes is going down. This method will have impact on the LinLBP classification algorithm. Then, we modify the judge criteria for negative nodes to reduce the impact on the LinLBP classification algorithm.

**4.4 Performance Testing for Modified Judgment Criteria**

We assume the number of the positive and negative nodes is 34079 and 33211, so the rate of them is 0.506 and 0.494, and the value of FNR is 0.46. We utilize the positive and negative nodes in Table 1 and Table 2 to plot Fig. 5.

Figure 5 demonstrates that the negative nodes will be classified as the positive nodes with the increase of the deleting edges. The reason of this situation is that the connection edges of the negative nodes are deleted and the posteriori reputation score of those nodes increases, some nodes are classified as the positive nodes. Therefore, we raise and lower the judgment criteria, operate several experiments to analyze the effect. The results are shown in Fig. 5.

We delete the edges randomly by using the method of training, select the negative nodes by the CC. Assume the number of the positive and negative nodes are 34079 and 33211, so the rate of them are 0.506 and 0.494, the value of FNR is 0.46,  $N = 30$ . We set up 6 experimental groups with the judgment

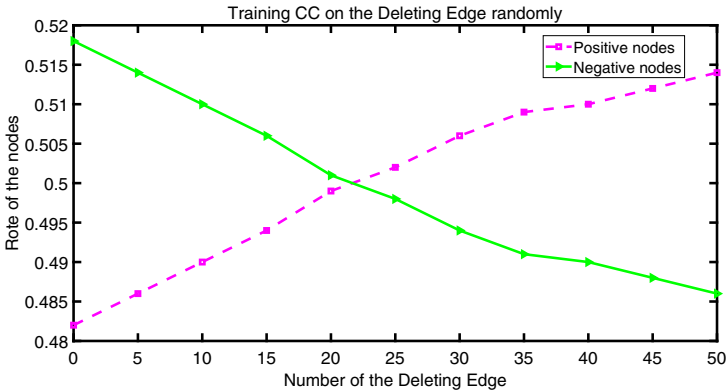


Fig. 5. Top N CC for the deleting edges randomly

Table 3. Training CC for the deleting edges randomly after the modification of standard

Group	$p_u$	After modifying the standard		
		FNR	Positive	Negative
1	0.0001	0.62	0.432	0.568
		62	29051	38199
2	0.00001	0.57	0.478	0.522
		57	32139	35111
3	0.000001	0.55	0.495	0.505
		55	33260	33990
4	-0.000001	0.21	0.52	0.48
		21	34961	32289
5	-0.00001	0.21	0.537	0.463
		21	36140	31110
6	-0.0001	0.18	0.585	0.415
		18	39342	27908

criteria  $p_u$  are 0.0001, 0.00001, 0.000001,  $-0.000001$ ,  $-0.00001$ ,  $-0.0001$ . The results of the experiments are listed in Table 3.

Table 3 shows that the value of FNR increases with the increasing of the judgment criteria, the proportion of misclassification for negative nodes will reduce. Then we tolerate the increasing of the FNR, raise the judgment criteria appropriately, the impact on the LinLBP classification algorithm will reduce. In order to obtain the lower value of FNR, we should tolerate the impact on the LinLBP classification algorithm and reduce the judgment criteria.

## 5 Conclusion and Future Work

We analyze the attacking which works on the Graph-based Classification via manipulating the graph structure, implement a replication of the attack algorithm and propose a novel defending scheme for Graph-based classification against Graph structure manipulating attacking. We design the experiments on multiple real-life social network data sets, the results demonstrate that our model can resist the attack greatly and has small impact on the original classification algorithm simultaneously. In the future work, we extend the defense to other collective classification algorithm and graph neural network classification algorithm. Meanwhile, we will improve the defense scheme for the purpose of choosing the deleting nodes high-efficiency and reducing the impact on the LinLBP classification algorithm.

**Acknowledgements.** This work was supported in part by National Key R&D Program of China (2018YFB1004301), NSFC-61872179, Fundamental Research Funds for the Central Universities (020214380052), NSFC-61425024, NSFC-61872176.

## References

1. Wang, B., Gong, N.: Attacking graph-based classification via manipulating the graph structure. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp. 2023–2040 (2019)
2. Tamersoy, A., Roundy, K., Chau, D.: Guilt by association: large scale malware detection by mining file-relation graphs. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1524–1533 (2014)
3. Wang, B., Jia, J., Zhang, L., et al.: Structure-based sybil detection in social networks via local rule-based propagation. *IEEE Trans. Netw. Sci. Eng.* **6**(3), 523–537 (2018)
4. Jia, J., Wang, B., Gong, N.: Random walk based fake account detection in online social networks. In: 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 273–284 (2017)
5. Cao, Q., Sirivianos, M., Yang, X., et al.: Aiding the detection of fake accounts in large scale social online services. Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation, pp. 197–210 (2012)

6. Akoglu, L., Chandy, R., Faloutsos, C.: Opinion fraud detection in online reviews by network effects. In: Seventh International AAAI Conference on Weblogs and Social Media (2013)
7. Wang, B., Jia, J., Gong, N.: Graph-based security and privacy analytics via collective classification with joint weight learning and propagation. arXiv preprint [arXiv:1812.01661](https://arxiv.org/abs/1812.01661) (2018)
8. Gong, N., Liu, B.: You are who you know and how you behave: attribute inference attacks via users' social friends and behaviors. In: 25th USENIX Security Symposium Security, pp. 979–995 (2016)
9. Jia, J., Wang, B., Zhang, L., et al.: AttrInfer: inferring user attributes in online social networks using Markov random fields. In: International Conference on World Wide Web (2017)
10. Zhuang, C., Ma, Q.: Dual graph convolutional networks for graph-based semi-supervised classification. In: Proceedings of the 2018 World Wide Web Conference, pp. 499–508 (2018)
11. Wang, W., Shang, Y., He, Y., et al.: BotMark: automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors. *Inf. Sci.* **511**, 284–296 (2020)
12. Anis, A., ElGamal, A., Avestimehr, A., et al.: A sampling theory perspective of graph-based semi-supervised learning. *IEEE Trans. Inf. Theory* **65**(4), 2322–2342 (2018)
13. Sun, Y., Wang, S., Tang, X., et al.: Adversarial attacks on graph neural networks via node injections: a hierarchical reinforcement learning approach. In: Proceedings of the Web Conference 2020, pp. 673–683 (2020)
14. Li, Z., Alrwais, S., Xie, Y., et al.: Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures. In: 2013 IEEE Symposium on Security and Privacy, pp. 112–126 (2013)
15. Battaglia, P., Hamrick, J., Bapst, V., et al.: Relational inductive biases, deep learning, and graph networks. arXiv preprint [arXiv:1806.01261](https://arxiv.org/abs/1806.01261) (2018)
16. Kipf, T., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907) (2016)
17. Cao, S., Lu, W., Xu, Q.: GraRep: learning graph representations with global structural information. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 891–900 (2015)
18. 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)
19. Boshmay, Y., Logothetis, D., Siganos, G., et al.: Integro: leveraging victim prediction for robust fake account detection in OSNs. In: NDSS, vol. 15, pp. 8–11 (2015)
20. Chen, Y., Nadji, Y., Kountouras, A., et al.: Practical attacks against graph-based clustering. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1125–1142 (2017)
21. Bojchevshi, A., Günnemann, S.: Adversarial attacks on node embeddings via graph poisoning. [arXiv:1809.01093](https://arxiv.org/abs/1809.01093) (2018)
22. Dai, H., Li, H., Tian, T., et al.: Adversarial attack on graph structured data. arXiv preprint [arXiv:1806.02371](https://arxiv.org/abs/1806.02371) (2018)
23. Zügner, D., Akbarnejad, A., Günnemann, S.: Adversarial attacks on neural networks for graph data. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2847–2856 (2018)

24. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint [arXiv:1206.6389](https://arxiv.org/abs/1206.6389) (2012)
25. Ganju, K., Wang, Q., Yang, W., et al.: Property inference attacks on fully connected neural networks using permutation invariant representations. In: The 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 619–633 (2018)
26. Tramèr, F., Zhang, F., Juels, A., et al.: Stealing machine learning models via prediction APIs. In: 25th USENIX Security Symposium Security, pp. 601–618 (2016)
27. Zhou, Z.: Abductive learning: towards bridging machine learning and logical reasoning. *Sci. China Inf. Sci.* **62**, 076101 (2019)



# Privacy-Preserving Graph Operations for Social Network Analysis

Peng Li<sup>1,2</sup>, Fucui Zhou<sup>1</sup>(✉), Zifeng Xu<sup>1</sup>, Yuxi Li<sup>1</sup>, and Jian Xu<sup>1</sup>

<sup>1</sup> Software College, Northeastern University, Shenyang, China  
peng\_lee@126.com, {fczhou,xuj}@mail.neu.edu.cn, dk@tnimdk.com,  
eliyuxi@gmail.com

<sup>2</sup> School of Information Engineering, Eastern Liaoning University, Dandong, China

**Abstract.** Nowadays, our daily life is surrounded by various social networks, and they play an important role for people to communicate with others. The social networks contain large amount of valuable information, that can be used for research and business purposes. As a result, social network analysis and data mining receive lots of research attentions in recent years. Graph structure is commonly used in social network analysis, since it is easy to convert the data in social networks into graph-structured data, and various graph algorithms can help to solve different computing problems. In this paper, we investigate performing graph operations in a privacy-preserving manner, which are widely used in social network analysis. We propose two protocols that allow two parties to jointly compute the intersection and union of their graphs. Our protocols utilize homomorphic encryption to prevent information leakage during the process, and we provide security proofs of the protocols in the semi-honest setting.

**Keywords:** Social network analysis · Privacy · Homomorphic encryption · Graph operation

## 1 Introduction

As the fast development of social networks, people are spending more and more time in social network platforms to communicate with old friends, make new friends and share various contents with others. As a result, social networks tend to contain very large amounts of valuable data. Therefore, social network analysis and data mining have become hot research areas in recent years.

Over the past decades, many data analysis, data mining, knowledge discovering and modeling methods for social network were proposed. Graph data structure and graph processing are commonly used in social network analysis. Social networks can be naturally converted into graph structures, and various graph algorithms can help to solve many practical problems. Lately, researchers have proposed several schemes that use graph data structure and graph algorithms to perform social network analysis [7, 13, 22].



In this paper, we consider the problem of computing graph operations between two parties, which has great potentials in social network analysis. For example, the people and entities in social networks can be represented as the vertices in a graph, and the connections within the social network can be represented as the edges. After that, many data analysis problems can be converted into performing graph operations.

**Our Results.** We study the problem of performing graph operations while protecting the privacies of the graphs. Suppose two parties, Alice and Bob, each has a private graph, denoted as  $G_A$  and  $G_B$ , respectively. Alice wishes to learn the intersection and union of these two graphs. In other words, Alice wish to learn  $G_I = G_A \cap G_B$  and  $G_U = G_A \cup G_B$ . In addition, both Alice and Bob do not wish to reveal any information about their graphs to the other party, beyond the final result. The contributions of this paper can be summarized as below:

- We present two graph operation protocols between two parties, a server and a client. The first protocol allows the server and the client to jointly compute the intersection of their input graphs, and the second protocol computes the union of the input graphs. At the end of the protocols, only the server learns the results. In the protocols, the vertices of the graphs are represented as sorted sets, and the edges are represented as adjacency matrices. Our constructions first use Paillier cryptosystem and oblivious polynomials evaluation to compute the intersection and the union of the vertices. After that, we use the homomorphic property of the Paillier cryptosystem to compute the edge intersection and union.
- We provide the security models of the protocols, and we proof that the protocols are secure against semi-honest adversarial servers.

## 2 Related Work

Secure Multi-party computation (MPC) has been extensively studied over the past years. Generally speaking, MPC allows multiple participants to jointly perform certain computations without losing the privacy of their input data, even when some players cheat during the process. MPC was first formally introduced by Yao in 1982 [21] and extended by Goldreich, Micali, and Wigderson [9]. Their works convert certain computation problems into combinatorial circuit, then the parties perform computations over the gates in the circuit. After that, a large number of MPC protocols have been proposed to solve various problems, such as private information retrieval [6], privacy-preserving set operations [12], and privacy-preserving data mining [1].

Private set operation is a special case of secure multi-party computation that can be applied to a wide range of practical problems. In 2005, Kissner and Song proposed several privacy-preserving set operation protocols [12], including union, intersection, and element reduction operations. Their protocols first use polynomial representation to convert the sets of the parties into polynomials, then encrypt the polynomials using additively homomorphic cryptosystem.

After that, the parties homomorphically perform computations over the ciphertexts. The proposed protocols are secure against semi-honest adversarial players. Moreover, Several other set operation protocols have been proposed, such as testing the disjointness between two sets [8, 11], computing the set intersection cardinality [20], and computing the subset relation [3].

Using graph structures to store and process web data has been extensively studied over the past decades. Representing static pages as vertices and the links between the pages as edges naturally convert web data into directed graphs. Furthermore, various types of graph operations can be used to solve different web problems, including web searching [15], web crawling [5], and data mining [4]. Beyond web data, many other areas start to store data as graph structures in order to convert different computing problems into graph problems, such as social network [10, 14, 18], biological network [17, 19], and communication network [2].

### 3 Preliminary

#### 3.1 Additive Homomorphic Encryption

In our private graph operation protocols, we utilize the Paillier cryptosystem [16], which allows performing additions and multiplications on the ciphertext space. The Paillier cryptosystem contains three algorithms, described as follows:

- $(pk, sk) \leftarrow \text{KeyGen}(1^k)$  is the key generation algorithm. The input is a security parameter  $k$ . The outputs are a public key  $pk$  and a secret key  $sk$ .
- $m^\oplus \leftarrow \text{Enc}(pk, m; r)$  is the encryption algorithm. The input is the public key  $pk$ , a plaintext  $m$  and a random number  $r$ . The output is the ciphertext  $m^\oplus$ . For simplicity, we use the notion  $m^\oplus = \text{Enc}(m)$ .
- $m \leftarrow \text{Dec}(sk, m^\oplus)$  is the decryption algorithm. The input is the secret key  $sk$  and a ciphertext  $m^\oplus$ . The output is the plaintext  $m \in \mathbb{Z}_N$ . For simplicity, we use the notion  $m = \text{Dec}(m^\oplus)$ .

**Correctness.** For any  $(pk, sk) \leftarrow \text{KeyGen}(1^k)$  and any  $m \in \mathbb{Z}_N$ ,  $\text{Dec}(\text{Enc}(m)) = m$  always holds.

**IND-CPA Security.** The ciphertexts of two plaintexts,  $m_0^\oplus$  and  $m_1^\oplus$ , are indistinguishable for probabilistic polynomial-time adversaries that only have access to the public parameters.

**Homomorphic Property.** For any  $m_0, m_1 \in \mathbb{Z}_N$ , there exists an operation  $\oplus$  in the ciphertext space, such that  $\text{Dec}(\text{Enc}(m_0) \oplus \text{Enc}(m_1)) = m_0 + m_1$ . Furthermore, there exists an operation  $\otimes$  in the ciphertext space, such that  $\text{Dec}(\text{Enc}(m_0) \otimes m_1) = m_0 \cdot m_1$ .

#### 3.2 Graph Representation

In our protocol, we represent a graph as  $G = (V, E)$ , where  $V$  is the set of vertices and  $E$  is the set of edges. We represent  $V$  as a sorted set with ascending order,

$V = \{v_1, v_2, \dots, v_m\}$ , where  $m$  is the number of vertices,  $v_i \in \mathbb{Z}$  and  $v_i < v_{i+1}$  for  $1 \leq i \leq m - 1$ . We represent  $E$  as an adjacency matrix,

$$E = \begin{pmatrix} e_{1,1} & \dots & e_{1,m} \\ \vdots & \ddots & \vdots \\ e_{m,1} & \dots & e_{m,m} \end{pmatrix},$$

where  $e_{i,j}$  is the adjacency relation between  $v_i$  and  $v_j$ , and  $e_{i,j} \in \{0, 1\}$ . If  $v_i$  and  $v_j$  are adjacent, i.e. there is at least one edge that connects them,  $e_{i,j} = 1$ , otherwise  $e_{i,j} = 0$ . Note that,  $E$  is a square matrix with  $m$  rows and  $m$  columns. For an undirected graph,  $E$  is a symmetric matrix, since the edges are two-way.

## 4 Private Graph Intersection Protocol

### 4.1 Problem Definition

We formally describe the private graph intersection protocol (PGI). The protocol involves two participants, a server and a client, denoted as  $S$  and  $C$ , respectively. Each of the participants holds a private graph, which is intended to be kept secret from the other participant. We denote the graphs of the server and client as  $G_S = (V_S, E_S)$  and  $G_C = (V_C, E_C)$ , respectively, where  $V$  and  $E$  are the sets of vertices and edges of the graphs. The intersection of the two graphs is defined as  $G_I = (V_I, E_I) = G_S \cap G_C$ , where  $V_I = V_S \cap V_C$  and  $E_I = E_S \cap E_C$ . The private graph intersection protocol allows the server and the client to jointly compute  $G_I$ . At the end of the protocol, only the server learns the result.

We assume that the protocol is performed in the semi-honest setting. In other words, both the server and the client perform the protocol faithfully, but they may try to learn any information about the graph of the other party. While achieving no information leakage is the ideal goal, our protocol leaks certain information during the process. We modeled such leakages as leakage functions  $\mathcal{L}_1$  and  $\mathcal{L}_2$ .  $\mathcal{L}_1$  contains the number of vertices of the client’s graph, and  $\mathcal{L}_2$  contains the vertex intersection  $V_I$  and the number of vertices of the server’s graph. Furthermore, we assume both the server and the client are semi-honest. In other words, the server and the client always follow the protocol faithfully, but they may try to learn any information about the graph of the other party.

**Definition 1 (Private Graph Intersection Protocol).** *Two probabilistic polynomial-time interactive Turing machines,  $S$  and  $C$ , define a private graph intersection Protocol if the following properties hold:*

**Correctness.** *If both participants are honest, for any  $G_S = (V_S, E_S)$  and any  $G_C = (V_C, E_C)$ , the private graph intersection protocol computes  $G_I = G_S \cap G_C$ . At the end of the protocol, only  $S$  learns  $G_I$ .*

**Security.** *A semi-honest server learns nothing about the client’s graph, beyond that can be deduced from  $G_I$  and the leakage function  $\mathcal{L}_1$ , and a semi-honest client learns nothing about the server’s graph, beyond the leakage function  $\mathcal{L}_2$ .*

## 4.2 Construction

In this section, we propose the construction of our private graph intersection protocol. We first use the FNP protocol [8] for computing the vertex intersection, then we use the Paillier cryptosystem to compute the edge intersection.

The server's graph is denoted as  $G_S = (V_S, E_S)$ , where  $V_S = \{s_1, \dots, s_m\}$  and

$$E_S = \begin{pmatrix} s_{1,1} & \cdots & s_{1,m} \\ \vdots & \ddots & \vdots \\ s_{m,1} & \cdots & s_{m,m} \end{pmatrix}.$$

The client's graph is denoted as  $G_C = (V_C, E_C)$ , where  $V_C = \{c_1, \dots, c_n\}$  and

$$E_C = \begin{pmatrix} c_{1,1} & \cdots & c_{1,n} \\ \vdots & \ddots & \vdots \\ c_{n,1} & \cdots & c_{n,n} \end{pmatrix}.$$

The private graph intersection protocol is described below:

*Input:*  $S$  and  $C$  hold the graphs  $G_S = (V_S, E_S)$  and  $G_C = (V_C, E_C)$ , respectively.

*Output:*  $S$  learns  $G_I = (V_I, E_I)$ .

*Protocol:*

**Step 1:**  $S$  runs the key generation algorithm of the Paillier cryptosystem,  $(pk, sk) \leftarrow \text{KeyGen}(1^k)$ . Then  $S$  publishes  $pk$ .

**Step 2:** (a)  $S$  constructs a polynomial  $P(x) = (x - s_1)(x - s_2)\dots(x - s_m) = \sum_{u=0}^m \alpha_u x^u$ , such that all the roots of  $P(x)$  are exactly the elements in  $V_S$ . In other words,  $P(x) = 0$  if and only if  $x \in V_S$ .

(b)  $S$  encrypts each  $\alpha_i$ , for  $0 \leq i \leq m$ , under the Paillier cryptosystem, and sends the set of ciphertexts  $\alpha^\oplus$  to  $C$ .

**Step 3:** (a) By using the homomorphic properties of the Paillier cryptosystem,  $C$  evaluates the polynomial  $P$  using each element in  $V_C$  as input. In other words,  $C$  computes  $\text{Enc}(P(c_i))$ , for  $1 \leq i \leq n$ .

(b) For each polynomial evaluation,  $C$  chooses a random value  $r$  and computes  $m_i^\oplus = \text{Enc}(rP(c_i) + c_i)$ . Then  $C$  sends  $M^\oplus = \{m_i^\oplus\}$  to  $S$ .

**Step 4:**  $S$  decrypts all the ciphertexts received, and compares the decrypted values with his vertex set  $V_S$ . If a decrypted value  $d = \text{Dec}(m_i^\oplus)$  has a corresponding element in  $V_S$ , it is an element of  $V_I$ . In other words, if  $d \in V_S$ ,  $d \in V_I$ . After decrypting all the received ciphertexts, the server obtains  $V_I$ .

**Step 5:** (a)  $S$  uses  $V_I$  to construct an adjacency matrix  $A$  of size  $t \times t$ , where  $t$  is the number of vertex in  $V_I$ :

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,t} \\ \vdots & \ddots & \vdots \\ a_{t,1} & \cdots & a_{t,t} \end{pmatrix}.$$

$A$  has the property that, for each vertex pair  $u_x \in V_I$  and  $u_y \in V_I$ , if an edge exists in  $G_S$  between vertices  $u_x$  and  $u_y$ ,  $a_{x,y} = 1$ ; otherwise  $a_{x,y} = 0$ .

(b)  $S$  encrypts each element in  $A$  under the Paillier cryptosystem, and obtains an encrypted matrix  $A^\oplus = \text{Enc}(A)$ .

(c)  $S$  sends  $A^\oplus$  and  $V_I$  to  $C$ .

**Step 6:** (a) By using  $V_I$ ,  $C$  constructs an adjacency matrix  $B$  using the same method in the last step:

$$B = \begin{pmatrix} b_{1,1} & \cdots & b_{1,t} \\ \vdots & \ddots & \vdots \\ b_{t,1} & \cdots & b_{t,t} \end{pmatrix}.$$

(b)  $C$  computes

$$E_I^\oplus = A^\oplus \otimes B = \begin{pmatrix} a_{1,1}^\oplus & \cdots & a_{1,t}^\oplus \\ \vdots & \ddots & \vdots \\ a_{t,1}^\oplus & \cdots & a_{t,t}^\oplus \end{pmatrix} \otimes \begin{pmatrix} b_{1,1} & \cdots & b_{1,t} \\ \vdots & \ddots & \vdots \\ b_{t,1} & \cdots & b_{t,t} \end{pmatrix} = \begin{pmatrix} a_{1,1}^\oplus \otimes b_{1,1} & \cdots & a_{1,t}^\oplus \otimes b_{1,t} \\ \vdots & \ddots & \vdots \\ a_{t,1}^\oplus \otimes b_{t,1} & \cdots & a_{t,t}^\oplus \otimes b_{t,t} \end{pmatrix}.$$

(c)  $C$  sends  $E_I^\oplus$  to  $S$ .

**Step 7:**  $S$  decrypts each element in  $E_I^\oplus$ , and obtains  $E_I = \text{Dec}(E_I^\oplus)$ . At last,  $S$  obtains  $G_I = (V_I, E_I)$ .

### 4.3 Security Analysis

**Lemma 1 (Correctness).** *If both participants are honest, for any  $G_S = (V_S, E_S)$  and any  $G_C = (V_C, E_C)$ , the private graph intersection protocol computes  $G_I = (V_I, E_I) = G_S \cap G_C$ .*

*Proof.* The correctness of the protocol is ensured by the correctness of the FNP protocol and the homomorphic property of the Paillier encryption scheme.

During **Step 2** to **Step 4** of the protocol, the client and the server jointly perform a FNP protocol using their vertex collections as inputs. At the end of **Step 4**, the server learns the vertex intersection  $V_I$ , and the client receives  $V_I$  from the server in **Step 5**.

In **Step 5** and **Step 6**, the server and the client construct two adjacency matrices by using  $V_I$ , denoted as  $A$  and  $B$ , respectively, which contain the adjacency relations between the vertices in  $V_I$  for graphs  $G_S$  and  $G_C$ . If an edge exists between two vertices in  $V_I$ , it leads to a value of 1 in the corresponding position of the adjacency matrix, otherwise it leads to a value of 0 instead. Therefore, the dot product of  $A$  and  $B$  will produce an adjacency matrix that represents the edge intersection. If an edge exists in both  $A$  and  $B$ , the dot product will result a value of 1. If an edge only exists in one of the input graphs, or the edge does not exist at all, the dot product will result a value of 0.

In **Step 6**, the client receives the encryption of  $A$  under the Paillier cryptosystem from the server. If the Paillier cryptosystem has the homomorphic property, i.e. it supports multiplication between a ciphertext and a constant, the client

can homomorphically compute the dot product of the  $A$  and  $B$ , and the result is the encryption of the edge intersection. Finally, in **Step 7**, the server obtains the edge intersection after decryption.

As a result, if the FNP protocol is correct and the Paillier cryptosystem has the homomorphic property, the private graph intersection protocol computes  $G_I = (V_I, E_I) = G_S \cap G_C$ .

**Lemma 2 (Server Zero-knowledge).** *A semi-honest server learns nothing about the client's graph, beyond that can be deduced from  $G_I$  and the leakage function  $\mathcal{L}_1$ .*

*Proof.* The proof of Server Zero-knowledge is trivial. During the protocol, there are two parts where the server receives information about the client's graph. The first part is during the FNP protocol in **Step 3**, and the second part is at the end of **Step 6**.

For the first part, in **Step 3**, the server receives a set of ciphertexts from the client. The server can learn the number of vertices in the client's graph by counting the number of ciphertexts, which is the pre-defined leakage function  $\mathcal{L}_1$ . By decrypting the ciphertexts, the server obtains a set of values. If a value exists in  $V_S$ , it is a common vertex between the two input graphs, which is a part of the final result of the protocol. Otherwise, if the value does not exist in  $V_S$ , it will be a random value, which has no relation to the client's graph.

For the second part, the server receives  $E_I^\oplus$  from the client, which is the ciphertext of the edge intersection. Upon decryption, the server only learns the edge intersection.

As a result, a semi-honest server learns nothing about the client's graph, beyond that can be deduced from  $G_I$  and the leakage function  $\mathcal{L}_1$ .

**Lemma 3 (Client Zero-knowledge).** *a semi-honest client learns nothing about the server's graph, beyond the leakage function  $\mathcal{L}_2$ .*

*Proof.* There are two parts where the client receives information about the server's graph. The first part is during the FNP protocol in **Step 2**, and the second part is at the end of **Step 5**.

For the first part, the client receives a set of encrypted coefficients  $\alpha^\oplus$  of the polynomial  $P$  from the server. The client can learn the number of vertices of the server's graph by counting the number of encrypted coefficients, which is a part of the pre-defined leakage function  $\mathcal{L}_2$ .

For the second part, the client receives an encrypted matrix  $A^\oplus$  and the vertex intersection  $V_I$ . Since  $V_I$  is also a part of the pre-defined leakage function  $\mathcal{L}_2$ , we need to show that  $A^\oplus$  does not reveal any information about the server's graph.  $A^\oplus$  contains the encryptions of adjacency relations between the vertices in  $V_I$  for the server's graph. Therefore, if the client cannot distinguish between the cases where the server has different input graphs, given the knowledge of  $A^\oplus$  and

$V_I$ , the zero-knowledge for the client holds. Consider the following experiment:

$$\begin{aligned}
 &EXP_{\mathcal{A}}(1^k) : \\
 &\quad (G_0, G_1) \leftarrow \mathcal{A} \\
 &\quad b \xleftarrow{\$} \{0, 1\} \\
 &\quad (pk, sk) \leftarrow \text{Step 1}(1^k) \\
 &\quad \alpha^\oplus \leftarrow \text{Step 2}(G_b, pk) \\
 &\quad M^\oplus \leftarrow \text{Step 3}(\alpha^\oplus, G_C) \\
 &\quad V_I \leftarrow \text{Step 4}(M^\oplus, sk) \\
 &\quad A^\oplus \leftarrow \text{Step 5}(G_b, V_I, pk) \\
 &\quad \hat{b} \leftarrow \mathcal{A}(\alpha^\oplus, V_I, A^\oplus) \\
 &\quad \text{if } \hat{b} = b, \text{ output } 1 \\
 &\quad \text{otherwise, output } 0
 \end{aligned}$$

In the above experiment,  $\mathcal{A}$  is a probabilistic polynomial-time adversarial client with a private graph  $G_C = (E_C, V_C)$ . The adversary first chooses two graphs, denoted as  $G_0 = (V_0, E_0)$  and  $G_1 = (V_1, E_1)$ , respectively. The two graphs have the property that  $V_0 \cap V_C = V_1 \cap V_C$  and  $|V_0| = |V_1|$ .  $\mathcal{A}$  then sends the graphs to the server. The server randomly picks a bit  $b = \{0, 1\}$ , and choose  $G_b$  as his private graph. After that, the server and  $\mathcal{A}$  jointly perform the private graph intersection protocol from **Step 1** to **5**.

At the end of **Step 5**,  $\mathcal{A}$  needs to output a bit  $\hat{b}$ , using the information he received during the protocol. If  $\hat{b} = b$ , the experiment outputs 1, otherwise outputs 0. The advantage of the above experiment for  $\mathcal{A}$  is defined as  $Adv_{\mathcal{A}} = |\Pr[EXP_{\mathcal{A}}(1^k) = 1] - \frac{1}{2}|$ .

During the protocol, the information that  $\mathcal{A}$  receives contains  $\alpha^\oplus$ ,  $V_I$ , and  $A^\oplus$ .  $\alpha^\oplus$  contains a set of ciphertexts under the Paillier cryptosystem,  $V_I$  is the vertex intersection, and  $A^\oplus$  is an encrypted adjacency matrix.

Due to the condition  $V_0 \cap V_C = V_1 \cap V_C$ , the vertex intersection  $V_I$  gives no useful information since  $V_I$  will be the same for both  $G_0$  and  $G_1$ . Since the Paillier cryptosystem is IND-CPA secure and  $\mathcal{A}$  cannot decrypt the ciphertexts without the private key,  $\alpha^\oplus$  and  $A^\oplus$  cannot help  $\mathcal{A}$  to distinguish which graph the server has chosen. As a result, if the Paillier cryptosystem is IND-CPA secure, the advantage of the above experiment for  $\mathcal{A}$  is negligible, i.e.  $Adv_{\mathcal{A}} = |\Pr[EXP_{\mathcal{A}}(1^k) = 1] - \frac{1}{2}| = \varepsilon$ , where  $\varepsilon$  is negligible.

At last, we construct a simulator  $Sim_S$  to simulate the view of the client in the ideal model.  $Sim_S$  is given the knowledge of the vertex intersection  $V_I$  and the vertex number  $m$  of the server's graph. In the experiment,  $Sim_S$  sends a set of  $m + 1$  random values to the client in **Step 2**, and sends  $V_I$  and a matrix with  $t \times t$  random values to the client in **Step 5**. Since the client cannot distinguish between the ciphertexts under the Paillier encryption scheme and random values, the view of the client in the ideal model is computationally indistinguishable from the view in the real model, i.e.  $View_C^{real}[S(G_S), C] \approx View_C^{ideal}[Sim_S(V_I, m), C]$ .

As a result, if the Paillier cryptosystem is IND-CPA secure, a semi-honest client learns nothing about the server's graph, beyond the leakage function  $\mathcal{L}_2$ .

## 5 Private Graph Union Protocol

### 5.1 Problem Definition

The setting of our private graph union protocol (PGU) is similar to the private graph intersection protocol. The only difference is that the server and the client wish to compute the union of their private graphs. The union of two graphs is defined as  $G_U = G_S \cup G_C = (V_U, E_U)$ , where  $V_U = V_S \cup V_C$  and  $E_U = E_S \cup E_C$ . At the end of the protocol, only the server learns the result.

We modeled the leakages of PGU as leakage functions  $\mathcal{L}_3$  and  $\mathcal{L}_4$ .  $\mathcal{L}_3$  contains the number of vertices of  $G_C$  and the number of common vertices, and  $\mathcal{L}_4$  contains the vertex union  $V_U$  and the number of vertices of the  $G_S$ . Furthermore, we also assume that the protocol is performed in the semi-honest setting.

**Definition 2 (Private Graph Union Protocol).** *Two probabilistic polynomial-time interactive Turing machines,  $S$  and  $C$ , define a private graph union protocol if the following properties hold:*

**Correctness.** *If both parties are honest, for any  $G_S = (V_S, E_S)$  and any  $G_C = (V_C, E_C)$ , the private graph union protocol computes  $G_U = G_S \cup G_C$ . At the end of the protocol, only  $S$  learns the result.*

**Security.** *A semi-honest server learns nothing about the client's graph, beyond that can be deduced from  $G_U$  and the leakage function  $\mathcal{L}_3$ , and a semi-honest client learns nothing about the server's graph, beyond the leakage function  $\mathcal{L}_4$ .*

### 5.2 Construction

Similar as the private graph intersection protocol, the graphs of the server and the client are represented as  $G_S$  and  $G_C$ , respectively. The private graph union protocol is described below:

*Input:*  $S$  and  $C$  hold the graphs  $G_S = (V_S, E_S)$  and  $G_C = (V_C, E_C)$ , respectively.

*Output:*  $S$  learns  $G_U = (V_U, E_U)$ .

*Protocol:*

**Step 1:**  $S$  runs the  $(pk, sk) \leftarrow \text{KeyGen}(1^k)$  algorithm, and obtains the public key and the secret key of the Paillier cryptosystem. Then  $S$  publishes  $pk$ .

**Step 2:** (a)  $S$  constructs a polynomial  $P(x) = (x - s_1)(x - s_2)\dots(x - s_m) = \sum_{u=0}^m \alpha_u x^u$ . All the roots of  $P(x)$  are exactly the elements in  $V_S$ . In other words,  $P(x) = 0$  if and only if  $x \in V_S$ .

(b)  $S$  encrypts each  $\alpha_i$ , for  $0 \leq i \leq m$ , under the Paillier cryptosystem, and sends the set of ciphertexts  $\alpha^\oplus = \{\alpha_i^\oplus\}_{0 \leq i \leq m}$  to  $C$ .



**Step 3:** (a) By using the homomorphic properties of the Paillier encryption scheme,  $C$  evaluates the polynomial  $P$  using each element in  $V_C$  as input. In other words,  $C$  computes  $\text{Enc}(P(c_i))$ , for  $1 \leq i \leq n$ .

(b) For each polynomial evaluation,  $C$  choose a random value  $r$  and computes  $m_i^\oplus = \text{Enc}(P(c_i)) \otimes r$ . Then  $C$  sends the set of all resulting ciphertexts  $M^\oplus = \{m_i^\oplus\}_{1 \leq i \leq n}$  to  $S$ .

**Step 4:**  $S$  decrypts each ciphertext received as  $m_i = \text{Dec}(m_i^\oplus)$ , and checks the decrypted value. If  $m_i = 0$ ,  $S$  computes  $n_i^\oplus = \text{Enc}(0)$ ; otherwise,  $S$  computes  $n_i^\oplus = \text{Enc}(1)$ . Then  $S$  sends  $N^\oplus = \{n_i^\oplus\}_{1 \leq i \leq n}$  to  $C$ .

**Step 5:** After receiving  $N^\oplus$ ,  $C$  computes  $d_i^\oplus = c_i \otimes n_i^\oplus$ , for  $1 \leq i \leq n$ . Then  $C$  sends  $D^\oplus = \{d_i^\oplus\}_{1 \leq i \leq n}$  to  $S$ .

**Step 6:** (a)  $S$  decrypts each value in  $D^\oplus$ , and checks if the decrypted value  $d_i = \text{Dec}(d_i^\oplus)$  is zero.

(b) By combining the server's vertex set  $V_S$  and the set of non-zero decrypted values  $\{d_i\}_{d_i \neq 0}$ ,  $S$  obtains  $V_U$ .  $V_U$  is then sorted in ascending order, and is represented as  $V_U = \{u_1, u_2, \dots, u_t\}$ .

**Step 7:** (a)  $S$  uses  $V_U$  to construct an adjacency matrix  $A$  of size  $t \times t$ :

$$A = \begin{pmatrix} a_{1,1} & \dots & a_{1,t} \\ \vdots & \ddots & \vdots \\ a_{t,1} & \dots & a_{t,t} \end{pmatrix}.$$

$A$  has the property that, for each vertex pair  $u_x \in V_U$  and  $u_y \in V_U$ , if an edge exists in  $G_S$  between vertices  $u_x$  and  $u_y$ ,  $a_{x,y} = 1$ ; otherwise  $a_{x,y} = 0$ .

(b)  $S$  encrypts each element in  $A$  under the Paillier cryptosystem, and sends the encrypted matrix  $A^\oplus$  and  $V_U$  to  $C$ .

**Step 8:** (a)  $C$  uses  $V_U$  to construct an adjacency matrix  $B$  in the same manner as  $S$  in the last step:

$$B = \begin{pmatrix} b_{1,1} & \dots & b_{1,t} \\ \vdots & \ddots & \vdots \\ b_{t,1} & \dots & b_{t,t} \end{pmatrix}.$$

(b)  $C$  encrypts each element in  $B$  using the Paillier cryptosystem, and obtains  $B^\oplus$ .

(c)  $C$  generates a matrix  $R$  with  $t \times t$  random values:

$$R = \begin{pmatrix} r_{1,1} & \dots & r_{1,t} \\ \vdots & \ddots & \vdots \\ r_{t,1} & \dots & r_{t,t} \end{pmatrix}.$$

(d)  $C$  computes:

$$\begin{aligned}
 E_U^\oplus &= (A^\oplus \oplus B^\oplus) \otimes R = \begin{pmatrix} (a_{1,1}^\oplus \oplus b_{1,1}^\oplus) \otimes r_{1,1} \cdots (a_{1,t}^\oplus \oplus b_{1,t}^\oplus) \otimes r_{1,t} \\ \vdots \quad \quad \quad \ddots \quad \quad \quad \vdots \\ (a_{t,1}^\oplus \oplus b_{t,1}^\oplus) \otimes r_{t,1} \cdots (a_{t,t}^\oplus \oplus b_{t,t}^\oplus) \otimes r_{t,t} \end{pmatrix} \\
 &= \begin{pmatrix} e_{1,1}^\oplus \cdots e_{1,t}^\oplus \\ \vdots \quad \quad \quad \vdots \\ e_{t,1}^\oplus \cdots e_{t,t}^\oplus \end{pmatrix}.
 \end{aligned}$$

(e)  $C$  sends  $E_U^\oplus$  to  $S$ .

**Step 9:**  $S$  decrypts the matrix  $E_U^\oplus$ . For each decrypted element  $e_{i,j}$ , if  $e_{i,j} \neq 0$ , set  $e_{i,j} = 1$ . At last,  $S$  obtains  $E_U$ .

### 5.3 Security Analysis

**Lemma 4 (Correctness).** *If both participants are honest, for any  $G_S = (V_S, E_S)$  and any  $G_C = (V_C, E_C)$ , the private graph union protocol computes  $G_U = (V_U, E_U) = G_S \cup G_C$ .*

*Proof.* The correctness of the protocol is ensured by the homomorphic property of the Paillier encryption scheme. **Step 2–6** of the protocol compute the vertex union, and **Step 7–9** compute the edge union.

In order to compute the vertex union, the server needs to obtain the vertices in  $G_C$  that is not in  $G_S$ . In **Step 2**, the server constructs a polynomial, such that the roots are exactly the vertices in  $G_S$ . After that, the client homomorphically evaluates the polynomial using all the vertices in  $G_C$ , and each polynomial evaluation is homomorphically multiplied by a random value. Therefore, the common vertices between  $G_S$  and  $G_C$  will result encryptions of zero, and other vertices will result encryptions of random values. In **Step 4**, the server decrypts all the polynomial evaluations. If the decryption is zero, the server generates an encryption of 0; otherwise, the server generates an encryption of 1. In the next step, the client homomorphically multiplies the received encryptions with the vertices in  $V_C$ . For an encryption of 0, the client will result an encryption of 0; for an encryption of 1, the client will result an encryption of the vertex. As a result, in **Step 6**, the server learns the set of vertices that only exists in  $G_C$ . By combining the above set and  $V_S$ , the server obtains the vertex union  $V_U$ .

In order to compute the edge union, the server needs to obtain an adjacency matrix, such that if an edge is not exist in neither of  $G_S$  and  $G_C$ , it will have a corresponding value of 0 in the matrix; otherwise it will have a corresponding value of 1. In **Step 7** and **8**, each of the server and the client constructs an adjacency matrix using the vertex union and his own graph, and encrypts each element. The client then homomorphically adds the encrypted values at the same locations in the two matrices. There are three circumstances for the addition results. If an edge is not exist in neither of the graphs, the addition will result

an encryption of 0; if an edge only exists in one of the graphs, the addition will result an encryption of 1; if an edge exists in both of the graphs, the addition will result an encryption of 2. Then the client homomorphically multiplies each result by a random value. Therefore, for the edges that are not exist in neither of the graphs, the result will still be an encryption of 0; for the edges that only exist in one of the graphs and the edges that exist in both of the graphs, the result will be encryptions of random values. Finally, in **Step 9**, the server decrypts the encrypted matrix, and replace all non-zero values to 1, which is  $E_U$ .

As a result, if the Paillier cryptosystem has the homomorphic property, the server learns the graph union  $G_U = (V_U, E_U)$  at the end of the protocol.

**Lemma 5 (Server Zero-knowledge).** *A semi-honest server learns nothing about the client's graph, beyond that can be deduced from  $G_U$  and the leakage function  $\mathcal{L}_3$ .*

*Proof.* There are three parts where the server receives information from the client, **Step 3**, **5** and **8**.

In **Step 3**, the server receives a set of ciphertexts,  $M^\oplus$ , from the client. Each vertex in  $V_C$  has a corresponding ciphertext in  $M^\oplus$ . If a vertex in  $V_C$  also exists in  $V_S$ , i.e. it is a common vertex in both graphs, it will result an encryption of 0; otherwise, it will result an encryption of a random value. By counting the number of ciphertexts in  $M^\oplus$ , the server can learn the number of vertices in  $G_C$ , and by decrypting and counting the number of 0s, the server can learn the number of common vertices. The above information is defined as leakage function  $\mathcal{L}_3$ .

In **Step 5**, the server receives a set of ciphertexts  $D^\oplus$  from the client. Each vertex in  $V_C$  has a corresponding ciphertext in  $D^\oplus$ . If a vertex exists in both  $V_S$  and  $V_C$ , it will result an encryption of 0; otherwise, it will result an encryption of the vertex itself. Therefore, upon decryption, the server learns the vertices in  $V_C$  that does not exist in  $V_S$ , which is a part of the vertex union.

In **Step 8**, the server receives an encrypted matrix  $E_U^\oplus$  from the client. Each element of  $E_U^\oplus$  represents the adjacency relation between two vertices in  $G_U$ . If an edge exists in at least one of the input graphs, the corresponding adjacency value will be a random number; if an edge does not exists in neither of the input graphs, it will result an adjacency value of 0. By decrypting the matrix and replacing the random values to 1, the server obtains the edge union.

As a result, a semi-honest server learns nothing about the client's graph, beyond that can be deduced from  $G_U$  and the pre-defined leakage function  $\mathcal{L}_3$ .

**Lemma 6 (Client Zero-knowledge).** *A semi-honest client learns nothing about the server's graph, beyond that can be deduced from  $V_U$  and the leakage function  $\mathcal{L}_4$ .*

*Proof.* There are three parts where the client receives information from the server, **Step 2**, **4** and **7**. In **Step 2**, the client receives a set  $\alpha^\oplus$  that contains  $m + 1$  ciphertexts, which are encryptions of the coefficients of the server's polynomial. The client can learn the vertex number of the server's graph by counting the cipertexts in  $\alpha^\oplus$ , which is the leakage function  $\mathcal{L}_4$ . In **Step 4**, the

client receives another set of ciphertexts  $N^\oplus$ , which contains  $n$  encryptions of 1s and 0s. In **Step 7**, the client receives an encrypted matrix of size  $t \times t$ , which contains encryptions of 1s and 0s. In order to prove that the above information does not reveal anything about the server's graph beyond that can be deduced from  $V_U$  and the leakage function  $\mathcal{L}_4$ , consider the following experiment:

$$\begin{aligned}
 & EXP_{\mathcal{A}}(1^k) : \\
 & (G_0, G_1) \leftarrow \mathcal{A} \\
 & b \xleftarrow{\$} \{0, 1\} \\
 & (pk, sk) \leftarrow \text{Step 1}(1^k) \\
 & \alpha^\oplus \leftarrow \text{Step 2}(G_b, pk) \\
 & M^\oplus \leftarrow \text{Step 3}(\alpha^\oplus, G_C) \\
 & N^\oplus \leftarrow \text{Step 4}(M^\oplus, pk, sk) \\
 & D^\oplus \leftarrow \text{Step 5}(N^\oplus, G_C) \\
 & V_U \leftarrow \text{Step 6}(D^\oplus, sk, G_b) \\
 & A^\oplus \leftarrow \text{Step 7}(G_b, V_U, pk) \\
 & \hat{b} \leftarrow \mathcal{A}(\alpha^\oplus, N^\oplus, A^\oplus, V_U) \\
 & \text{if } \hat{b} = b, \text{ output } 1 \\
 & \text{otherwise, output } 0
 \end{aligned}$$

In the above experiment,  $\mathcal{A}$  is a probabilistic polynomial-time adversarial client with a private graph  $G_C = (E_C, V_C)$ . The adversary first chooses two graphs, denoted as  $G_0 = (V_0, E_0)$  and  $G_1 = (V_1, E_1)$ , respectively.  $G_0$  and  $G_1$  have the property that  $V_0 \cup V_C = V_1 \cup V_C$  and  $|V_0| = |V_1|$ .  $\mathcal{A}$  then sends the graphs to the server. The server randomly picks a bit  $b = \{0, 1\}$ , and choose  $G_b$  as his private graph. After that, the server and  $\mathcal{A}$  jointly perform the private graph union protocol from **Step 1** to **7**. At the end of **Step 7**,  $\mathcal{A}$  needs to output a bit  $\hat{b}$ , using the information he received during the protocol. If  $\hat{b} = b$ , the experiment outputs 1, otherwise outputs 0. The advantage of the above experiment for  $\mathcal{A}$  is defined as  $Adv_{\mathcal{A}} = |\Pr[EXP_{\mathcal{A}}(1^k) = 1] - \frac{1}{2}|$ .

During the protocol, the information that  $\mathcal{A}$  receives contains  $\alpha^\oplus, N^\oplus, A^\oplus$ , and  $V_U$ .  $\alpha^\oplus$  and  $N^\oplus$  are both sets of ciphertexts. Since  $G_0$  and  $G_1$  satisfied the condition  $|V_0| = |V_1|$ , the numbers of ciphertexts in  $\alpha^\oplus$  will be the same for both  $G_0$  and  $G_1$ .  $A^\oplus$  is a matrix filled with  $t \times t$  ciphertexts. Since  $\mathcal{A}$  cannot decrypt the ciphertexts without the private key,  $\alpha^\oplus, N^\oplus$  and  $A^\oplus$  cannot help  $\mathcal{A}$  to distinguish which graph the server has chosen. Furthermore, since  $G_0$  and  $G_1$  satisfied the condition  $V_0 \cup V_C = V_1 \cup V_C$ ,  $V_U$  will be the same for both  $G_0$  and  $G_1$ . As a result, if the Paillier cryptosystem is IND-CPA secure, the advantage of the above experiment for  $\mathcal{A}$  is negligible, i.e.  $Adv_{\mathcal{A}} = |\Pr[EXP_{\mathcal{A}}(1^k) = 1] - \frac{1}{2}| = \varepsilon$ , where  $\varepsilon$  is negligible.

At last, we construct a simulator  $Sim_S$  to simulate the view of the client in the ideal model.  $Sim_S$  is given the knowledge of  $V_U$  and the vertex number  $m$  of

$G_S$ . In the ideal model,  $Sim_S$  generates a set of  $m+1$  random values in **Step 2**, a set of  $n$  random values in **Step 4** and a matrix of size  $t \times t$  filled with random values in **Step 7**. Since the Paillier cryptosystem is IND-CPA secure, the client cannot distinguish the ciphertexts and random values. Therefore, the view of the client in the ideal model is computationally indistinguishable from the view in the real model, i.e.  $View_C^{real}[S(G_S), C] \approx View_C^{ideal}[Sim_S(V_U, m), C]$ .

As a result, if the Paillier cryptosystem is IND-CPA secure, the client learns nothing about the server's graph, beyond that can be deduced from  $V_U$  and the leakage function  $\mathcal{L}_4$ .

## 6 Conclusion

In this work, we proposed two privacy-preserving graph operation protocols, which can be used for social network analysis. The first protocol allows a server and a client to jointly compute the intersection between their private graphs, while the second protocol computes the union of the graphs. The protocols first use polynomial representations and oblivious polynomial evaluation to compute the intersection and union of the vertices. The intersection and union of the edges are then computed by using the Paillier cryptosystem. We proved that the proposed protocols are secure in the semi-honest setting.

**Acknowledgement.** This work is supported by the National Natural Science Foundation of China (61872069) and the Fundamental Research Funds for the Central Universities (N2017012).

An earlier version of this paper was presented at the 22nd Australasian Conference on Information Security and Privacy, 2017 [23].

## References

1. Agrawal, R., Srikant, R.: Privacy-preserving data mining. ACM SIGMOD Rec. **29**, 439–450 (2000)
2. Ahlswede, R., Cai, N., Li, S.Y., Yeung, R.W.: Network information flow. IEEE Trans. Inf. Theory **46**(4), 1204–1216 (2000)
3. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. Conjunctive, subset, and range queries on encrypted data, vol. 4392, pp. 535–554. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-70936-7\\_29](https://doi.org/10.1007/978-3-540-70936-7_29)
4. Buehrer, G., Chellapilla, K.: A scalable pattern mining approach to web graph compression with communities. In: Proceedings of the 2008 International Conference on Web Search and Data Mining, pp. 95–106. ACM (2008)
5. Chakrabarti, S., Van den Berg, M., Dom, B.: Focused crawling: a new approach to topic-specific web resource discovery. Comput. Netw. **31**(11–16), 1623–1640 (1999)
6. Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: Proceedings of IEEE 36th Annual Foundations of Computer Science, pp. 41–50. IEEE (1995)
7. Fan, W., et al.: Graph neural networks for social recommendation. In: The World Wide Web Conference, pp. 417–426 (2019)

8. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 1–19. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-24676-3\\_1](https://doi.org/10.1007/978-3-540-24676-3_1)
9. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: STOC (1987)
10. Khalajzadeh, H., Yuan, D., Grundy, J., Yang, Y.: Cost-effective social network data placement and replication using graph-partitioning. In: IEEE International Conference on Cognitive Computing (2017)
11. Kiayias, A., Mitrofanova, A.: Testing disjointness of private datasets. In: Patrick, A.S., Yung, M. (eds.) FC 2005. LNCS, vol. 3570, pp. 109–124. Springer, Heidelberg (2005). [https://doi.org/10.1007/11507840\\_13](https://doi.org/10.1007/11507840_13)
12. Kissner, L., Song, D.: Privacy-preserving set operations. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 241–257. Springer, Heidelberg (2005). [https://doi.org/10.1007/11535218\\_15](https://doi.org/10.1007/11535218_15)
13. Li, Y., Constantin, C., Du Mouza, C.: SGVCut: a vertex-cut partitioning tool for random walks-based computations over social network graphs. In: Proceedings of the 29th International Conference on Scientific and Statistical Database Management, pp. 1–4 (2017)
14. Myers, S.A., Sharma, A., Gupta, P., Lin, J.: Information network or social network?: the structure of the twitter follow graph. In: Proceedings of the 23rd International Conference on World Wide Web, pp. 493–498. ACM (2014)
15. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. Technical report, Stanford InfoLab (1999)
16. Paillier, P.: Public-Key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). [https://doi.org/10.1007/3-540-48910-X\\_16](https://doi.org/10.1007/3-540-48910-X_16)
17. Pavlopoulos, G.A., et al.: Using graph theory to analyze biological networks. *BioData Min.* **4**(1), 10 (2011). <https://doi.org/10.1186/1756-0381-4-10>
18. Rong, H., Ma, T., Tang, M., Cao, J.: A novel subgraph  $K^+$ -isomorphism method in social network based on graph similarity detection. *Soft Comput.* **22**(8), 2583–2601 (2017). <https://doi.org/10.1007/s00500-017-2513-y>
19. Tian, Y., Mceachin, R.C., Santos, C., States, D.J., Patel, J.M.: SAGA: a subgraph matching tool for biological graphs. *Bioinformatics* **23**(2), 232–239 (2006)
20. Vaidya, J., Clifton, C.: Secure set intersection cardinality with application to association rule mining. *J. Comput. Secur.* **13**(4), 593–622 (2005)
21. Yao, A.C.C.: Protocols for secure computations. *FOCS* **82**, 160–164 (1982)
22. Zhang, X., et al.: Predicting happiness state based on emotion representative mining in online social networks. In: Kim, J., Shim, K., Cao, L., Lee, J.-G., Lin, X., Moon, Y.-S. (eds.) PAKDD 2017. LNCS (LNAI), vol. 10234, pp. 381–394. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-57454-7\\_30](https://doi.org/10.1007/978-3-319-57454-7_30)
23. Zhou, F., Xu, Z., Li, Y., Xu, J., Peng, S.: Private graph intersection protocol. In: Pieprzyk, J., Suriadi, S. (eds.) ACISP 2017. LNCS, vol. 10343, pp. 235–248. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-59870-3\\_13](https://doi.org/10.1007/978-3-319-59870-3_13)



# The Movie Recommendation System Based on Differential Privacy

Min Li<sup>1</sup>, Yingming Zeng<sup>2</sup>, Yue Guo<sup>1</sup>, and Yun Guo<sup>1</sup>(✉)

<sup>1</sup> College of Cyber Science, Nankai University, Tianjin, China  
{limintj, 2120180514, guoyun}@nankai.edu.cn

<sup>2</sup> Hangtian INC, Beijing, China  
yingmingblue@163.com

**Abstract.** In the past decades, the ever-increasing popularity of the Internet has led to an explosive growth of information, which has consequently led to the emergence of recommendation systems. A series of encryption measures were adopted in the current recommendation systems in the cloud to protect users' privacy security. However, there are many other privacy attacks in this recommendation system of the cloud-based device. Therefore, this paper studies the encryption interference of setting differential privacy protection mechanism for user data in user's private devices based on untrusted servers. A dynamic privacy budget allocation method is proposed based on localized differential privacy protection technology and specific scenes recommended by movies.

**Keywords:** Differential privacy · Privacy budget · Movie recommendation · Collaborative

## 1 Introduction

With the development of information technology, users have a variety of ways to obtain information. What users spend the most time on the thing that no longer where to get information, but to find the content they are interested in among the numerous information. In this environment, recommendation system emerges as The Times require. Personalized movie recommendation service has been widely used nowadays. Personalized recommendation system usually needs a lot of user data to provide high quality recommendation services. And the data leakage means the user's privacy leakage. Most existing recommendation systems [1–6], such as Netflix movie recommendation system, are based on scenes that are trusted by servers. In the most common collaborative filtering algorithms, trusted servers need to collect all user data and analyze user behavior to perform such personalized recommendations. In the recommended method above, only privacy protection scenarios are considered when publishing to three parties, and no more scenarios of being attacked are taken into account. For example, when user data is transferred from the device to the cloud, an attacker can eavesdrop on the transmission channel and launch a “man-in-the-middle attack,” so the data should be encrypted during transmission. In addition, attackers can directly hack into the cloud, servers and steal user

data. This requires some encryption algorithms to protect the data stored on the cloud. In addition, people inside the server may also leak user data. Under this recommendation mode, users' privacy data cannot be effectively protected.

Therefore, this paper studies the encryption interference of setting differential privacy protection mechanism for user data on user's private device [7] based on untrusted server. In this paper, on the basis of existing research, the difference of privacy protection technology and classic movie recommendation algorithm, emphatically explores the application of localization difference privacy protection technology to solve the privacy issue in movie recommendation algorithm, the main contributions include: (1) based on localization difference privacy protection technology and film recommend specific scenarios, puts forward a method to dynamically allocate budget for privacy. The equal probability of the user's viewing frequency of the movie type is assigned to each node of the privacy prefix tree, and then the noise satisfying Laplace distribution is added according to the allocated privacy budget. (2) the user-based collaborative filtering algorithm is improved according to the actual movie scenes. In the process of user similarity calculation, a matrix similarity calculation method is used instead of the traditional vector-based similarity calculation method to find the similar group of target users. Let's define this process as DP-MRE (Differential Privacy-Movie Recommendation System).

## 2 Theoretical Basis

### 2.1 Differential Privacy Definition

Dwork defined differential privacy [8–10] as a method similar to data encryption in 2006. Differential privacy assumes that the attacker owns all the information except the target information.

Let the data set  $D_1$  and  $D_2$  have the same property structure, and the symmetry difference between them is denoted as  $D_1 \Delta D_2$ , and  $|D_1 \Delta D_2|$  denotes the number of records in  $D_1 \Delta D_2$ . If  $|D_1 \Delta D_2| = 1$ , then  $D_1$  and  $D_2$  are said to be adjacent data sets.

**Define 1  $\varepsilon$ -Differential Privacy.** Assume  $\varepsilon > 0$  is a real number and  $M$  is a random algorithm that takes the data set as input.  $M(x)$  is a query result obtained for the random algorithm  $M$ .  $R$  is a subset of  $M$  of  $x$ . For all adjacent data sets  $D_1$  and  $D_2$  as well as all subsets  $R$  of  $M(x)$  of non-single element, the algorithm  $M$  satisfies the  $\varepsilon$ -differential privacy if the following equation is satisfied:

$$\Pr[M(D_1) \in R] \leq e^\varepsilon \times \Pr[M(D_2) \in R]$$

**Nature 1 (Sequence).** With algorithm  $M_1, M_2, \dots, M_n$ , its privacy protection budget respectively  $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ , so for the same data set  $D$ , composed of these algorithms combination algorithm of  $M(M_1(D), M_2(D), \dots, M_n(D))$  provide  $\sum_{i=1}^n \varepsilon_i$ -differential privacy protection, provide privacy protection level for the sum of total budget.

**Laplace Noise Mechanism.** When the initial query results are obtained, the Laplace mechanism implements  $\varepsilon$ -differential privacy protection by adding noise following the



Laplace distribution to the original results. The mean value is 0, the Laplace distribution of the scale parameter is  $Lap(\sigma)$ , and its probability density function is:

$$p(x) = \frac{1}{2\sigma} \exp\left(-\frac{|x|}{\sigma}\right)$$

### 2.2 Differential Privacy Definition

The movie recommendation system based on differential privacy protection proposed in this paper combines with the characteristic structure of prefix Tree [11] to construct (DP-Tree) based on the historical record information of users' watching movies. The privacy prefix Tree is an improved prefix Tree. The data structure of the privacy prefix Tree is shown in Fig. 1:

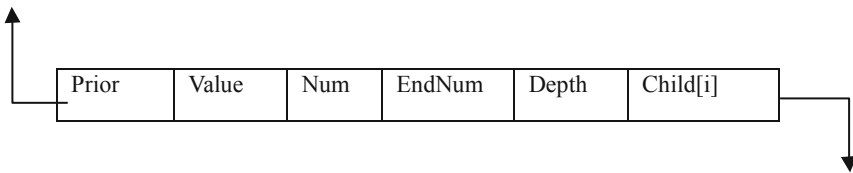


Fig. 1. Data structure of privacy prefix tree

In Fig. 1, *Prior* is the previous pointer pointing to the parent node; *Value* is the stored Value; *Num* is the number of times the value appears; *Depth* is the Depth of the value; *Child[i]* is an array of Pointers to Child nodes, and *EndNum* stores the current node in each path as the number of end nodes. The records of all movie types watched by users are abstracted into a privacy prefix Tree (DP-Tree) whose Root node is Root. Each node in the tree represents a movie type.

## 3 Design of the Differential Privacy Protection Method

### 3.1 The Steps of the Differential Privacy Protection Method

In the differential privacy protection method, the main steps are roughly divided into two steps: the first step is to select the appropriate privacy budget parameters and allocate the appropriate privacy budget for the protected data. The second is to add some noise interference to the protected data.

The amount of noise added is closely related to the allocation of the privacy budget  $\epsilon$ . The value of the privacy budget  $\epsilon$  is inversely proportional to the added noise. The privacy budget not only determines the level of differential privacy protection, but also determines the noise addition, which is the core parameter of differential privacy protection method. This article focuses on how to allocate your privacy budget. For the purpose of this article is based on the difference of privacy movie recommendation system, depending on the type of users to watch film of history data privacy prefix tree structure, privacy prefix tree

high frequency sequence of film type and, in large probability on this type of movie for a user’s interest degree is higher, the probability of being attacked the greater, in order to prevent privacy budget be exhausted and need budget to allocate more commonly used data privacy.

### 3.2 Privacy Budget Allocation Scheme Based on Prefix Tree

The film recommendation system based on differential privacy introduced in this paper is based on the data under the tree structure for protection and encryption. As shown in Fig. 2, is the information of users based on the prefix tree structure chart, of which the user to watch the movie, in accordance with the film type extract features, and in accordance with the prefix tree constructed the privacy prefix tree structure, the specific method is that the user watched a movie, extract the genre of the film, storage in the form of sequence to the tree of the substructure, each film type series combination is a path in a tree, and record the frequencies of each node and the frequencies of each node as a pseudo leaf node. In order to reasonably allocate the privacy budget to the privacy prefix tree, this paper allocates the privacy budget to each node in the privacy prefix tree in an equal proportion allocation method. In the privacy prefix tree, the abstract root node R is not a real movie type, so it will not consume the privacy budget. All other subtree nodes need to be assigned a privacy budget.

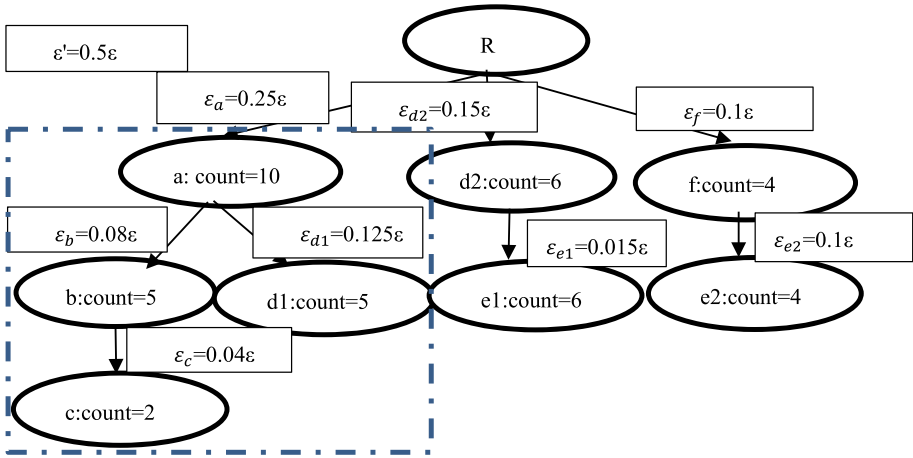


Fig. 2. Prefix tree privacy budget allocation scheme

Instead of storing the movie type directly in the prefix tree, the corresponding letter representation of the movie type is stored, as shown in Table 1.

**Table 1.** Mapping table of movie type letters

Film type	Love	Suspense	Action	Comedy	Plot	Ragedy
Letters mapping	a	b	c	d	e	f

The data of each node in the privacy prefix Tree (DP-Tree) structure shown in Fig. 2 is shown in Table 2.

**Table 2.** DP-Tree data structure

Prior	Value	Num	EndNum	Depth	Child[i]
-	R	-	-	0	[a, d2, f]
R	a	10	0	1	[b, d1]
a	b	5	3	2	[c]
b	c	2	2	3	-
a	d1	5	5	2	-
R	d2	6	0	1	[e1]
d2	e1	6	6	2	-
f	e2	4	4	2	-
R	f	4	0	1	[e2]

As shown in Fig. 2, assuming that the total privacy budget of the whole tree is, viewing frequencies of movie types a, d2 and f are 10, 6 and 4 respectively from the first layer structure of the tree. Then, the total privacy budget allocation proportion of the tree with node a as the root node is  $(10/20)\epsilon$ , the privacy budget is allocated in the same way to  $\epsilon_b = (0.5 * 0.5 * 0.6)/2\epsilon$ . When a movie type is distributed in different sequences, the privacy budget of the movie type is equal to the sum of its allocated privacy budget, for example, the privacy budget of movie type  $\epsilon_d = \epsilon_{d1} + \epsilon_{d2} = 0.125\epsilon + 0.15\epsilon = 0.275\epsilon$ . According to the nature 1 of the differential privacy protection method, it can be concluded that:

$$\epsilon = \epsilon_a + \epsilon_b + \dots + \epsilon_f$$

It can be seen that compared with other privacy budget allocation methods [12–16], this way of allocating privacy budget based on prefix tree based on the value of each node, not on the level structure alone. This allocation method can reasonably dynamically allocate the privacy budget in the case of big differences in tree structure, and it does not need to adjust the value of privacy budget allocation artificially.

### 3.3 Prefix Tree Privacy Budget Allocation Algorithm

The privacy budget allocation algorithm based on the prefix tree is as follows. Where,  $TMovie$  stores the result of privacy budget allocation of movie-type nodes; DP-Tree movie type node  $\langle v, \varepsilon_v \rangle$  and its privacy are calculated as  $\varepsilon_v$  in the queue set  $TQueue$ ;  $P_v$  is the viewing statistical frequency of the current node  $v$ ; GetTop (LinkQueue Q, string r, float e) represents the queue function of queue header element.

---

Privacy budget allocation algorithm:

---

**Input:** Privacy budget  $\varepsilon$ , The prefix tree DP-Tree, The root node R

---

**Output:** Privacy budget allocation results set TMovie

---

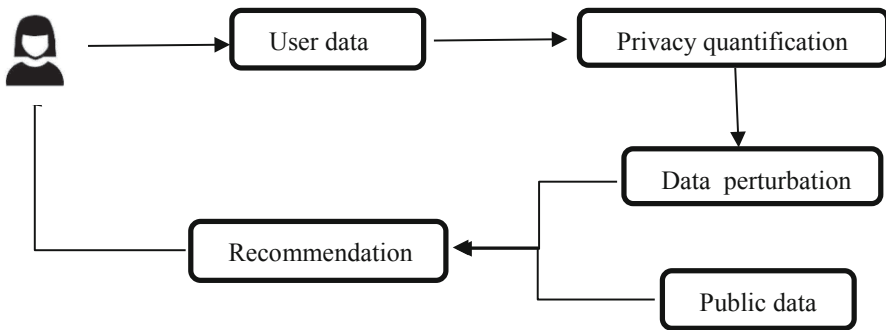
1. Initialize the TMovie and TQueue collections to 0
  2. IF (R==')
  3.  $\varepsilon_R=0$
  4.  $R \rightarrow \text{child}(R)$
  5. Else
  6. Add the current node  $\langle R, \varepsilon_R \rangle$  to the TQueue
  7. While TQueue  $\neq$  NULL Do
  8. GetTop (TQueue,R, $\varepsilon_R$ )
  9. IF  $R \in TMovie$  Then
  10.  $\varepsilon_R \leftarrow TMovie$
  11.  $TMovie \leftarrow \langle R, \varepsilon_R + \varepsilon_{P_R} \rangle$
  12. Else
  13.  $TMovie \leftarrow \langle R, \varepsilon_{P_R} \rangle$
  14. End If
  15. If ( $P_R == P_{R-parent}$ )
  16.  $\varepsilon \leftarrow \varepsilon/2$
  17. Else
  18.  $\varepsilon \leftarrow (\varepsilon - \varepsilon_{P_R})/2$
  19. For v
  20.  $P_v \leftarrow$  Probability of watching movie type v
  21. Add  $\langle v, \varepsilon_{P_v} \rangle$  to TQueue
  22. EndFor
  23. End while
-

In the above algorithm, the  $TMovie$  and  $TQueue$  sets are initialized to empty after the input of the privacy budget  $\epsilon$ , the prefixed prefix tree structure and the root node  $R$  of the prefix. Will the current node and the current node privacy are added to the  $TQueue$  budget ( $R$  for the root node), and then determine whether the current node and its parent weights are the same, if is the same accounts for half of the current budget privacy, if not the same with his brother calculate the current node weight ratio, such as half of the parent node privacy budget proportion distribution of the remaining half of privacy. Then loop the children of the current node.

## 4 Detailed Design of Film Recommendation System Based on Differential Privacy Protection

### 4.1 The Overall Framework of Film Recommendation System Based on Differential Privacy Protection

Figure 3 is the overall frame diagram of the movie recommendation system based on differential privacy protection. The system is composed of five components, in the first place in the user's local private equipment end users' private data collection, based on the user's personal data to construct privacy prefix tree and dynamically in accordance with this privacy budget allocations, and add meet the Laplace distribution noise, after using interference user data and public data together after a recommendation algorithm calculation item want to recommend to users. The meaning of each component in the figure is as follows:



**Fig. 3.** Frame diagram of movie recommendation system based on differential privacy protection

**Public data:** to obtain Public information related to users' private data from internal or external resources. This article Users the MovieLens 1M data set, which contains 100 million ratings from 6,000 users on nearly 4,000 movies. This data set will be used as experimental data set and test data set for experimental verification in this paper.

**User data:** the User data is the User history data collected on the User's private device. This paper obtains the historical information record of the User's watching movies, such as the frequency of watching a certain type of movie and the User's rating information on the movie. This information is not subject to interference.

Privacy quantification: the recommendation system algorithm based on differential Privacy proposed in this paper builds the Privacy prefix tree according to the user’s behavior record, and dynamically allocates the Privacy budget according to the frequency probability of each node in the Privacy prefix tree.

Data perturbation: according to the privacy prefix tree each node in the distribution of the privacy of our budget to each node adding meet the noise of the Laplace distribution, disturbance of the original Data, the Data is available to add a moderate amount of noise to protect the user’s personal Data privacy. In addition, the disturbed data needs to meet two objectives: privacy protection security and recommended accuracy.

Recommendation: an untrusted third-party server obtains user information data after adding noise to build a user-movie type interest matrix, performs matrix similarity calculation based on multiple dimensions to find out similar user groups, and user-based collaborative filtering algorithm to recommend the desired information to users.

### 4.2 Privacy Security Analysis

The following is the security analysis of differential privacy protection based on the DP-MRE algorithm proposed in this paper. Let  $D_1, D_2$  be the adjacent data set (that is,  $d(D_1, D_2) = 1$ ),  $f(D_i)$  be the category set of user private data,  $C$  is the size of the public movie set,  $j$  is the user private movie data, and  $z(j)$  is the size of the Laplace noise added to the movie type  $j$ . For any  $r = (r_1, \dots, r_c) \in \text{Range}(\text{DP} - \text{MRE})$ , on the basis of the definition of differential privacy can know, if the algorithm DP - MRE content:

$$\frac{\Pr[\text{DP} - \text{MRE}(D_1) = r]}{\Pr[\text{DP} - \text{MRE}(D_2) = r]} \leq e^\epsilon$$

The algorithm DP-MRE satisfies the constant  $\epsilon$ -differential privacy protection.

According to the differential privacy protection proposed in this paper, the differential privacy protection is carried out on the user’s local private device, so the privacy protection analysis only focuses on the steps of privacy budget allocation and noise addition, while there is no privacy leakage problem in the user similarity calculation and recommended steps. Therefore, privacy security analysis can be performed in the privacy budget allocation and noise addition steps. The analysis is as follows:

$$\begin{aligned} \frac{\Pr[\text{DP} - \text{MRE}(D_1) = r]}{\Pr[\text{DP} - \text{MRE}(D_2) = r]} &= \prod_{j \in C} \frac{\Pr[\text{DP} - \text{MRE}(D_1)(j) = r(j)]}{\Pr[\text{DP} - \text{MRE}(D_2)(j) = r(j)]} \\ &\geq \exp\left(-\sum_{j \in C} \frac{1}{z(j)} |f_j(D_1) - f_j(D_2)|\right) \\ &\geq \exp\left(-\max_{d(D_1, D_2)=1} \sum_{j \in C} \frac{1}{z(j)} |f_j(D_1) - f_j(D_2)|\right) \geq e^{-\epsilon} \end{aligned}$$

So the DP-MRE satisfies:

$$\frac{\Pr[\text{DP} - \text{MRE}(D_1) = r]}{\Pr[\text{DP} - \text{MRE}(D_2) = r]} \leq e^\epsilon$$

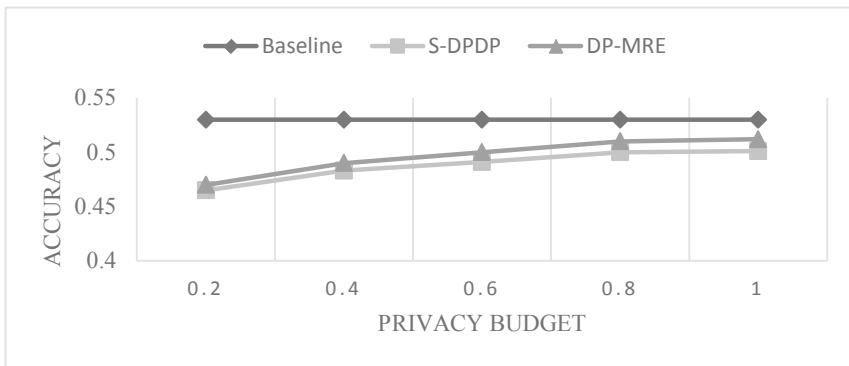
In the first step above, the independent injected noise on each category set is obtained from the combinational property of difference privacy and remains unchanged. In the second step, the injected Laplace noise and triangle inequality can be derived, and the above proof is completed.

## 5 Experimental Results and Analysis

In order to reflect the impact of differential privacy on the recommendation quality of the recommendation system (DP-MRE) in this paper, the precision rate and recall rate are used to evaluate the recommendation system model in this paper.

### 5.1 Influence of Accuracy

In order to objectively analyze the feasibility and effect of DP-MRE algorithm based on differential privacy protection proposed in the film recommendation system, this method is compared with S-DPDP algorithm based on differential privacy protection proposed by Shen et al. We set the difference privacy parameter as an independent variable, took different values for the privacy budget parameter in the experiment, and controlled a single variable to compare multiple recommendation algorithms. In addition, in order to more intuitively reflect the impact of privacy protection on the overall recommendation algorithm, this paper added the data recommendation algorithm Baseline without privacy protection to the experiment and compared it with it. Now, S-DPDP and DP-MRE data with privacy protection are compared with the data algorithm without privacy protection.



**Fig. 4.** Impact of difference privacy on the accuracy of the recommendation system

Figure 4 shows the impact of differential privacy protection on the accuracy of the recommendation system. By the experimental results show that the Fig. 4 in not in privacy protection recommendation system, the accuracy of the collaborative filtering recommendation system based on user recommendation about 0.53 or so, and privacy protection based on the difference of DP-MRE and S-DPDP will cause a certain degree of recommendation quality loss, when the differential privacy parameter epsilon close

to 1, DP-MRE and S-DPDP algorithm recommended accuracy of about 0.51. With the increase of the parameter of the privacy algorithm, the accuracy of DP-MRE and S-DPDP algorithms gradually increases to the recommended level of the original data. Compared with S-DPDP, DP-MRE has a smaller loss of precision, because DP-MRE is a privacy budget allocated based on DP-Tree structure, which maintains the type combination sequence and frequency characteristics of the movies watched by users and distributes the Laplace noise reasonably, thus reducing the loss of recommended quality caused by noise addition. However, S-DPDP user an iterative algorithm to add noise, which blurs the similarity between users. Therefore, in the aspect of recommendation quality loss, DP-MRE is better than S-DPDP algorithm, but DP-MRE has a high time complexity in the privacy budget allocation process, which affects the overall system efficiency.

## 5.2 Influence of Recall Rate

Figure 5 for the influence of difference of privacy to recall rate, by the experimental results show that the Fig. 5 in not in privacy protection recommendation system, collaborative filtering recommendation system based on users recommend the recall rate of around 0.51 or so, DP-MRE based on differential privacy and S-DPDP will cause a certain degree of recommendation quality loss, but with the increase of difference algorithm parameter epsilon privacy, the recall rate gradual in the recommended level with the original data. In the recommendation results, the higher the precision rate and recall rate, the higher the quality of the recommendation system. According to the experimental results, the recall rate of DP-MRE is very similar to that of S-DPDP, which means that when the data set base is very large, the recall rate of the two recommendation algorithms is basically similar, but the recall rate of DP-MRE is still slightly higher than that of S-DPDP algorithm.

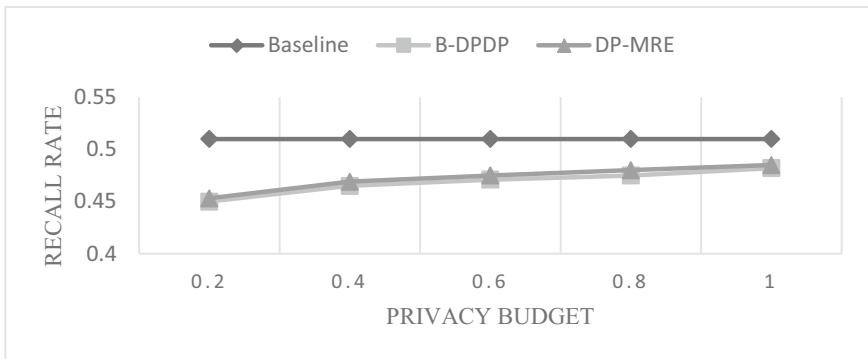


Fig. 5. Impact of difference privacy on recall rate

## 6 Conclusion

This paper mainly introduces how to apply the differential privacy protection technology to the movie recommendation system to solve the user privacy protection problem in the



recommendation process and ensure the recommendation performance will not suffer too much loss. In conclusion, this paper first dynamically adds noise to the sensitive information of users locally to ensure the privacy and security of users, then sends the user data with noise to the server for similarity calculation, and recommends the movie to users according to the user-based collaborative filtering algorithm. In this way, users' privacy data will not be violated during the whole recommendation process. The performance and effect of the recommendation system are verified by experiments. Although we have done some research work in the field of differential privacy and recommendation system, there are still many fields to be further studied in the application of differential privacy, and many security problems in the recommendation system have not been solved. Therefore, the relevant research still has a long way to go.

## References

1. Chaudhuri, K., Sarwate, A., Sinha, K.: Near-optimal differentially private principal components. In: NIPS, pp. 989–997 (2012)
2. Chaudhuri, K., Vinterbo, S.A.: A stability-based validation procedure for differentially private machine learning. In: NIPS, pp. 2652–2660 (2013)
3. Fung, B.C.M., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: a survey of recent developments. *ACM Comput. Surv.* **42**(4), 14:1–14:53 (2010)
4. Thakurta, A.G., Smith, A.: (Nearly) optimal algorithms for private online learning in full-information and bandit settings. In: NIPS, pp. 2733–2741 (2013)
5. Hardt, M., Ligett, K., Mcsherry, F.: A simple and practical algorithm for differentially private data release. In: NIPS, pp. 2339–2347 (2012)
6. McSherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the net. In: KDD, pp. 627–636 (2009)
7. Ye, Q., Meng, X., Zhu, M., et al.: A review of localized differential privacy. *J. Softw.* **29**(07), 159–183 (2018)
8. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: privacy via distributed noise generation. In: Vaudenay, S. (ed.) *EUROCRYPT 2006*. LNCS, vol. 4004, pp. 486–503. Springer, Heidelberg (2006). [https://doi.org/10.1007/11761679\\_29](https://doi.org/10.1007/11761679_29)
9. 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](https://doi.org/10.1007/11681878_14)
10. Dwork, C., Mcsherry, F., Talwar, K.: The price of privacy and the limits of LP decoding. *ACM Symposium on Theory of Computing*. ACM (2007)
11. Vágvölgýi, S.: Descendants of a recognizable tree language for prefix constrained linear monadic term rewriting with position cutting strategy. *Theor. Comput. Sci.* **732**, 60–72 (2018)
12. Hay, M., Rastogi, V., Miklau, G., et al.: Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow.* **3**(1/2), 1021–1032 (2009)
13. Chen, R., Fung, B.C.M., Desai, B.C.: Differentially private transit data publication: a case study on the Montreal transportation system. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 213–221. ACM, New York (2012)
14. Shang, T., Zhao, Z., Shu, W., et al.: Algorithm of big data decision tree based on isometric privacy budget allocation. *Eng. Sci. Technol.* **51**(02), 134–140 (2019)
15. Wang, X., Han, H., Zhang, Z., Yu, Q., Zheng, X.: Budget allocation method for tree index data differential privacy. *Comput. Appl.* **38**(07), 1960–1966 (2008)
16. Demin, H., Liao, Z.: Differential privacy location privacy protection method for m-fork average tree. *J. Small Micro Comput. Syst.* **40**(03), 76–82 (2019)



# Reliability Analysis of Heterogeneous CPS Under Different Swapping Inter-links Strategies

Hao Peng<sup>1,2</sup>, Can Liu<sup>1</sup>, Dandan Zhao<sup>1(✉)</sup>, Zhaolong Hu<sup>1</sup>, Jianmin Han<sup>1</sup>,  
and Jianfeng Lu<sup>1</sup>

<sup>1</sup> College of Mathematics and Computer Science, Zhejiang Normal University,  
Jinhua 321004, Zhejiang, China  
ddzhao@zjnu.edu.cn

<sup>2</sup> Shanghai Key Laboratory of Integrated Administration Technologies  
for Information Security, Shanghai 200240, China

**Abstract.** In this paper, we study the robustness of the Cyber-Physical System (CPS), which consists of interdependent physical resources and computational resources. Numerous infrastructure systems can evolve into the Cyber-Physical System. These networks depend on their interdependent networks, which provide information or energy to function. In a CPS architecture, a small failure could trigger serious cascading failures within the entire interdependent networks. Different systems' structures would influence the performance of swapping inter links strategies on improving the reliability of networks. We observe that reducing the influence of cascading failures is independent from the attack strategies and can be optimized by adjusting a Cyber-Physical System topology.

**Keywords:** Cyber-Physical System · Interdependent networks · Cascading failures · Swap inter-links strategy · Robustness · Giant component

## 1 Introduction

Cyber-Physical Systems(CPS) is designed to integrate computing components, networks, and physical devices into well-defined environments for a specific purpose [1, 10, 11, 28]. With the rapid development and deep studying of Cyber-Physical Systems, we think that CPS typically consists of physical elements, a communication network, and a computation and control unit [22, 29]. The communication network can exchange data with other systems. The control unit is necessary to interact with the real-world and process the data obtained [15, 17]. In this way, increasing infrastructure systems evolve into CPS. For example, the smart grid system [19, 21, 28, 30] is regarded as the typical representative of CPS.

As the characteristic of the cascading failure, a failure may trigger the entire interdependent networks collapsing [2]. A small failure in infrastructure systems

may cause property damage or even loss of human life. Therefore, it is important to improve the reliability of infrastructure systems. It is especially important to reduce the probability of large-scale failure of a CPS happening. Understanding how to improve the ability of the CPS to resist cascading failure is a major challenge, which is vital for understanding the resilience of natural systems [24].

### 1.1 Improving Direction

To enhance the reliability of CPS, many scholars have explored from the hardware direction [7, 8, 12, 23, 27]. Pennekamp et al. [21] propose that keeping the safety of the new dataflows which stream in CPS could make the CPS work stably. Zhang et al. [31] investigate embodiment multi-state channel symbol energy and the influence of the CPS's security threshold.

Some research about CPS reliability is based on software [12, 25]. They ensure the system in a safe state through software. With the rise of machine learning, many scholars apply machine learning to enhance the reliability of CPS [7, 8].

In addition to changing hardware and software to improve network reliability, some scholars have also promoted CPS reliability research from the direction of evaluating CPS security [3].

Another direction to study the reliability of CPS is to abstract the CPS system into interdependent networks. Interdependent networks' nodes will be treated as different devices with similar functions. In interdependent networks, we pay more attention to the topological relationship of nodes. One interdependent networks is always considered an unweighted and undirected graph.

### 1.2 Organization

The outline of this work is as follows: In Sect. 2, we propose our functional model for CPS. Section 3 performs the processes of different swapping strategies in detail. Section 4 is the results of the simulation and analysis points. Conclusions and summarized in Sect. 5.

## 2 The Model

In Sect. 2, we review different correspondence relationships of the interdependent networks and the processes of the cascading failure. To slightly describe the cascading failure in interdependent networks, we build a simple model to perform cascading processes in detail.

### 2.1 Interdependent Networks Model

'One-to-one correspondence' model is suggested by Buldyrev to represent the cascading failure between networks [2]. The couple two networks that name  $A$  and  $B$  in the correspondence model. Based on the 'one-to-one correspondence' relationship between networks, each node in network  $A$  has an inter-link with

one node from network  $B$ , and vice versa. For this reason, the number of nodes in two networks that establish the ‘one-to-one correspondence’ model is the same.

Without the ‘one-to-one correspondence’ model, Shao et al. [6, 13, 26] build a ‘multiple-to-multiple correspondence’ model to imitate the real-world networks. This correspondence relationship implies that a node in network  $A$  operates depending on more than one node in network  $B$ , and vice versa. ‘Multiple-to-multiple correspondence’ can better display some characteristics of the realistic networks.

‘One-to-multiple correspondence’ model [5, 6, 9, 14] is different to ‘one-to-one correspondence’ and ‘multiple-to-multiple correspondence’ model. It combines some features of the above two models. Firstly, it increases the singularity of the ‘one-to-one correspondence’ model inter-links connection. Then, it improves the over-complexity of ‘multiple-to multiple correspondence’ model. ‘One-to-multiple correspondence’ model can well simulate the connection of equipment in the power grid. One power station can provide power for multiple devices but one control device only controls one power station. According to this correspondence relationship, ‘one-to-multiple correspondence’ model is widely used in power grid model simulation.

## 2.2 Cascading Failure Model

Buldyrev et al. study the robustness of the ‘one-to-one correspondence’ model and they put forward two conditions that must be met at the same time when one node in interdependent networks can work normally [2]:

- I. The node must belong to the giant component of its own network;
- II. The node must have at least one inter-link from other networks.

They derived the theoretical formula of final nodes number after cascade failure and verified its correctness through experimental simulation:

$$\begin{cases} x = g_A(y)p \\ y = g_B(x)p \end{cases} \quad (1)$$

where  $g_A(y)(g_B(x))$  means the fraction of nodes that belong to the giant component of network  $A(B)$ .  $p$  is the remaining fraction of nodes in initial attack. They give the derivation formula of the critical value  $p_c$ :

$$1 = p^2 \frac{dg_A}{dx} [pg_B(x)] \frac{dg_B}{dx} (x) \Big|_{x=x_c, x=p_c} \quad (2)$$

The Eq. 1 in ER networks will transform to:

$$\begin{cases} x = p[1 - f_A] \\ y = p[1 - f_B] \end{cases} \quad (3)$$

where

$$\begin{cases} f_A = \exp[ay(f_A - 1)] \\ f_B = \exp[bx(f_B - 1)] \end{cases} \quad (4)$$

In SF networks, the Eq. 1 is changing into:

$$x = p\langle k_A \rangle [p\tilde{\kappa}_A \langle k_B \rangle (\tilde{\kappa}_B x)^{1/(3-\lambda_B)}]^{1/(3-\lambda_A)} \quad (5)$$

$\tilde{\kappa}_A(\tilde{\kappa}_B)$  is the number of normal working nodes in network  $A(B)$  after the first stage in cascading failure.  $\langle k_A \rangle(\langle k_B \rangle)$  is the average degree value of network  $A(B)$ .  $\lambda_A(\lambda_B)$  is the parameter of SF network  $A(B)$ .

These conditions have been extensively studied and used in network theory. In this study, we make use of the above conclusions to measure the number of normal working nodes. The cascading failures are triggered by insufficient failed nodes in either network  $A$  and  $B$ . The proportion of the initial failed nodes is usually denoted by  $1 - p$ . The nodes which are not following the above two conditions will be removed with their links.

### 3 The Method

In this section, we introduce 6 swapping inter-links strategies which we apply to model in detail.

#### 3.1 Strategy 1: Low Degree (LD)

Degree centrality is one of the most important and simplest metrics to reflect the importance of one node locality in the network [2, 16]. LD swapping strategy is calculating all nodes' degree and ranking nodes in an increasing order after building an interdependent networks model. An inter-link is swapped between two nodes which have the lowest degree centrality values in their own network. We must ensure that the total number of inter-links and the number of each nodes' inter-links in the entire model remain unchanged. For example, node  $B_2$  has two inter-links with network  $A$  and the total number of inter-links are 10 in Fig. ??(a). We must maintain two inter-links with node  $B_2$  and the number of inter-links in the entire system is ten after swapping processing. The swapping operation is repeated until demanding number of nodes' inter-links are swapped.

#### 3.2 Strategy 2: High Degree (HD)

High degree (HD) swapping strategy describes as following: calculating all nodes by degree values and ranking nodes in decreasing order. An inter-link is swapped between two nodes which have the highest degree centrality values in own network. We ensure that the total number of inter-links and the number of each nodes' inter-links in the entire model remain unchanged. The swapping operation is repeated a demanded number of times.

### 3.3 Strategy 3: Low Betweenness (LB)

Betweenness centrality is a metric to evaluate nodes' importance by paths [16, 18]. The betweenness centrality values of nodes can be calculated by the following equation:

$$B(v) = \sum_{i \neq j} \frac{\sigma_{ij}(v)}{\sigma_{ij}} \quad (6)$$

where  $\sigma_{ij}$  is the number of the shortest paths going from node  $i$  to node  $j$  and  $\sigma_{ij}(v)$  is the number of shortest paths going from node  $i$  to node  $j$  through node  $v$  [16, 20].

LB swapping strategy is as following: calculating all nodes betweenness centrality values and ranking nodes in an increasing order. An inter-link is swapped between two nodes which have the lowest betweenness centrality values. We have to ensure that the total number of inter-links and the number of each nodes' inter-links in the entire model remain unchanged. The swapping operation is repeated until the specified number of nodes' inter-links is swapped.

### 3.4 Strategy 4: High Betweenness (HB)

HB swapping strategy is as following: calculating all nodes betweenness centrality values and ranking nodes in a descending order. An inter-link is swapped between two nodes which have the highest betweenness centrality values. We have to ensure that the total number of inter-links and the number of each nodes' inter-links in the entire model remain unchanged. The swapping operation is repeated until a demanded number of nodes' inter-links are swapped.

### 3.5 Strategy 5: Low Eigenvector Centrality (LEC)

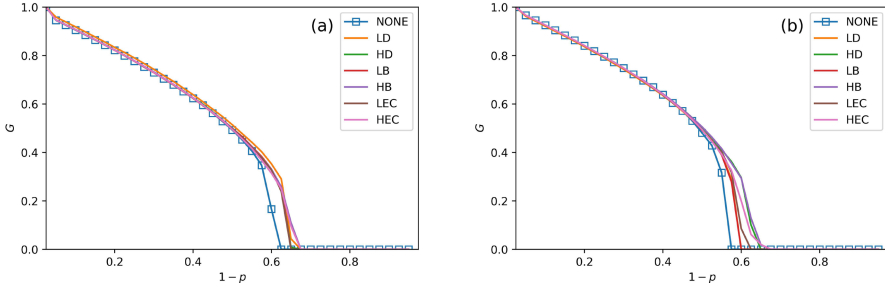
The eigenvector centrality is a metric of nodes' importance and it is an extension of degree centrality [20]. The eigenvector centrality fully considers both of the importance of nodes' neighbors and the number of the neighbors. If one node's neighbor is important, the node will be considered important, too.

To calculate all nodes' eigenvector centrality values, we need to construct an all nodes' adjacency matrix  $A$  and  $A_{ij}$  is an element of this matrix.  $x_i$  means the eigenvector centrality value of node  $i$  and the initial value of  $x_i$  is setting to 1. Then we use the initial  $x_i$  to calculate a new value of  $x'_i$ , which we define to be the sum of the eigenvector centrality values of node  $i$ 's neighbors: [20]:

$$x'_i = \kappa_1^{-1} \sum_j A_{ij} x_j \quad (7)$$

where  $\kappa_1$  is the largest eigenvector value of  $A$ .

LEC swapping strategy is as following: calculating all nodes eigenvector centrality values and ranking nodes in an increasing order. An inter-link is swapped between two nodes which have the lowest eigenvector centrality values. We have to ensure that the total number of inter-links and the number of each nodes' inter-links in the entire model remain unchanged. The swapping operation is repeated until the demanded number of nodes' inter-links are swapped.



**Fig. 1.** The fraction of function nodes in systems when  $f_N = 30\%$  in ER-ER and ER-SF system which is shown in Fig(a) and (b), respectively. Six swapping strategies are compared with original independent networks (NONE) in different system structures. In two subfigures, NONE yields the worst performance in increasing the values of  $G$  and  $p_c$ . HB is the best effect in enhancing  $G$  and HEC shows the best performance in improving  $p_c$  values. The values of  $p_c$  in these two figures under HEC strategy are 0.66 and 0.69. The curves under swapping interlinks with high centrality values are more smooth than NONE. The swapping inter-links with high centrality values can get better  $G$  values than other strategies.

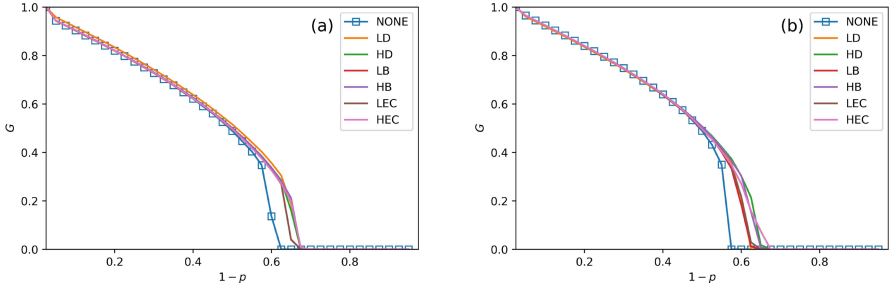
### 3.6 Strategy 6: High Eigenvector Centrality (HEC)

HEC swapping strategy is as following: calculating all nodes eigenvector centrality values and ranking nodes in a descending order. An inter-link is swapped between two nodes which hold the highest eigenvector centrality values. We make sure that the total number of inter-links in the entire model and the number of each of nodes' inter-links remains unchanged. The swapping operation is repeated until the specified number of nodes' inter-links are swapped.

## 4 Simulation Results and Analysis

In Sect. 4, we simulate the systems and the cascading failure to obtain the best strategy to enhance interdependent networks reliability. In [4], scholars have studied system reliability in 'one-to-one correspondence' and 'one-to-multiple correspondence' under HB strategy. Their models are built by BA networks. More swapping strategies are simulated in 'one-to-one correspondence' system. To study how six swapping strategies affect the robustness of 'one-to-multiple correspondence' systems, we conduct the following simulations.

To get more universal conclusions, we build four kinds of system models. These models are constructed by Erdős-Rényi network (ER network) and scale-free network (SF network). The average degree is  $\langle k \rangle = 4$  on all networks which we build. The parameter  $\lambda = 3$  in the SF network. 'One-to-multiple correspondence' is the dependence relationship of two networks in our simulation models. We set  $N_A$  and  $N_B$  are 15000 and 5000 and the connection ratio of inter-links is 3:1.



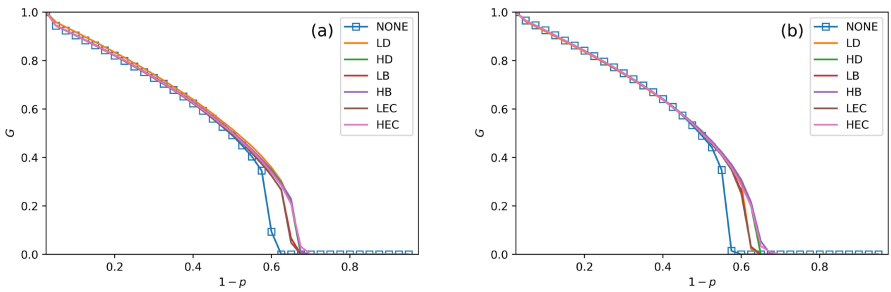
**Fig. 2.** The fraction of function nodes in systems when  $f_N = 50\%$  in ER-ER and ER-SF systems which are shown in Fig(a) and (b), respectively. Six swapping strategies are compared with original independent networks (NONE) in different system structures. In two subfigures, NONE yields the worst performance in increasing the values of  $G$  and  $p_c$ . HB is the best effect in enhancing  $G$  and HEC shows the best performance in improving  $p_c$  values. The values of  $p_c$  in these four figures under HEC strategy are 0.68 and 0.69. The curves with swapping interlinks with high centrality values are more smooth than NONE. The swapping inter-links under high centrality values can get better  $G$  values than other strategies.

### 4.1 System Robustness Metrics

Due to constraints such as economic conditions and operational complexity, we define the fraction of swapping inter-links between nodes  $f_N$  as:

$$f_N = \frac{N'_S}{N_A} \tag{8}$$

where  $N'_S$  means the number of inter-links which are swapped by strategies.



**Fig. 3.** The fraction of function nodes in systems when  $f_N = 70\%$ . Fig(a) and (b) are the systems that are combined with ER-ER and ER-SF, respectively. Six swapping strategies are compared with original independent networks (NONE) in different system structures. In two figures, NONE yields the worst performance. HEC shows the best performance in improving  $p_c$  and HB is the best effect in enhancing  $G$ . The values of  $p_c$  in these two figures under HEC strategy are 0.68 and 0.7. The curves under swapping interlinks with high centrality values are more smooth than others.



We implement a random removal of the network  $A$  node with the ratio  $(1 - p)$  as the failed node under random attacks. To reduce the error of the experimental results, we simulate 20 times for each  $1 - p$  under one certain swapping strategy and  $f_N$ . We take the average of these results as the final simulation results. We take  $G$  which means the proportion of nodes in the giant component to assess the reliability of the entire system and  $G$  can be calculated by:

$$G = \frac{N'_A + N'_B}{N_A + N_B} \quad (9)$$

where  $N'_A(N'_B)$  is the number of normal nodes at steady state. To measure maximum tolerant ability against random failure, we observe the values of  $p_c$ .

## 4.2 Impact of Network Size

We conduct performance comparisons among the six swapping strategies discussed in Sect. 3. The values of  $f_N$  in Fig. 1, Fig. 2 and Fig. 3 are 30%, 50% and 70%. In all figures, we plot the relationship between  $G$ ,  $p_c$  and  $1 - p$  under no swapping operation (NONE) as a contrast experiment for other strategies. From Fig. 1, Fig. 2 and Fig. 3, we can observe the following situations and conclusions:

- I. All swapping strategies perform better than NONE in improving  $G$ . The values of  $G$  are clearly bigger in swapping strategies than NONE when  $1 - p$  increases. For example, the values of  $G$  in NONE are lower than the other strategies when  $(1 - p) > 0.58$  in Fig. 1(a). When  $(1 - p) > 0.5$ , the values of  $G$  in NONE is lower than other strategies in Fig. 2(b). This situation is reflected in both two subfigures in Fig. 1, Fig. 2, Fig. 3.
- II. In Fig. 1, Fig. 2 and Fig. 3, all curves can be divided into 3 categories. The first is NONE which shows the worst performance in improving system reliability. The second is swapping inter-links by low centrality values which are LD, LB and LEC strategy. Although they show better performer than NONE in enhancing  $G$ , they are not the best choices to achieve more robustness systems. The last category is swapping inter-links with high centrality values. High centrality swapping strategies increases the values of  $G$ . We should adopt high centrality values swapping strategy in improving system reliability. This finding is the same conclusion as [4].
- III. From all subfigures we plot in Fig. 1, Fig. 2 and Fig. 3, we conclude that HB swapping strategy can be the first choice in improving  $G$ . This finding is different from [4]. We reveal that the networks constructive plays a vital role in system reliability.

## 4.3 Compare Network's $p_c$

From the above figures, all swapping strategies perform better than NONE in enhancing the value of network's  $p_c$ . The values of  $p_c$  are increasing with  $f_N$  increases. For instance, the values of  $p_c$  in Fig. 1(a), Fig. 2(a) and Fig. 3(a) is getting closer and closer to 0.7.

In all strategies, we can find that swapping inter-links by high centrality swapping strategies could get better  $p_c$ . HEC is the first choice in improving  $p_c$ . This conclusion is more significant in SF-SF systems.

## 5 Conclusion and Future Work

To improve the reliability of interdependent networks, we build two kinds of heterogeneous interdependent network models with the coupled ratio at 3:1 to achieve this goal. Then, we swap inter-links in different models under different swapping rates. Finally, we measure the reliability of systems by calculating the fraction of the giant component  $G$  and maximum resistance to random attacks of  $p_c$  after cascading failure. We find that swapping inter-links by high centrality values can get better system robustness. The simulation results show that high betweenness centrality swapping inter-links strategy performs the best effect in enhancing  $G$ , and high eigenvector centrality is the best choice in improving  $p_c$ .

However, our proposed model still has some limitations, which we will work on in the future. We should build more complex interdependent systems to simulate realistic networks better. An example is using an unfixed coupled ratio, distributing inter-links in one certain regularity, and using other single network models. We are trying to reach different strategies to maximize the number of nodes in the giant component.

**Acknowledgments.** This work was supported in part by the National Natural Science Foundation of China (Grant No. 61902359, No. 61672467 and No. 61672468), in part by the Social Development Project of Zhejiang Provincial Public Technology Research (Grant No. 2016C33168), in part by Zhejiang Provincial Natural Science Foundation of China (Grant No. LQ19F030010), and in part by the Opening Project of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security (Grant No. AGK2018001).

## References

1. Arafsha, F., Laamarti, F., El Saddik, A.: Development of a wireless CPS for gait parameters measurement and analysis. In: 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 1–5. IEEE (2018)
2. Buldyrev, S.V., Parshani, R., Paul, G., Stanley, H.E., Havlin, S.: Catastrophic cascade of failures in interdependent networks. *Nature* **464**(7291), 1025–1028 (2010)
3. Castellanos, J.H., Zhou, J.: a modular hybrid learning approach for black-box security testing of CPS. In: Deng, R.H., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds.) ACNS 2019. LNCS, vol. 11464, pp. 196–216. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-21568-2\\_10](https://doi.org/10.1007/978-3-030-21568-2_10)
4. Chattopadhyay, S., Dai, H., Hosseinalipour, S., et al.: Designing optimal interlink patterns to maximize robustness of interdependent networks against cascading failures. *IEEE Trans. Commun.* **65**(9), 3847–3862 (2017)
5. Chen, L., Yue, D., Dou, C.: Optimization on vulnerability analysis and redundancy protection in interdependent networks. *Phys. A: Stat. Mech. Appl.* **523**, 1216–1226 (2019)

6. Chen, L., Yue, D., Dou, C., Cheng, Z., Chen, J.: Robustness of cyber-physical power systems in cascading failure: survival of interdependent clusters. *Int. J. Electr. Power Energy Syst.* **114**, 105374 (2020)
7. Cohen, S., Gluck, T., Elovici, Y., Shabtai, A.: Security analysis of radar systems. In: *Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy*, pp. 3–14 (2019)
8. Dibaji, S.M., Pirani, M., Flamholz, D.B., Annaswamy, A.M., Johansson, K.H., Chakraborty, A.: A systems and control perspective of CPS security. *Ann. Rev. Control* **47**, 394–411 (2019)
9. Dong, G., Chen, Y., Wang, F., Du, R., Tian, L., Stanley, H.E.: Robustness on inter-dependent networks with a multiple-to-multiple dependent relationship. *Chaos: Interdiscip. J. Nonlinear Sci.* **29**(7), 073107 (2019). Kindly provide the volume number for Ref. [10], if applicable
10. Gao, H., Duan, Y., Shao, L., Sun, X.: Transformation-based processing of typed resources for multimedia sources in the IoT environment. *Wirel. Netw.* 1–17 (2019)
11. Gao, H., Xu, Y., Yin, Y., Zhang, W., Li, R., Wang, X.: Context-aware QoS prediction with neural collaborative filtering for Internet-of-Things services. *IEEE Internet Things J.* **7**, 4532–4542 (2019)
12. Gardiner, J., Craggs, B., Green, B., Rashid, A.: Oops i did it again: further adventures in the land of ICS security testbeds. In: *Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy*, pp. 75–86 (2019)
13. Huang, Z., Wang, C., Ruj, S., Stojmenovic, M., Nayak, A.: Modeling cascading failures in smart power grid using interdependent complex networks and percolation theory. In: *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1023–1028. IEEE (2013)
14. Huang, Z., Wang, C., Stojmenovic, M., Nayak, A.: Characterization of cascading failures in interdependent cyber-physical systems. *IEEE Trans. Comput.* **64**(8), 2158–2168 (2014)
15. Jazdi, N.: Cyber physical systems in the context of industry 4.0. In: *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 1–4. IEEE (2014)
16. Ji, X., et al.: Improving interdependent networks robustness by adding connectivity links. *Phys. A: Stat. Mech. Appl.* **444**, 9–19 (2016)
17. Koc, H., Shaik, S.S., Madupu, P.P.: Reliability modeling and analysis for cyber physical systems. In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0448–0451. IEEE (2019)
18. Kumari, P., Singh, A.: Approximation and updation of betweenness centrality in dynamic complex networks. In: Verma, N.K., Ghosh, A.K. (eds.) *Computational Intelligence: Theories, Applications and Future Directions - Volume I. AISC*, vol. 798, pp. 25–37. Springer, Singapore (2019). [https://doi.org/10.1007/978-981-13-1132-1\\_3](https://doi.org/10.1007/978-981-13-1132-1_3)
19. Mihalache, S.F., Pricop, E., Fattahi, J.: Resilience enhancement of cyber-physical systems: a review. In: Mahdavi Tabatabaei, N., Najafi Ravadanegh, S., Bizon, N. (eds.) *Power Systems Resilience. PS*, pp. 269–287. Springer, Cham (2019). [https://doi.org/10.1007/978-3-319-94442-5\\_11](https://doi.org/10.1007/978-3-319-94442-5_11)
20. Newman, M.: *Networks*. Oxford University Press, Oxford (2018)
21. Pennekamp, J., et al.: Dataflow challenges in an internet of production: a security & privacy perspective. In: *Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy*, pp. 27–38 (2019)
22. Poovendran, R.: Cyber-physical systems: close encounters between two parallel worlds [point of view]. *Proc. IEEE* **98**(8), 1363–1366 (2010)

23. Prathiba, A., Bhaaskaran, V.K.: Hardware footprints of s-box in lightweight symmetric block ciphers for IoT and CPS information security systems. *Integration* **69**, 266–278 (2019)
24. Reis, S.D., et al.: Avoiding catastrophic failure in correlated networks of networks. *Nat. Phys.* **10**(10), 762 (2014)
25. Romagnoli, R., Krogh, B.H., Sinopoli, B.: Design of software rejuvenation for CPS security using invariant sets. In: 2019 American Control Conference (ACC), pp. 3740–3745. IEEE (2019)
26. Shao, J., Buldyrev, S.V., Havlin, S., Stanley, H.E.: Cascade of failures in coupled network systems with multiple support-dependence relations. *Phys. Rev. E* **83**(3), 036116 (2011)
27. Tippenhauer, N.O., Wool, A.: CPS-SPC 2019: fifth workshop on cyber-physical systems security and privacy. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp. 2695–2696 (2019)
28. Xuhong, L., Muhai, L.: Application of CPS in the complex network. In: 2011 International Conference on Intelligent Computation Technology and Automation (ICICTA), vol. 2, pp. 1067–1069. IEEE (2011)
29. Yin, Y., Chen, L., Xu, Y., Wan, J., Zhang, H., Mai, Z.: QoS prediction for service recommendation with deep feature learning in edge computing environment. *Mob. Netw. Appl.* **25**, 1–11 (2019)
30. Zhang, J., Yeh, E., Modiano, E.: Robustness of interdependent random geometric networks. *IEEE Trans. Netw. Sci. Eng.* **6**, 474–487 (2018)
31. Zhang, J., Yang, A., Hu, Q., Hancke, G.P.: Security implications of implementing multistate distance-bounding protocols. In: Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy, pp. 99–108 (2019)



# Adapt Swarm Path Planning for UAV Based on Artificial Potential Field with Birds Intelligence Extensions

Yifei He<sup>1</sup>, Jiqiang Liu<sup>1</sup>, Endong Tong<sup>1</sup> (✉), Wenjia Niu<sup>1</sup> (✉), Xinyu Huang<sup>1</sup>, Ying Zhou<sup>1</sup>, Chenyang Li<sup>1</sup>, and Liang Chang<sup>2</sup>

<sup>1</sup> Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, 3 Shangyuan Village, Haidian District, Beijing 100044, China

{edtong, niuwj}@bjtu.edu.cn

<sup>2</sup> Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

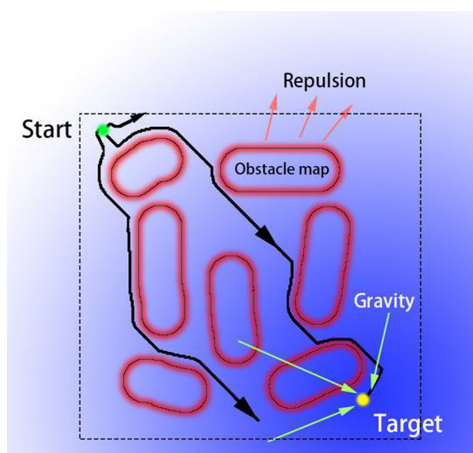
**Abstract.** Artificial potential field (APF), a concept from physic field, has been successfully adopted for path planning of unmanned aerial vehicle (UAV). The cooperation between the expulsion against obstacle and the gravity from target, ensures a global planning optimization considering obstacle avoidance. Unfortunately, under different UAV-to-UAV distance conditions, the APF also has a weakness unable to support swarm path planning for multiple UAVs due to the highly dynamic shift between the repulsion and the gravity. To utilize APF to realize robust swarm planning, we redesign the APF and embed it into the swarm avoidance mechanism from bird intelligence involving group collision avoidance (GCA) and individual collision avoidance (ICA), forming two kinds of a APF-based swarm planning respectively: GCA-APF and ICA-APF. We then propose an adapt switch mechanism for dynamically choosing GCA-APF or ICA-APF in contexts of different obstacle environment. Experiments show the effectiveness of our approach, 15.25% higher planning efficiency than that of original GCA and avoiding certain polynomial cost increase from original ICA.

**Keywords:** Path planning · Artificial potential field · Collision avoidance · Unmanned aerial vehicle · Swarm intelligence

## 1 Introduction

Path planning is highly important for unmanned aerial vehicle (UAV), involving drones and self-driving cars. Among different path planning methods in the field of drone, the artificial potential field (APF)-based method draws more interests and has shown obvious advantages to become one state of the art: fast calculation and nearly real-time response in a complex environment. Previous APF improvement work in 2018 [1] further contributes to solve the local optimal problem through increasing a tangent vector of drone in the direction of potential field, ensuring a global path planning on single drone in good performance.

With the development of drone applications, a drone cluster formed by a lot of drones, can be utilized to accomplish tasks better than single drone. Thus, swarm intelligence is introduced in this field: bird swarm [2, 3], ant colony [4] and bee colony [5, 6]. However, above swarm intelligences focus on collaboration of drones, while for path planning the mainstream APF-method is still not studied. One main reason is that the current APF is difficult to support swarm path planning for multiple drone swarm due to the highly dynamic shift between the repulsion and the gravity. As Fig. 1 shows, the target point attracts the drone, the obstacle repulses the drone, and there is also a repulsive relationship between the drones. In this occasion, only one drone can successfully reach the target point, and two others cannot accurately reach the target point due to the relatively high potential field around them.



**Fig. 1.** APF influence on swarm path planning

In this work, we aim to redesign a new APF suitable for swarm path planning. A natural idea is based on the swarm avoidance mechanism from bird intelligence proposed by Reynolds [7], which shows complex and orderly cluster behaviors in group flight, and have the characteristics of decentralized proximity interaction, internal stability of self-organized clusters, and adaptability to dynamic changes in the environment. Within bird swarm, there is no central node control and each individual (flying bird) is autonomous and can perceive the neighboring individuals within a certain range in the vicinity. Three basic collaboration rules are: 1) Separation-the members of the cluster should keep a certain safe distance from their neighbors to avoid congestion; 2) Alignment-the speed direction of the cluster members should be consistent with the average flight direction of neighboring units; 3) Gathering-the members of the cluster should fly towards the center of the neighboring unit.

Based on the three principles of Reynolds, we aim to improve APF by group collision avoidance (GCA) and individual collision avoidance (ICA) [8, 9]. The GCA and the ICA are fundamentally different: the first allows a UAV swarm to avoid collision as a cluster, while the second allows individual swarm members to achieve collision

avoidance by individual actions. We form two kinds of an APF-based swarm planning respectively: GCA-APF and ICA-APF. Compared with traditional GCA and ICA algorithms, GCA-APF and ICA-APF incorporate the idea of artificial potential field, which leads to our algorithms can dynamically plan the optimal moving path in a complex environment, instead of choosing a direction for obstacle avoidance without basis, thus its flight efficiency has been greatly improved.

Also, an adapt switch mechanism should be developed for dynamically choosing GCA-APF or ICA-APF in contexts of different obstacle environment. GCA is suitable for the situation where the obstacle is large, and the swarm prefers to maintain a complete formation while avoiding obstacles, and ICA is more suitable for the situation where the obstacle is relatively small, thus the cost of obstacle avoidance is lower than GCA. We summarize our contributions as follows:

- We are the first to modify APF suitable for swarm planning and we redesign the APF and embed it into the swarm avoidance mechanism from bird intelligence involving GCA and ICA, forming two kinds of an APF-based swarm planning respectively: GCA-APF and ICA-APF.
- We propose an adapt switch mechanism for dynamically choosing GCA-APF or ICA-APF in contexts of different obstacle environment, which makes it more flexible for implementation in real drone platforms.

## 2 Preliminary

In this section, we introduce the preliminary work about the UAV swarm and cluster path planning algorithm.

### 2.1 The Artificial Potential Fields (APF)

Artificial potential field (APF) is commonly used in the field of robot path planning. The most prominent advantage of APF algorithm is briefness. APF model consists of two kind of potential field, the gravitational and repulsion field which based on the obstacles and the goal respectively corporately attract robot complete path planning. The trajectory generated by APF is smooth in mathematic which that mean the tangent and the velocity is always continuous. Those two characteristics improved robot stability during the motion process. Attraction field,  $U_{Aat}(q_{now})$  is construct as the following equation,

$$U_{Aat}(q_{now}) = \frac{1}{2}k\rho^2(q_{now}, q_{goal}) \quad (1)$$

where  $k$  is the coefficient of attraction field,  $q_{now}$  is current position,  $q_{goal}$  is goal,  $\rho(q_{now}, q_{goal})$  is the distance between  $q_{now}$  and  $q_{goal}$ . We sign the attraction force produced by attraction field at  $q_{now}$  as  $F_{Aat}(q_{now})$ ,

$$F_{Aat}(q_{now}) = -\nabla U_{Aat}(q_{now}) = k\rho(q_{now}, q_{goal}) \quad (2)$$

Repulsion field is generated by all obstacles corporately, it is donated as  $U_{Ref}(q_{now})$ ,

$$U_{Ref}(q_{now}) = \sum_{i=1}^n U_{Ref}^i(q_{now}) \quad (3)$$

where  $U_{Ref}^i(q_{now})$  is repulsion field produced by the  $i$ -th obstacle at  $q_{now}$ ,

$$U_{Ref}^i(q_{now}) = \begin{cases} \frac{1}{2}\theta \left( \frac{1}{\rho(q_{now}, q_{obs}^i)} - \frac{1}{\rho_0} \right)^2, & \rho(q_{now}, q_{obs}^i) < \rho_0 \\ 0, & \rho(q_{now}, q_{obs}^i) \geq \rho_0 \end{cases} \quad (4)$$

Where  $\theta$  is the coefficient of repulsion field,  $\rho_0$  is the radius of repulsion field,  $\rho(q_{now}, q_{obs}^i)$  is the distance between  $q_{now}$  and the  $i$ -th obstacle, we sign the repulsion force produced by repulsion field at  $q_{now}$  as  $F_{Ref}(q_{now})$ ,

$$F_{Ref}(q_{now}) = -\nabla U_{Ref}(q_{now}) \quad (5)$$

The total force  $F_{Total}$  a robot received at  $q_{now}$  is calculated as follow.

$$F_{Total}(q_{now}) = F_{Aat}(q_{now}) + F_{Ref}(q_{now}) \quad (6)$$

## 2.2 GCA and ICA

In the GCA case, all drones are kept in a group while avoiding collisions. The weight of the cohesion rule  $w_c$  is increased so that the UAV will not deviate from the team when performing the avoidance action. The following formula can be used to implement GCA to calculate the weight of the avoidance rule for the  $i$ -th UAV.

$$W_{AVi} = \begin{cases} 1 & \text{if } (P_i - C_{p,i}) < \rho_{col} \\ 0 & \text{if } (P_i - C_{p,i}) > \rho_{col} \\ 0 & \text{alldaylog} \end{cases} \quad (7)$$

The desired avoidance heading angle of the  $i$ -th UAV is given as:

$$\psi_{AVi} = f_1\left(\frac{\pi}{2}\right) + \psi_i \quad (8)$$

Where the direction of turn ( $f_1$ ) required for collision avoidance in the horizontal plane is given by Fig. 2(a).

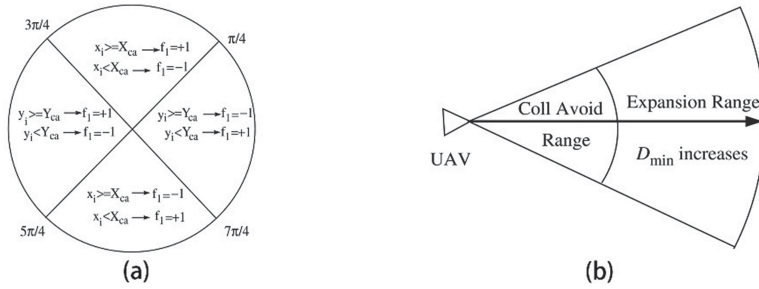
To achieve ICA, the algorithm used is similar to the GCA, but with different rule weights. Further, in this case two detection sensor ranges are defined. These are the expansion sensor range and the collision avoidance sensor range (Fig. 2(b)). ICA can be implemented using the following algorithm for each UAV. The minimum allowed distance between two UAVs ( $d_{min}$ ) in a group is set as,

$$d_{min} = \begin{cases} d_1 & \text{if } (P_i - C_{p,i}) \leq \rho_{exp} \\ d_2 & \text{if } (P_i - C_{p,i}) > \rho_{exp} \end{cases} \quad (9)$$

With  $d_1 > d_2$ . The avoidance rule weight, heading angle and direction of turn, avoidance pitch angle and direction is determined by (6), (7).

When alien UAVs are detected inside the expansion sensor range,  $d_{min}$  is increased from  $d_2$  to  $d_1$  so that the whole group expands enabling the alien UAV to pass through this group. Whereas, when the alien UAV comes within the collision avoidance range, a collision avoidance action is taken. In this case, in contrast to group-wise collision avoidance,  $w_c$  is decreased so that the expansion can take place easily.





**Fig. 2.** (a) Logic diagram to find the direction of turn in horizontal plane. (b) Two partition of sensor range for the ICA.

### 3 Adapt Swarm Planning

In this section, we discussed the design idea of GCA-APF and ICA-APF algorithm. And design a fusion algorithm to obtain the advantages of each model while making up for its deficiencies.

#### 3.1 GCA-APF Implementation

It can be seen from the second chapter that the GCA algorithm treats the entire cluster as a large rigid body, and the entire cluster remains compact in turning, hovering, and obstacle avoidance. In order to achieve this goal, there is a relatively large cohesion  $w_c$  within the cluster. During the operation of the entire cluster, there is a cluster centroid that can represent the operating state of the entire cluster, that is, all the aircraft in the cluster are roughly distributed around the cluster centroid. Therefore, we can let the center of mass of the cluster advance along the path planned by the artificial potential field, and let the entire cluster still wrap around the center of mass of the cluster. The distance between any UAV in the cluster and the centroid can be expressed as

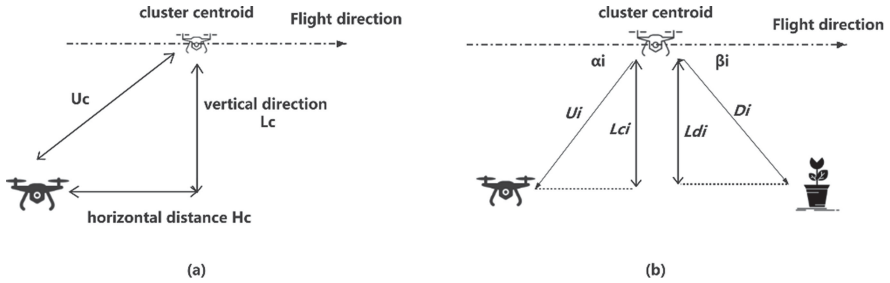
$$U_c^2 = H_c^2 + L_c^2 \tag{10}$$

Where  $U_c$  is the Euclidean distance between the UAV and cluster centroid,  $H_c$  represents the horizontal distance between the two, and  $L_c$  represents the distance between the two in the vertical direction. As we can see in Fig. 3(a), the vertical distance refers to the distance on the plane perpendicular to the current flight path, and the horizontal distance refers to the distance parallel to the current flight path. We set the angle of the connection between the centroid and the  $U_c$  of the current drone in the cluster as  $\alpha_i$ . During the forward flight of the drone cluster, as shown in Fig. 3(b), if the cluster detects an obstacle D in the direction  $\beta_i$  and the distance between them is  $D_i$ , thus the vertical distances  $U_{ci}$  and  $D_{ci}$  of the drone and obstacle can be calculated.

$$U_{ci} = \frac{U_i}{\sin\alpha_i} \tag{11}$$

$$D_{ci} = \frac{D_i}{\sin\beta_i} \tag{12}$$

Thus, if the  $D_{ci} > U_{ci}$ , At this time, the UAV in the current direction will not collide with obstacles, so it can fly normally without any changes. But if the  $D_{ci} \leq U_{ci}$ , It indicates that when the current UAV cluster moves to an obstacle, the  $U_{ci}$  will collide with the obstacle, making it necessary to implement the corresponding obstacle avoidance strategy.

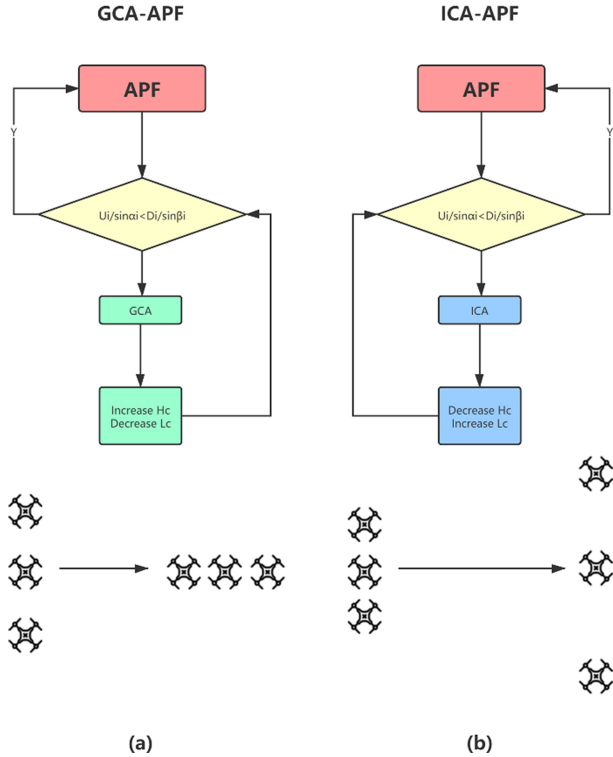


**Fig. 3.** (a) Schematic diagram of the relationship between drones in the cluster. (b) Relationship between drone and obstacle.

In the GCA-APF model, we design such an obstacle avoidance strategy. We need to increase the current drone  $H_c$ , in other word is that the position of the drone moves toward the back of the cluster, and reduce  $L_c$ , which makes the position of the drone close to the centroid. At the same time, according to the situation, calculate the location of the current  $U_{ci}$  and broadcast it out to ensure that the distance between all drones is within a safe threshold to avoid collisions. Therefore, in the actual process, we manually intervene in the position of the drone  $U_{ci}$  that is predicted to collide, and at the same time, let the cluster adaptively adjust the position between each drone according to the GCA algorithm.

Due to the flight path of the UAV is planned by the artificial potential field algorithm, and the entire cluster is calculated as one UAV in the planning process. Therefore, it can be ensured that the currently planned path can at least let one single drone pass through. In extreme cases, if the current path can only pass one drone at a time, the formation of the entire drone cluster will be aggregated into a drone queue with the centroid in the middle of the cluster to ensure the entire drone pass through obstacles smoothly. The flow of the entire algorithm is shown in the Fig. 4(a).

During the flight, if the cluster encounter narrow terrain, first the entire cluster will slow down and adjust the shape according to the above pattern to sequentially pass through the narrow space. In short, the cluster will be transformed into a suitable formation to avoid collision, but when encountering a small obstacle, the cluster will deflect as a rigid body to avoid the obstacle. Therefore, it can be seen that the GCA-APF algorithm has a relatively large obstacle avoidance cost, because its calculation and formation transformation require corresponding time to perform, so the entire cluster will slow down or even hover. Therefore, the GCA-APF algorithm works better when it encounters large fixed obstacles. If the obstacles are so small that they are smaller than the distance between the drones, then the drones need to be slightly pulled apart. Obstacle avoidance can be completed at intervals. However, the GCA-APF algorithm will still



**Fig. 4.** The respective processes of the two algorithms. (a) GCA-APF (b) ICA-APF

cause the entire UAV to turn in a larger direction to avoid obstacles, so it is not a perfect obstacle avoidance solution in this case. On the other hand, if the obstacle moves faster, the use of GCA-PAF for obstacle avoidance will also cause the entire cluster to fail to avoid obstacles quickly, making obstacle avoidance ineffective.

### 3.2 ICA-APF Implementation

On the other hand, we also designed the ICA-APF model in order to cope with more situations. Similar to the GCA-APF model, the ICA-APF model also allows the cluster centroid to advance along the path planned by the artificial potential field. When an obstacle is encountered, the ICA-APF model allows the drone cluster to be dispersed to avoid collision with the obstacle. When it is detected that the obstacle has been bypassed, the entire drone cluster will gather together and continue to move forward in a cluster.

When the UAV cluster encounters an obstacle, Similar to the GCA-APF, if  $U_i/\sin\alpha_i < D_i/\sin\beta_i$ , it will open the ICA obstacle avoidance mode, treat the other UAVs in the cluster as obstacles, and establish a new artificial potential field map. Because other drones are regarded as obstacles, the current drone will be separated from the cluster. By detecting the size of the current obstacle and dynamically adjusting the repulsion coefficient of the drones as obstacles, the drone and the drone can be changed.

The degree of dispersion of cluster centroids. When it detects that there is no obstacle in front, the system's ICA obstacle avoidance mode is turned off. At this time, the current UAV will not regard other UAVs as obstacles, so it gathers to the centroid due to the gravity of the artificial potential field. The flow of the entire algorithm is shown in the Fig. 4(b).

### 3.3 Architecture

It can be seen that the GCA-APF model has better obstacle avoidance effect when facing larger obstacles as well as in narrow terrain, because it can make the entire UAV cluster aggregate into a straight line, so that the cluster can better pass narrow obstacles. In this case, if the ICA-APF model is used for obstacle avoidance, the unmanned aerial vehicle needs to travel a longer distance to bypass large obstacles. If the obstacles are too large, it may cause some UAVs cannot reach the target point.

On the other hand, if it is a relatively small obstacle, using ICA-APF will get a better obstacle avoidance effect. Since GCA-APF always regards the UAV cluster as a whole, when a small obstacle is encountered, the UAV cluster will move in a direction away from the obstacle as a whole. In this case, if you use the ICA-APF method, you only need to move the drone in the direction of the obstacle away from the cluster. The other drones in the entire cluster can fly normally without affecting their status. Therefore, the ICA-APF model has lower obstacle avoidance cost and faster response speed. In addition, if the obstacle is moving, when the UAV sensor detects the obstacle, the ICA-APF model can be used to avoid obstacles faster, because other drones in the model do not need to slow down Adjust the formation, so its obstacle avoidance speed is much higher than GCA-APF.

We assume that there is currently an obstacle  $D$ , the diameter of which is  $D_0$ , and the minimum distance between the UAV and the obstacle is  $L_0$ . Assuming that there are  $n$  drones in the current cluster, use the GCA-APF and the ICA-APF to calculate the obstacle avoidance cost separately.

- GCA-APF algorithm:

$$\text{Cost}_{\text{GCA-APF}} = \sum_{i=1}^n \pi \cdot (D_0 + i \cdot L_0) \quad (13)$$

- ICA-APF algorithm:

$$\text{Cost}_{\text{ICA-APF}} = \sum_{i=1}^n \pi \cdot (D_0 + L_0) \quad (14)$$

It can be clearly seen that the cost of the GCA-APF is significantly higher than ICA-APF, and the greater the number of drones in the cluster, the faster the cost increases. The cost of ICA-APF obstacle avoidance is linearly related to the number of drones. The difference between these two algorithms is:

$$\text{Difference} = \sum_{i=1}^n \pi L_0 \cdot (i - 1) \quad (15)$$

### 3.4 Switch Mechanism

In our model, the UAV cluster should have three states, namely Normal-APF state, GCA-APF state and ICA-APF state. Among them, the Normal-APF state indicates that the current cluster environment is relatively stable, and there are no obstacles that need to be avoided around. At this time, the center of mass of the UAV cluster moves along the path planned by the artificial potential field. The UAVs in the cluster are evenly and orderly distributed around the center of mass; and the GCA-APF state indicates that the current cluster is in the GCA obstacle avoidance state. Faced with relatively large obstacles, the GCA algorithm was chosen to avoid obstacles. The ICA-APF state indicates that the current cluster is in the ICA obstacle avoidance state. At this time, the UAVs in the cluster are dispersed to avoid obstacles and re-assemble at the appropriate time.

How to define the size of obstacles to carry out the corresponding obstacle avoidance strategy is the core problem of our model. Here we use the threshold method for calculation. Assuming that in the current UAV cluster, the minimum distance between each UAV is  $D_{minU}$ , there are the following situations:

1. When the maximum diameter of the obstacle is less than  $D_{minU}$ , the UAV cluster can basically avoid the obstacle without increasing the distance between them. Therefore, the method of selecting ICA-APF works best at this time.
2. When the maximum diameter of the obstacle is greater than  $D_{minU}$ . Suppose that the distance that the UAV cluster can detect obstacles is  $D_{det}$ , and the UAV cluster begins to avoid obstacles when it detects the obstacles. At this time, the distance of obstacle avoidance is  $D_{obs}$ , and the speed of cluster is  $v_{clu}$ , thus the time which can be used for obstacle avoidance is given by:

$$t_{obs} = D_{obs}/v_{clu} \quad (16)$$

And the drone's lateral vertical speed refers to the maximum speed that the drone can reach in the vertical direction, we record as  $v_{ver}$ , then the maximum obstacle avoidance distance can be calculated, that is:

$$D_{maxobs} = t_{obs} \cdot v_{ver} = v_{ver} \cdot D_{obs}/v_{clu} \quad (17)$$

Therefore, if the maximum diameter of the obstacle detected by the UAV cluster is less than  $D_{maxobs}$ , we should use the ICA-APF model for obstacle avoidance. If the maximum distance detected by the UAV is greater than  $D_{maxobs}$ , we should use the GCA-APF model for obstacle avoidance. Thus, the  $D_{maxobs}$  is mainly determined by the UAV's own attributes and the positional relationship with obstacles, and can be calculated according to needs in practice.

## 4 Experiment

In this section, we verify the application of artificial potential field in the cluster through experiments, and verify the model we designed.

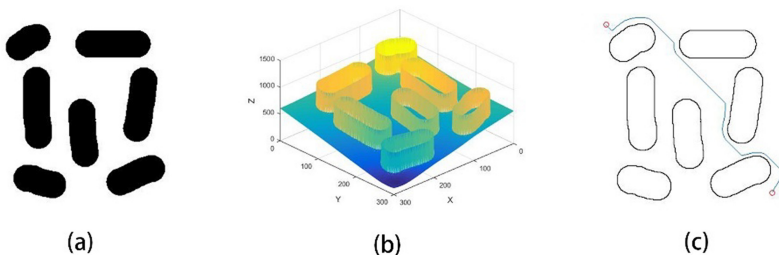
we chose the 2016 version of Matlab as the experimental platform, setting the linear velocity of the drone to 5 m/s and the angular velocity to 200 °/s. Observe the movement state of the UAV cluster in different modes and its potential field map through a map with random obstacles.

#### 4.1 Improved APF Algorithm in Clusters

We let the drone cluster use the improved artificial potential field algorithm based on particle swarm to calculate its planned path. In the experiment, the random obstacle we generated is shown in the Fig. 5(a), and the global potential field diagram is calculated according to the boundary of the obstacle. The reason why the calculation is based on the boundary of the obstacle instead of using all obstacles is because it can speed up the calculation without affecting the accuracy of the artificial potential field.

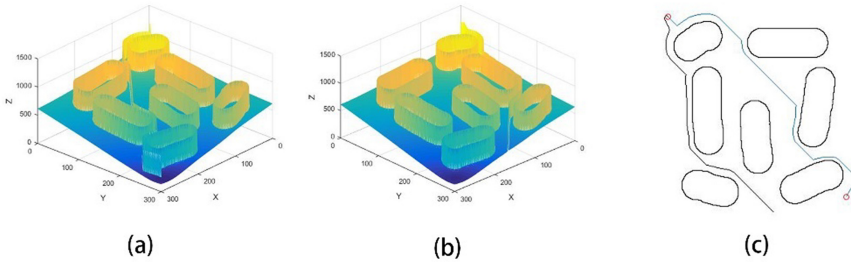
Firstly, a single drone is used for artificial potential field path planning. The artificial potential field diagram and path are shown separately in the Fig. 5(b) and (c). It can be seen that the UAV can choose the optimal path and reach the target point accurately. Therefore, the path planning for a single UAV is effective.

In Fig. 5(b), the x-axis and y-axis respectively represent the coordinates of points in space, while the z-axis represents the intensity of the potential field. It can be seen that the potential field at the boundary of the obstacle is the highest, which can give the drone a repelling force, and the potential field of the target point is the lowest, which can give the drone an attractive force. The drone moves towards the target point under the combined force. The specific path is shown in Fig. 5(c) and the two red circles represent the start and end points. On the other hand, it can be seen that the intensity of the potential field in the middle of the obstacle is very low. This is an optimization to speed up the calculation. Ignoring the influence of the repulsion on the drone inside the obstacle can speed up the path planning.

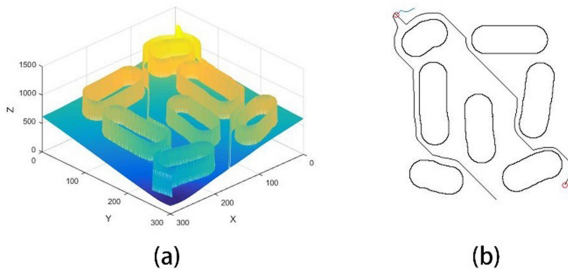


**Fig. 5.** Application of single drone in improved artificial potential field. (a) Schematic diagram of obstacles (b) Artificial potential field diagram (c) Route planning route diagram

Then we use two drones for path planning. In this process, both drones regard other drones as obstacles, so they will provide a repelling force to the current drones. The path planned by the entire system is shown in the Fig. 6(a). It can be seen that when there are two drones, due to the change of the potential field, the path taken by one of the drones is too far beyond the range of the map, so that the drone cluster cannot reach the target well.



**Fig. 6.** Application of two drones in improved artificial potential field. (a), (b) The potential field diagram of two drones. (c) Route planning route diagram.



**Fig. 7.** Application of three drones in improved artificial potential field. (a) Artificial potential field diagram (b) Route planning route diagram (Color figure online)

We third use three drones for the experiment, and the distribution of obstacles remained the same as previous. When three drones are included in the drone cluster, the potential field at each location detected by the third drone is too high, which prevents the drones from finding the optimal path. The potential field diagram detected by the third drone is shown in the Fig. 7(a), which can be seen that there is no suitable path to choose. The planned path of the third UAV is shown in the blue line in the Fig. 7(b).

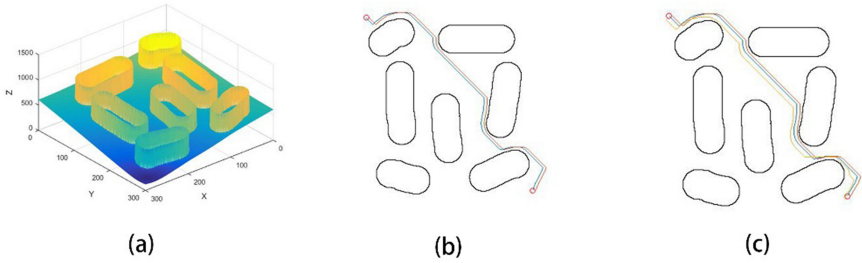
It can be seen from the above experiments that the application effect of the artificial potential field in the cluster environment is not ideal. It needs to be further improved to get better results.

## 4.2 GCA-APF

In this section, we use the GCA-APF model to perform obstacle avoidance operations for UAV clusters. The obstacle map in Fig. 5(a) is still used in the experiment. In the GCA-APF, the drones in the group will not treat other drones as obstacles, so whether it is two drones or three drones, the experimental potential field map is consistent, as the picture shows. Among them, the path of two UAVs flying together is shown in the figure. The path of the coordinated flight of the three drones is shown in the Fig. 8.

It can be seen that when the UAV uses the GCA-APF model to fly, the cluster generally maintains a complete formation. When the path is relatively narrow, the speed

of the entire cluster will slow down, and the UAV near the obstacle will move closer to the center of mass to avoid the obstacle.



**Fig. 8.** Experimental results of GCA-APF algorithm. (a) UAV potential field diagram (b) Path planning of two UAVs clusters (c) Path planning for three UAVs clusters

**Table 1.** Comparison of GCA and GCA-APF.

Path planning algorithm	Planning times	Success rate	Average time
GCA	50	78%	1.536 s
GCA-APF	50	100%	1.023 s

And in this environment, we use the traditional GCA for path planning, and compared with the GCA-APF in this paper, the results are shown in the Table 1. We can see that the traditional GCA has about 22% planning failures. At the same time, due to its field of view is much narrower than GCA-APF, leading to the longer plan path, which cannot guarantee the optimal solution. The complete flight time of the drone is longer, too.

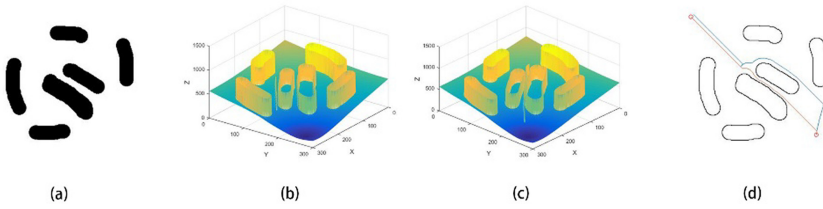
### 4.3 ICA-APF

In this section, we use the ICA-APF model to perform obstacle avoidance operations for UAV clusters. Since the ICA-APF model is generally applicable to relatively small obstacles, we use obstacles as shown in the Fig. 9(a). During the flight of a drone, the ICA-APF mode is turned on, which causes the drones in the cluster to see other drones as obstacles, so different drones use different potential field diagrams. When there are two drones in the cluster, their potential field diagrams are Fig. 9(b), (c). It can be seen that in these two figures, there is a line with a very high potential field, which is the potential field generated when other drones are regarded as obstacles. The planned path is shown in Fig. 9(d).

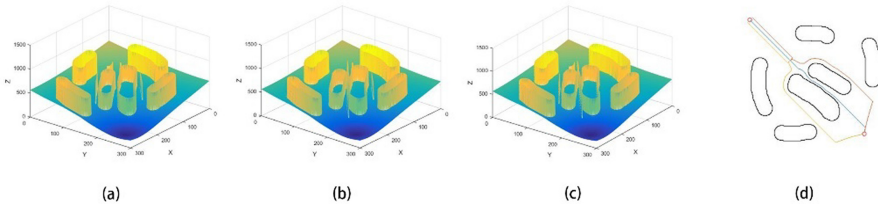
When there are three drones in the cluster, we use the ICA-APF algorithm for path planning again. The distribution of obstacles is consistent with the previous experiment which shown in Fig. 9(a), and its experimental results are shown in the Fig. 10.

It can be seen that the drone cluster can bypass obstacles and reach the end point, but it can also be found that because the obstacles used in the experiment are relatively





**Fig. 9.** Experimental results of ICA-APF algorithm. (The number of UAVs is two) (a) Schematic diagram of obstacles used in the experiment (b), (c) The artificial potential field diagram of two drones (d) Path planning for the UAVs clusters



**Fig. 10.** Experimental results of ICA-APF algorithm. (The number of UAVs is three) (a), (b), (c) The artificial potential field diagram of three UAVs (d) Path planning for the UAVs clusters

**Table 2.** Comparison of ICA and ICA-APF.

Path planning algorithm	Planning times	Success rate	Average time
ICA	50	98%	1.096 s
ICA-APF	50	98%	1.023 s

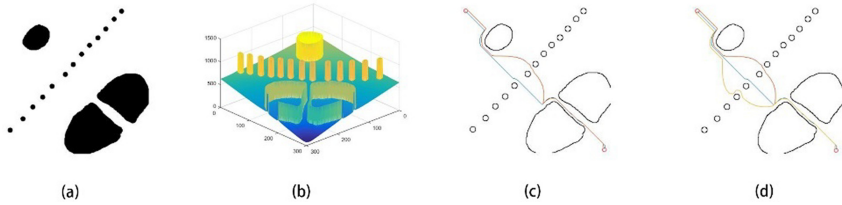
large, the distance that the drone bypasses is relatively long, and the cost of obstacle avoidance is relatively large.

We also use the traditional ICA for path planning, and compared with the ICA-APF in the same environment. The results are shown in the Table 2. It can be seen that the experimental results of the ICA and APF algorithms are basically the same. This may be due to the relatively simple environment currently in use and the inability to take advantage of the ICA-APF algorithm.

**4.4 Adapt Switch Mechanism Choosing GCA-APF or ICA-APF**

In this section, we conduct experiments on the ASM-APF algorithm model proposed in this paper. In order to make the experimental results more intuitive, we constructed a map with both small obstacles and relatively large obstacles. The schematic diagram is shown in the Fig. 11.

We use two and three UAVs to fly separately. In order to simplify the calculation, the conversion threshold of the GCA-APF mode and ICA-APF mode we selected in the



**Fig. 11.** Experimental results of adapt switch mechanism choosing GCA-APF or ICA-APF algorithm. (a) Schematic diagram of obstacles used in the experiment (b) The artificial potential field diagram of three UAVs (c), (d) Path planning for the UAVs clusters (The number of clusters is two and three)

experiment is the size of a drone diameter. That is, when the diameter of the obstacle exceeds the drone, the GCA-APF obstacle avoidance mode is turned on, otherwise the ICA-APF obstacle avoidance mode is used. The experimental results are shown in the figure. It can be seen that when a small obstacle is encountered, the drone will spread out, bypassing the obstacle at a lower cost. When encountering obstacles that are so large that only one path can pass through, the UAV can be aggregated into a UAV queue in the form of GCA to pass through the obstacle, achieving a relatively good experimental effect.

We also compare the current algorithm with the GCA-APF. The reason why not compared with the ICA-APF is that there is only one path to the target point in the current environment and only one drone is allowed to pass, so the ICA-APF algorithm cannot reach the target point. The experimental results are as shown in Table 3. And it can be calculated that the current efficiency has increased by about 15.25%.

**Table 3.** Comparison of current algorithm and GCA-APF.

Path planning algorithm	Planning times	Success rate	Average time
Adapt algorithm	50	98%	1.656 s
GCA-APF	50	96%	1.954 s

## 5 Conclusion and Further Work

This paper discusses the shortcomings of the artificial potential field algorithm in the cluster environment, and improves the artificial potential field algorithm by combining the GCA and ICA bird swarm intelligence models so that it can be applied to the cluster. Further, we designed an adapt switch mechanism for dynamically choosing GCA-APF or ICA-APF in contexts of different obstacle environment. The algorithm allows the cluster to adaptively select the ICA model and GCA model according to the surrounding environment, and achieves relatively good results in the experiment.

In the future work of optimizing our method, we plan to combine the artificial potential field algorithm with other swarm intelligence algorithms (such as bee colony, ant colony and other ideas) to explore its advantages and disadvantages.

**Acknowledgements.** This work was supported by the National Natural Science Foundation of China (61972025, 61802389, 61672092, U1811264, 61966009), the Fundamental Research Funds for the Central Universities of China (2018JBZ103, 2019RC008), Science and Technology on Information Assurance Laboratory, Guangxi Key Laboratory of Trusted Software (KX201902).

## References

1. Zhou, Z., Wang, J., Zhu, Z., et al.: Tangent navigated robot path planning strategy using particle swarm optimized artificial potential field. *Optik* **158**, 639–651 (2018)
2. Meng, X.B., Gao, X.Z., Lu, L., et al.: A new bio-inspired optimisation algorithm: bird Swarm algorithm. *J. Experimental Theor. Artif. Intell.* **28**(4), 673–687 (2016)
3. Aljarah, I., Faris, H., Mirjalili, S., et al.: Evolving neural networks using bird swarm algorithm for data classification and regression applications. *Cluster Comput.* **22**(4), 1317–1345 (2019)
4. Mazzeo, S., Loiseau, I.: An ant colony algorithm for the capacitated vehicle routing. *Electron. Notes Discrete Math.* **18**, 181–186 (2004)
5. Saied, M., Slim, M., Mazeh, H., et al.: Unmanned aerial vehicles fleet control via artificial bee colony algorithm. In: 2019 4th Conference on Control and Fault Tolerant Systems (SysTol), pp. 80–85. IEEE (2019)
6. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. *Appl. Math. Comput.* **214**(1), 108–132 (2009)
7. Reynolds, C.W.: Flocks, herds and schools: a distributed behavioral model. In: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, pp. 25–34 (1987)
8. Jha, P.K., Ghose, D.: Avoidance between cluster of UAVs. In: Proceedings of International Conference on Computational Intelligence, Robotics and Autonomous Systems (2003)
9. Cai, Z., Lou, J., Zhao, J., et al.: Quadrotor trajectory tracking and obstacle avoidance by chaotic grey wolf optimization-based active disturbance rejection control. *Mech. Syst. Signal Process.* **128**, 636–654 (2019)



# Automatic Counting of Railway Tools Based on Deep Learning

Wei Wei<sup>1</sup>, Jin Yang<sup>1</sup>, Sikai Wang<sup>1</sup>, Deng Chen<sup>1,2</sup>(✉), Yanduo Zhang<sup>1</sup>, Zihang Zhang<sup>1</sup>, Wei Liu<sup>1</sup>, Gonghao Duan<sup>1</sup>, Chaohui Zheng<sup>3</sup>, Jianping Ju<sup>4</sup>, and Jianyin Tang<sup>4</sup>

<sup>1</sup> Hubei Province Key Laboratory of Intelligent Robot,  
Wuhan Institute of Technology, Wuhan, China  
dchen@wit.edu.cn

<sup>2</sup> Lingyun Technology Group Co. LTD, Wuhan, China

<sup>3</sup> Wuhan Railway Vocational College of Technology, Wuhan, China

<sup>4</sup> College of Artificial Intelligence, Hubei Business College, Wuhan, China

**Abstract.** The daily railway operation and maintenance suffers from a severe problem of the loss of railway tools. Aiming at the problem, an automatic counting method of railway tools based on deep learning is proposed. Our method is based on toolkit images obtained before a toolkit is delivered to a worker and after it is returned. By recognizing and comparing railway tools in toolkit images, our approach can detect missing tools automatically. Our work extends the research of object detection to the practical application of railway operation and maintenance. In order to resolve the sparsity problem of image samples, an image dataset augmentation algorithm is used for oversampling. Combined with the transfer learning strategy, our approach is able to count railway tools based on images automatically and accurately in complex outdoor environment. Experiments were conducted based on real-world datasets. Results show that our method can detect railway tools accurately with a mAP of 83%, which satisfies requirements of practical applications. Above all, our work provides a strong technical basis for intelligent railway operation and maintenance.

**Keywords:** Railway tools counting · Railway operation and maintenance · Deep learning · Object detection · Data augmentation · Transfer learning

## 1 Introduction

The railway is an important part of the transportation system. The railway department usually works at night to ensure the safe operation of the railway. During the process of railway operation and maintenance, railway tools are always missing which leads to huge economic losses to the railway department. At present, the solution is to obtain the toolkit images before a toolkit is delivered to a worker and after it is given back. Then, by comparing and analyzing the toolkit images manually to find the missing tools.

---

W. Wei and J. Yang—contributed equally.

However, this approach may suffer from subjective mistakes and is less time efficient, which cannot satisfy requirements of intelligent railway operation and maintenance.

In order to reduce the cost of railway operation and maintenance, we propose an automatic counting method for railway tools. Our approach is based on deep learning techniques for image object detection. Image object detection is a critical task of computer vision. In recent years, various kinds of object detection approaches based on ConvNets have been proposed, such as R-CNN [1], Fast-RNN [2], Faster-RNN [3], YOLO [4], SSD [5], etc. These approaches have a high accuracy and made a great success in many applications, such as face recognition [6, 7], human pose estimation [8–10], and automatic sorting of agricultural products [11, 12], etc. However, image object detection for railway tools has the following difficulties: 1) railway operation and maintenance is generally carried out at night, toolkit images captured under artificial light have problems of low and uneven illumination which may influence object detection accuracy; 2) toolkit images are achieved in an open environment, the various different kinds of complex background of toolkit images further increase the difficulty of railway tool detection; 3) railway tools have as many as more than a thousand different varieties. It is difficult to collect enough training samples for all kinds of railway tools. The sparsity of image samples may cause overfitting.

In order to resolve the above problems in the automatic counting of railway tools. We introduce a dataset augmentation [16] algorithm to pre-process image samples, which can effectively increase the diversity of training samples. Aiming at the problem of insufficient training samples, we employ the transfer learning strategy [13]. First, a deep neural network for object detection is pre-trained based on a commonly used image dataset. Then, a self-constructed toolkit image dataset is applied for retraining. Experimental results show that our approach can effectively detect railway tools from images with a mAP of 83%, which can satisfy requirements of practical applications.

Our main contributions are summarized as follows:

- A railway tool image dataset is constructed, with a total of 1208 images, which provides a data basis for the future research of railway tool detection.
- An automatic counting method of railway tools is proposed. By combining dataset augmentation algorithm with transfer learning strategy, our approach can mitigate the sparsity problem of training data and count railway tools from real-world images accurately.
- Extensive experiments were conducted based on real word datasets and promising results were achieved.

## 2 Our Technique

### 2.1 General Framework

As shown in Fig. 1, our automatic counting method of railway tools comprises modules of dataset augmentation, transfer learning and railway tool counting. In order to mitigate the problem of small dataset, we adopt the transfer learning strategy for railway tool detection. Firstly, the ImageNet dataset is used to pre-train the object detection model. Then, the pre-trained model parameters are further tuned by a retraining process with a

self-constructed railway tool image dataset. Since the railway tool image dataset is small, a dataset augmentation technique is employed. Based on spatial geometry transformation and pixel color transformation, the dataset augmentation method can increase the size of railway tool image dataset by five times. Leveraging the above object detection model, we can recognize missing railway tools effectively by comparing railway tool detection results from toolkit images captured before a toolkit is delivered to a worker and after it is given back.

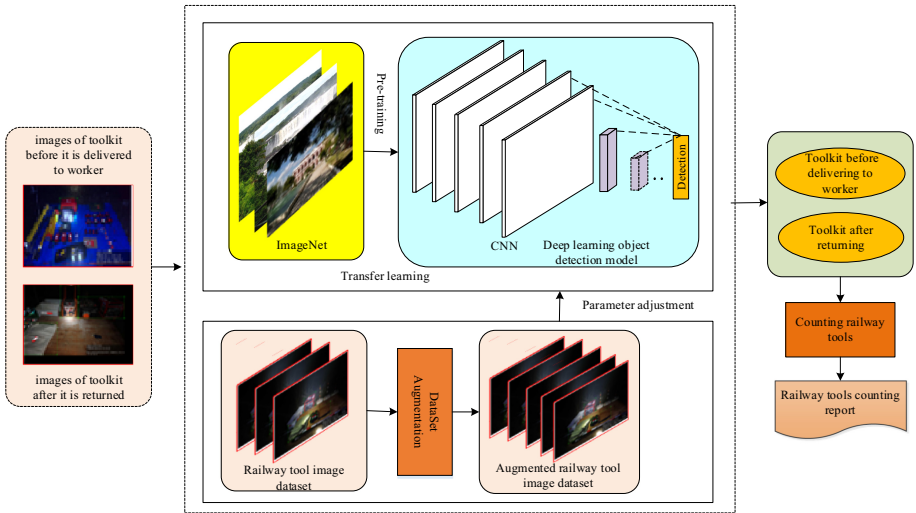


Fig. 1. The overall framework of railway tool automatic counting.

## 2.2 Dataset Augmentation

Since railway tools have a large number of varieties, the collected railway tool image dataset may be unbalanced, that is, image samples for a railway tool may be rare. On the other hand, collecting a large amount of railway tool images in outdoor environment will cost significant human effort. In order to mitigate the overfitting problem caused by insufficient training data, we apply a dataset augmentation algorithm to collect railway tool images.

The dataset augmentation algorithm derives new image samples by performing spatial geometry transformation and pixel color transformation to existing images. The outline of the algorithm is shown in Algorithm 1. Let's assume that  $R$  is the self-constructed railway tool image dataset. For each image  $I \in R$ , we perform brightness regulation, translation transformation, rotation transformation, scale transformation and shear transformation to  $I$  respectively. The dataset  $I'$  and the transformed images form an augmented dataset  $R'$ .

- **Brightness regulation:** we transform image  $I$  to  $I'$  regarding the brightness component in the HSV color space according to equation  $val(I') = val(I) \times \gamma$ , where  $val(I)$  is

the brightness component value of image  $I$ , and  $\gamma \in [0.8, 1.2]$  is a random brightness regulation factor.

- Translation transformation: we perform the translation transformation to image  $I$  based on the translation matrix  $A$ .

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dx & dy & 1 \end{pmatrix}$$

$$dx = \text{width}(I) \times g_x$$

$$dy = \text{height}(I) \times g_y \quad (1)$$

where  $\text{width}(I)$  and  $\text{height}(I)$  denote the width and height of image  $I$  respectively.  $g_x \in [-0.2, 0.2]$ , and  $g_y \in [-0.2, 0.2]$  are random translation factors in horizontal and vertical directions, respectively,  $dx$  and  $dy$  denote the horizontal and vertical translation distance respectively.

- Rotation transformation: we perform the rotation transformation to image  $I$  based on the rotation matrix  $R$ .

$$R = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

where  $\theta \in [-10, 10]$  is a random angle of rotation.

- Scaling transformation: we conduct scaling transformation to image  $I$  based on the scaling matrix  $S$ .

$$S = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

where  $s_x \in [0.8, 1.2]$  and  $s_y \in [0.8, 1.2]$  are random scaling factors in horizontal and vertical directions, respectively.

- Shear transformation: we perform horizontal shear transformation to image  $I$  based on the matrix  $H$ .

$$H = \begin{pmatrix} 1 & k & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

where  $k \in [-0.2, 0.2]$  is the random shear factor in horizontal direction.

The augmentation algorithm applies five different transformations to an image. Therefore, our technique can increase the number of railway tool images by five times. Additionally, our approach can increase the diversity of image samples in terms of geometric and color characteristics, which is beneficial for improving the object detection accuracy of railway tools.

---

**Algorithm 1** Dataset Augmentation Algorithm

---

**input :**  $R$  : railway tool image dataset

**output:**  $R'$  : augmented dataset

**symbols:**  $T$  : The set of geometric transformation matrices, including translation matrix, rotation matrix, scaling matrix and shear matrix.

**Method:**

- 1) **for**  $I$  in  $R$  **do**
  - 2)   **add**  $I$  to  $R'$
  - 3)    $I' \leftarrow I$
  - 4)    $val(I') \leftarrow val(I) \times \gamma, \gamma \in [0.8, 1.2]$
  - 5)   **add**  $I'$  to  $R'$
  - 6)   **for**  $M$  in  $T$  **do**
  - 7)      $I' \leftarrow I$
  - 8)      $I' = I \cdot M$
  - 9)     **add**  $I'$  to  $R'$
  - 10) **output:** the augmented dataset  $R'$
- 

### 2.3 Railway Tool Detection Model

We detect railway tools from images based on the SSD [5] (single shot multiBox detector) model. As shown in Fig. 2, the SSD model uses the former 13 layers of MobileNet [14] as the backbone network. It detects objects based on features of multiple layers including conv11, conv13, conv14\_2, conv15\_2, conv16\_2, and conv17\_2. Since SSD model comprehensively utilizes feature maps of multiple convolutional layers, it has excellent detection performance for both large and small objects. The shallow convolutional layers (such as conv11 and conv13) have a small receptive field, which are suitable for detecting small objects. The deep convolutional layers (such as Conv14\_2, Conv15\_2, Conv16\_2 and Conv17\_2) have a large receptive field, which are appropriate for detecting large objects. In order to detect objects, the SSD model generates bounding boxes of different scales at each pixel. Based on the generated initial bounding boxes, classification and regression models are used to predict real bounding boxes of objects.

Apart from that, SSD model adopts the batch normalization [15] technique, which is described formally in Eqs. (5)–(8).

$$\mu = \frac{1}{m} \sum_{i=1}^m x_i \quad (5)$$



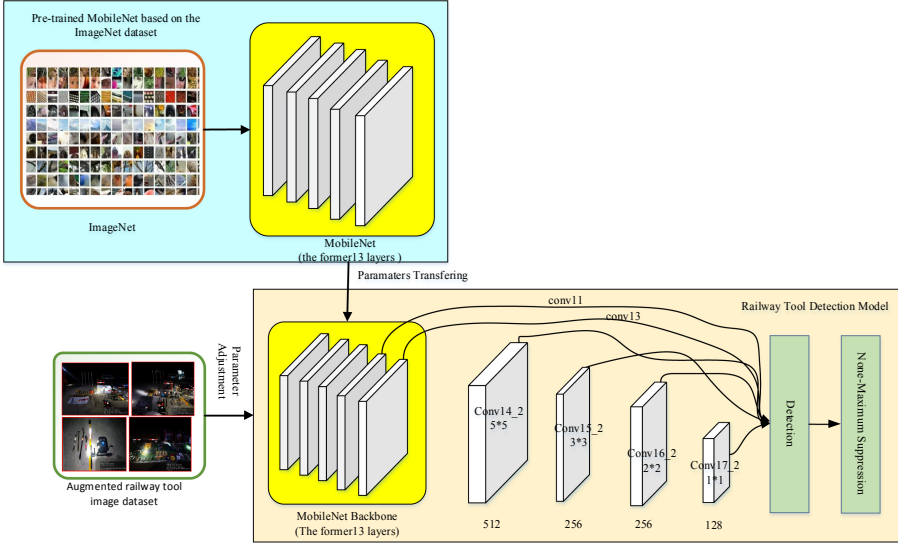


Fig. 2. The structure of the SSD model for railway tool detection.

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu)^2 \tag{6}$$

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2}} \tag{7}$$

$$y_i = \gamma \hat{x}_i + \beta \tag{8}$$

where  $x_i, i = 1 \dots m$  is the input image samples,  $m$  is the number of samples,  $\mu$  is the sample mean,  $\sigma^2$  is the sample variance,  $\hat{x}_i$  is the normalized output, and  $y_i$  is the output after the batch normalization operation,  $\gamma$  and  $\beta$  are the scaling and translation parameters, respectively. The batch normalization technique is able to speed up the training process, reduce overfitting and improve the generalization ability.

The excellent performance and powerful object detection capability of the SSD model provide a strong assurance for the accurate counting of railway tools of different scales in outdoor complex environment.

To avoid overfitting caused by the sparsity of training data, we employ the transfer learning strategy. First, we pre-train MobileNet based on the ImageNet dataset. Then, we transfer parameters of the former 13 layers of the pre-trained MobileNet to SSD model as initial weight values. After that, we retrain SSD model with the augmented railway tool image dataset. The retraining process will further adjust parameters of the SSD model and make it more appropriate for the detection of railway tools.

What should be noted is that, the size of the input layer of the SSD model is  $300 \times 300$  pixels. Thus, we should resize input images to the required dimension. When an image has the same width and height, we perform the task based on a scaling transformation.

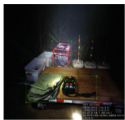
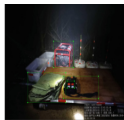


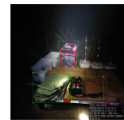







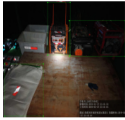
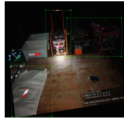

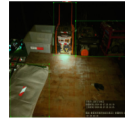








Otherwise, we fill the image with zero pixels to convert it into an image with the same width and height, and then scale it the required dimension.

### 3 Experimental Evaluation

#### 3.1 Dataset

We collected a total of 1208 real world toolkit images from Wuhan Railway Bureau and denote it by dataset 1. Then, we get the augmented dataset 2 by applying the dataset augmentation method on dataset 1, which contains 7248 toolkit images. Table 1 gives some representative images augmented by the dataset augmentation algorithm. The first column presents the original images and the second to six columns are images output by rotation transformation, translation transformation, brightness regulation, scaling transformation and shear transformation, respectively. As can be seen from Table 1, the augmented images have been slightly adjusted in spatial geometry as well as pixel color, while maintaining the basic structure and color distribution of the original images, which can effectively increase the diversity of the image dataset.

**Table 1.** Part of the image samples in augmented dataset 2

original images	rotation transformation	translation transformation	brightness regulation	scaling transformation	shear transformation
					
					
					
					

### 3.2 Experiment Evaluation

**Experiment Configuration.** The experiments were performed on a high-performance graphics server. It's configurations are as follows:

- CPU: Intel® Core™ i7-8700 CPU @ 3.20 GHz;
- RAM: 16 GB DDR3;
- GPU: GeForce GTX-1080Ti
- Software Environment: Ubuntu 18.04LTS, Caffe1, OpenCV3.4.4, Python3.5

**Evaluation indicator.** In this paper, we use the mAP (mean average precision) as the evaluation index for our tool detection model. The mAP is the mean of APs (average precision), which can be computed by the area enclosed by the P-R curve and coordinate axes. The formal description of AP is presented as follows.

$$AP = \int_0^1 p(r)dr \quad (9)$$

where  $p(r)$  is the Precision-Recall curve. Precision and recall can be computed based on the Eqs. (10) and (11) respectively

$$precision = \frac{TP}{TP + FP} \quad (10)$$

$$recall = \frac{TP}{TP + FN} \quad (11)$$

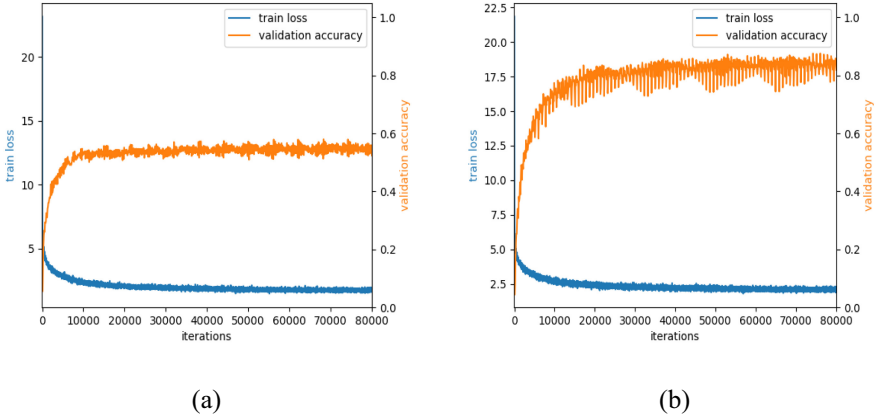
where the TP, FN and FP denote the number of true positives, false negatives and false positives, respectively.

**Evaluation Method.** In order to verify the railway tool detection performance and the effect of the dataset augmentation algorithm, we train the railway tool detection model with dataset 1 and datasets 2, respectively.

Railway tools have more than a thousand different varieties, for ease of display, we have shown the 10 types of most frequently used railway tools in our experiment. They are shovel, motor, wire, electric drill, workbag, woven bag, iron wire, bucket, signal light and lighting lamp. Firstly, we used the LabelImage [17] to label the 10 types of tools in both dataset 1 and dataset 2. Then, we divided dataset 1 and dataset 2 into the training dataset, verification dataset and testing dataset according to the ratio of 6: 2: 2, respectively. Next, we adopted the Stochastic Gradient Descent (SGD) of Newtonian Momentum [18] to train and optimize the railway tool detection model. The batch size is one of the most important parameters, whose value directly affects the model's training effect. A large value may cause the memory overflow and result in training failure, while a small value will increase the training time and is not easy to converge. Considering our experiment configurations, we employed a batch size of 16. The learning rate and momentum were assigned a value of 0.005 and 0.9 respectively. Additionally, we change the learning rate every 5000 iterations to converge the training loss. After 80000 iterations, our model reaches a stable state.

### 3.3 Experiment Result

**Railway Tool Detection Result.** For notational convenience, we use model 1 and model 2 to denote the railway tool detection models trained with dataset 1 and dataset 2, respectively. Figure 3 (a) and (b) present the training results of 10 types of tools achieved by model 1 and model 2, respectively. The horizontal axis represents the number of iterations, and the vertical axis denotes the training loss and validation accuracy. The blue curve reflects the training loss, and the yellow one is the validation accuracy.



**Fig. 3.** (a) and (b) shows the training loss and validation accuracy of model 1 and model 2 respectively. (Color figure online)

It can be seen from Fig. 3 (a) and (b) that the training loss begins to converge after 40,000 iterations and finally reaches a stable value between 2 and 2.5. The validation accuracy of model 1 comes to stabilize at 55% after 10,000 iterations, while the validation accuracy of the model 2 gradually converges to 83% at 30,000 iterations. The curves of training loss and validation accuracy indicate that the training model can effectively converge in a short time. Furthermore, model 2 has a higher validation accuracy than model 1, which justifies the effectiveness of our dataset augmentation algorithm.

Table 2 shows the AP of Model 1 and Model 2 regarding 10 types of railway tools.






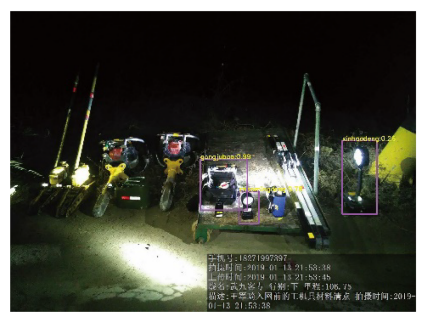
As we can see, the workbag has the highest AP in Model 1 with 79.17%, while the wire has the lowest AP with 29.31%. However, in Model 2, the iron wire has the highest AP with 100.00%, and the signal light has the lowest AP in 63.52%. Through comparative analysis, we can find that the model 2 outperforms the model 1. The most significant increase in AP is the wire, which increased from 29.31% to 79.67%. By comparing the mAPs of the two models, it can be found that the dataset augmentation method can effectively improve the accuracy of railway tool detection. The mAP of Model 2 reached 83%, which demonstrated that our method can satisfy requirements for the practical application requirements of automatic counting of railway tools.

**Table 2.** AP comparison of model1 and model2

AP	Model	
	Model 1	Model 2
shovel	53.99	73.18
motor	67.42	92.73
wire	<b>29.31</b>	79.67
electric drill	31.80	81.02
workbag	<b>79.17</b>	90.08
woven bag	65.14	95.64
iron wire	71.56	<b>100.00</b>
bucket	49.63	79.33
signal light	32.36	<b>63.52</b>
lighting	50.04	76.56
mAP	53.00	83.00

In order to carry out a more detailed analysis of the detection capability of model2, detection results (red font) of several types of tools with the highest and lowest AP are shown in Fig. 4, including object bounding boxes, classification labels and scores. It can be seen from Fig. 4 that railway tools with a large size (such as woven bags, iron wires, and motors) are more likely to be detected with a high AP, while the shovel, signal light and lighting lamp with a small size achieve low APs and are difficult to be detected. The above problem may be caused by the inherent limitation of SSD object detection model. In the future, we will improve the detection ability of SSD model for small objects.

**P-R Curve.** For further comparison analysis of Model 1 and Model 2, we present the P-R curves of 6 types of tools in Fig. 5. As we can see, the blue and yellow curves are the P-R curves of Model 1 and Model 2 respectively. It can be known from the P-R curve that for each type of tool, the area enclosed by the yellow curve and coordinate axes is larger than that enclosed by the blue curve and coordinate axes. It shows that the performance of Model 2 is better than the Model 1, and can obtain higher detection accuracy.

 <p>Handwritten text in image: 手机号:18271997173 拍摄时间:2018 08 13 06:04:56 上传时间:2018 08 13 06:05:06 描述:信阳东检测班组人钟世桐地前上机具清点 拍摄时间:2018 08 13 06:04:56</p>	 <p>Handwritten text in image: 手机号:18271997173 拍摄时间:2018 08 13 06:04:56 上传时间:2018 08 13 06:05:06 描述:信阳东检测班组人钟世桐地前上机具清点 拍摄时间:2018 08 13 06:04:56</p>
<p>woven bags : 0.93 shovel : 0.75 workbag : 0.31</p>	<p>woven bags : 1.00 shovel : 1.00</p>
 <p>Handwritten text in image: 手机号:18271997173 拍摄时间:2018 08 13 06:04:56 上传时间:2018 08 13 06:05:06 描述:信阳东检测班组人钟世桐地前上机具清点 拍摄时间:2018 08 13 06:04:56</p>	 <p>Handwritten text in image: 手机号:18271997173 拍摄时间:2018 08 13 06:04:56 上传时间:2018 08 13 06:05:06 描述:信阳东检测班组人钟世桐地前上机具清点 拍摄时间:2018 08 13 06:04:56</p>
<p>motors : 1.00 lighting lamp : 1.00 workbag : 0.99</p>	<p>motors : 0.97 woven bags : 0.94 lighting lamp : 0.40 shovel : 0.31</p>
 <p>Handwritten text in image: 手机号:18271997173 拍摄时间:2018 08 13 06:04:56 上传时间:2018 08 13 06:05:06 描述:信阳东检测班组人钟世桐地前上机具清点 拍摄时间:2018 08 13 06:04:56</p>	 <p>Handwritten text in image: 手机号:18271997173 拍摄时间:2018 08 13 06:04:56 上传时间:2018 08 13 06:05:06 描述:信阳东检测班组人钟世桐地前上机具清点 拍摄时间:2018 08 13 06:04:56</p>
<p>iron wires : 1.00 shovel : 0.41</p>	<p>signal light : 0.26 lighting lamp : 0.78 workbag : 0.99</p>

**Fig. 4.** The detection results of the railway tool detection model 2 (Color figure online)

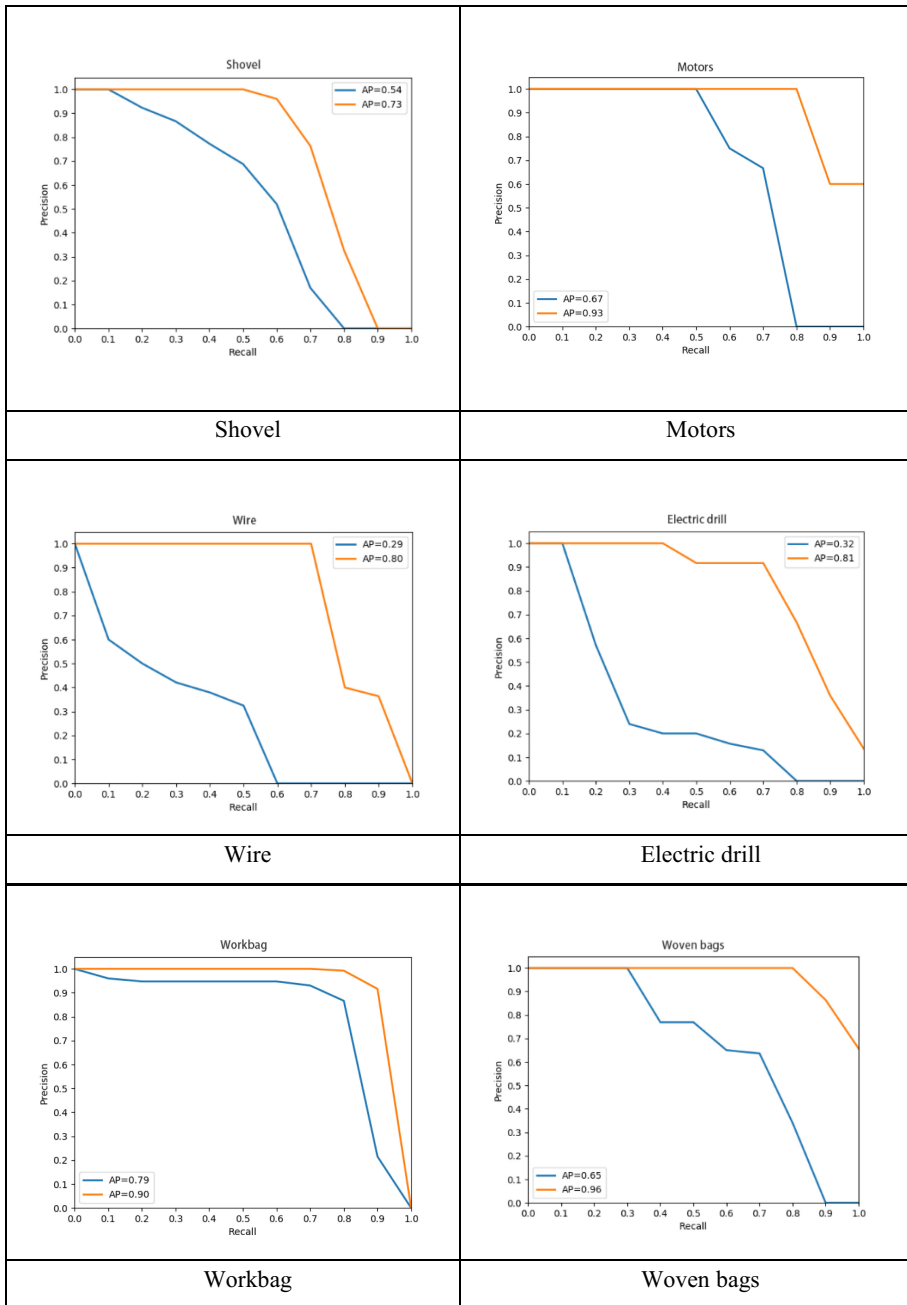


Fig. 5. The P-R curves of 6 types of tools (Color figure online)

## 4 Conclusion and Future Work

The automatic counting of railway tools is an important part of intelligent railway operation and maintenance. In this paper, we have provided an effective automatic counting method for railway tools by applying a MobileNet-SSD network model. We adopt the dataset augmentation algorithm and transfer learning strategy to oversample the toolkit images to resolve the sparsity problem of image samples. The experiments on railway tool dataset (augmented and unaugmented) prove that our method has a good effect on detecting railway tools in complex outdoor environment. We use the method to compare the railway tool detection results in the toolkit image captured before the toolkit is delivered to the worker and after it is given back, which can effectively recognize the missing railway tools.

In the future, it is necessary to improve the detection accuracy of small objects under uneven illumination or occlusion, and to further study the multi-scale object detection algorithms deeply to improve and optimize our model.

**Acknowledgements.** This work was supported in part by Education Sciences Planning of Hubei Province of China (No.2019GA090), Hubei Province Technology Innovation Project (No.2019AAA045), Provincial Undergraduate Training Programs for Innovation and Entrepreneurship (S201910490051) and Research Project of Hubei Chinese Vocational Education Association (No. HBZJ2020016).

**Conflict of Interest.** Authors have no conflict of interest to declare.

## References

1. Girshick, R., Donahue, J., Darrell, T.: Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(1), 142–158 (2016)
2. Girshick, R.: Fast R-CNN. In: *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile (2016)
3. Ren, S., He, K., Girshick, R.: Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* Montreal, Canada **39**(6), 1137–1149 (2015)
4. Redmon, J., Divvala, S., Girshick, R.: You only look once: Unified, real-time object detection. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA (2016)
5. Liu, W., Anguelov, D., Erhan, D.: SSD: Single Shot MultiBox Detector, pp. 21–37 (2015)
6. Sarkar, S.D., Ajitha, S.K.B.: Face recognition using artificial neural network and feature extraction. In: *2020 7th International Conference on Signal Processing and Integrated Networks (2020)*
7. Arsal, M., Wardijono. B.A., Anggraini, D.: Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN (2020)
8. Zhang, Z., Wang, C., Qin, W.: Fusing Wearable IMUs with Multi-View Images for Human Pose Estimation: A Geometric Approach (2020)
9. Hua, G., Li, L., Liu, S.: Multipath affinity stacked—hourglass networks for human pose estimation. *Frontiers Comput. Sci.* **14**(4), 1–12 (2020)



10. Zheng, X., Chen, X., Lu, X.: A joint relationship aware neural network for single-Image 3D human pose estimation. *IEEE Trans. Image Process.* **29**, 4747–4758 (2020)
11. Chenjiao, T., Yilin, L., Dongfei, W.: Review on automatic navigation technologies of agricultural machinery. *J. Agric. Mech. Res.* (2020)
12. Wspanialy, P., Brooks, J., Moussa, M.: An Image Labeling Tool and Agricultural Dataset for Deep Learning (2020)
13. Yosinski, J., Clune, J., Bengio, Y.: How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* **27**, 3320–3328 (2014)
14. Howard, A.G., Zhu, M., Chen, B.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications (2017)
15. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on International Conference
16. Data-Augmentation Homepage: <https://blog.paperspace.com/data-augmentation-for-bounding-boxes/>. Accessed 21 Mar 2020
17. LabelImage Homepage: [www.label-image.com/](http://www.label-image.com/). Accessed 21 Mar 2020
18. Sutskever, I.: On the importance of initialization and momentum in deep learning. In: International Conference on Machine Learning (2013)



# Risk Assessment of Heterogeneous CPS Systems Under Different Proportions of Links

Hao Peng<sup>1,2</sup>, Zhe Kan<sup>1</sup>, Dandan Zhao<sup>1(✉)</sup>, Jianmin Han<sup>1</sup>, and Zhaolong Hu<sup>1</sup>

<sup>1</sup> College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321004, Zhejiang, China  
ddzhao@zjnu.edu.cn

<sup>2</sup> Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai 200240, China

**Abstract.** In this paper, we study a new coupled network model that multiple equal-mixed models to study the security of the coupled system in distributed heterogeneous environments. We propose a method for analyzing cascading failures, in which we can derive the critical threshold of the coupled system. Then we verify the correctness of the critical threshold by simulation experiments. And the simulation experiments are concluded that the coupled system exhibits a first-order phase transition near the critical threshold. The analytical methods we propose in this paper can analyze the wide range of applicability of coupling systems at various scales.

**Keywords:** Cyber-Physical Systems · Cascading failures · Interdependent network

## 1 Introduction

The development of science and technology has made our lives more and more convenient. The development of network technology has produced a variety of artificial networks. These artificial networks are used in our daily lives widely [14]. The continuous integration of technologies such as information perception, ubiquitous computing and management and control has realized the interconnection and deep integration of physical space and information space, and finally formed Cyber-Physical System (CPS). The security study for the coupled CPS system is mainly to study the coupled grid in the initial stage [1, 9]. Such as the smart grid, communication networks, transportation networks and so on. And these artificial networks are not independent, but are connected to each other and interact with each other. For example, the communication network and the power network interact with each other in the smart grid [4]. The failure of the nodes in the power network will invalidate the nodes in the communication network. The failure of the nodes in the communication network will also invalidate the nodes in the power network due to overload. Studying the behavior of these coupled systems after attack can effectively prevent the coupling system from causing

large damage, which has important practical significance. In recent years, all of the blackouts in the northeastern United States in 2003, the blackouts in Europe in 2006 and the blackouts in India in 2003 indicate that even a very small number of nodes fail in the initial stage, which will have a great impact on the entire power grid, affecting millions of people's lives and causing huge economic losses [16, 19]. With the widespread use of computer technology in life, a variety of networks are interconnected to form a coupled system. When nodes in a coupled CPS are attacked or invalidated due to internal reasons will cause cascading failure due to the coupling between the networks in the coupled system [12, 13]. Some infrastructures are interconnected together to bring great convenience to people's [10], but the coupling system composed of networks is vulnerable to random attacks or failures due to their own reasons. It will cause cascading failure between networks, which will cause great damage to the coupling system [20]. It has great practical significance to study and analyze the cascade failure of coupled CPS systems in order to improve the security of coupled CPS systems.

In order to improve the reliability of CPS, it is necessary to explore the cascading failures in actual interdependent CPS systems. Recently many researchers have paid more attentions in this research field. Currents research in smart grid systems [5, 11, 18] mainly focuses on failures about load balancing and load distribution. Most of these techniques rely on methods commonly used in distributed systems. Architecture for distributed generation way, which can help prevent cascading failures, is described in Ref. [15]. However, fault analysis and the impact of communication network on power grid were not mentioned. Optimization mechanisms have been used to balance demand and supply in Ref. [2]. Fault location method in cyber-physical has been investigated in Ref. [7, 17]. Obviously, existing work on modelling smart grid systems is mainly about extracting properties from physical systems and assumed associated cyber system and matching with some physical network families. For example, Huang et al. [8] proposed a mechanism that generates random topology power grids featuring the same topology and electrical characteristics generated from the real world. Zhu and Gao et al. [3, 6, 21] focused on the challenges of modeling cyber-physical systems that arise from the intrinsic heterogeneity and sensitivity to timing. Specific technologies applied in some cyber-physical systems include hybrid system models and heterogeneous models of computation, the use of domain-specific ontologies to strengthen modularity, and the joint modeling of functionality and implementation architectures. Then we can find that most of the previous research works mainly consider the process of cascading failure in the same type of interdependent network, and does not perform reliability analysis on networks of different networks types. However, the actual interdependent CPS systems are often different networks types, so this paper will study the reliability of interdependent CPS systems under different networks types.

The remainder of the paper is organized as follows: We describe the process of modeling and the characteristics of the established model specifically in Sect. 2. In Sect. 3, we perform a detailed mathematical analysis of each step of the cascade failure and obtained an iterative equation. In Sect. 4, we use simulation

experiments to verify the correctness of the theoretical analysis, and through simulation experiments to analyze the factors affecting the security of the coupled CPS system. We summarize the work of this paper and briefly describe the work to be done in the future in Sect. 4.2.

## 2 System Model

In this section, we first establish a coupled network model with complex connections between networks. Then we explain some basic concepts in network science.

### 2.1 Model Construction

Through the research and analysis of the coupled system in life, we know that the coupled network is generally composed of two or more networks, and the connections between the networks are diverse. Some are one-to-one connections, some are proportional connections, and more are mixed cases of various ratios. The previous researches mainly study one-to-one and random connections, but the two mixed connections methods are not consistent with the actual ones, and cannot analyze the realistic network very well. So, in this paper, the model of the coupled system consists of two networks which are both ER networks. The connection between the two networks that make up the coupled system is a mixture of multiple equal connections. The connections between networks in real life coupling system are complicated. We first simplify the connection between the networks. Then classify the connections between the networks, and regard the connection between the networks as a hybrid connection connected by several equal ratios. In order to clearly show the analysis process of cascading failure, we assume that there are only two ways to connect between two networks in a coupled system for 2:1 and 1:1.

### 2.2 Concept

In a system of coupled networks, when one of the networks that make up the coupled network fails due to random attack, the other network will also fail because of the coupling between the two networks, and this failure will iteratively proceed. We call this failure as cascade failure. We don't know when the cascade failure will stop during the cascade failure, but we know that when one network in the coupled network is attacked, only the functional nodes that satisfies the following two conditions in the network as follows:

- The node must belong to the giant connected component;
- The node must be connected to a functional node in internal network.

When a network in coupled network is attacked, the failure of the nodes in one network affects the function of the nodes in the other network. If none of the two networks fails or the two networks completely collapse, the network reaches

steady state. This iterative failure process is called cascading failures. Cascading failures are a common failure process in coupled systems. If cascading failures are not controlled, cascading failures can cause severe damage.

### 3 Analysis of Cascading Failures Process

In this section, we will perform detailed mathematical analysis of each step of the cascade failure, and finally we will get an iterative equation. For the convenience of analysis, we denote A and B as the two networks that make up the coupled system. The number of nodes in network A and network B is represented by  $N_A$  and  $N_B$ , respectively. The two networks that compose the coupled network one is SF network, the other is ER network. The generating function of the SF network is  $G_{A0}(z) = \sum_k P_A(k)z^k$ . Analogously, the generating function of the ER network is  $G_{B0}(z) = \sum_k P_B(k)z^k$ . Then the generating function of the underlying branching processes is

$$G_{A1}(z) = \frac{G'_{A0}(z)}{G'_{A0}(1)} \tag{1}$$

We denote the number of nodes remaining after the node has been removed as  $N'_{A1}$ , we know that  $N'_{A1} = pN_A$ . The fraction of the nodes belonging to the giant connected component to the number of nodes is

$$g_A(p) = 1 - G_{A0}[1 - p(1 - f_A)] \tag{2}$$

Where  $f_A$  is function of  $p$ .  $f_A$  and  $p$  satisfy the following equation:

$$f_A = G_{A1}[1 - p(1 - f_A)] \tag{3}$$

#### 3.1 Random Failure in Network A

We assume that after being attacked, the proportion of deleted nodes is  $1 - p$ . So the number of remaining nodes in network A is

$$N'_{A1} = pN_A = \mu'_1 N_A \tag{4}$$

We denote the giant component as  $N_{A1}$ , then we can obtain:

$$\mu'_1 = p \tag{5}$$

#### 3.2 Cascading Failures on Network B

Owing to network A and network B depends on each other, nodes in network B will fail because of the failure of nodes in network A. We can calculate the number of nodes in network B that connect to nodes in network A:

$$N'_{B2} = \frac{(\frac{2}{3}N_A)}{2}(\mu_1^2 + 2\mu_1(1 - \mu_1)) + \frac{1}{3}N_A\mu_1 = \frac{1}{2}(3\mu_1 + \mu_1^2)N_B \tag{6}$$

Then we will again apply the apparatus of generating functions and calculate the number of nodes in network B that belong to the giant connected component:

$$N_{B2} = g_B(\mu'_2)N'_{B2} = \mu'_2 g_B(\mu'_2)N_B = \mu_2 N_B \tag{7}$$

### 3.3 Further A-Nodes Cascading Failure Due to B-Node Failures

Due to the iterative effect of cascading failure, the failure of nodes in network B will invalidate the nodes in network A. After the cascading failure of the first step and the second step, the connection relationship between the networks becomes complicated, but according to the initial connection relationship, we can calculate the proportion of various connection relationships. The fraction of nodes belong to the equal ratio connection 2:1 in network B is  $\frac{(\mu_1^2)}{2}$ , The fraction of nodes belong to the equal ratio connection 1:1 in network B is  $\frac{3\mu_1 - \mu_1^2}{2}$ . Here there is no relationship within or between networks, so the number of nodes with dependencies in network A is

$$N'_{A3} = ((3 - 2\mu_1)/(3 - \mu_1) + (2\mu_1)/(3 - \mu_1))\mu_2 N_B = (2\mu_2)/(3 - \mu_1)N_A \quad (8)$$

From  $N_{A1}$  to  $N'_{A3}$ , we obtain

$$N_{A1} - N'_{A3} = (1 - g_B(\mu'_1 1_2))N_{A1} \quad (9)$$

Since deleted nodes do not belong to  $N_{B2}$ ,  $N_{A1}$ , and  $N'_{A3}$ , the proportion of nodes removed from  $N_{A1}$  is equal to the same proportion of nodes removed from  $N'_{A3}$

$$N_{A1} - N'_{A3} = \left(1 - \frac{2\mu_2}{(3 - \mu_1)\mu_1}\right)N_{A1} = \left(1 - \frac{2\mu_2}{(3 - \mu_1)\mu_1}N'_{A1}\right)N'_{A3} \quad (10)$$

### 3.4 Further Fragment in Network B

The nodes in network B will fail due to the failure of the nodes in network A because of the interdependence of the coupled networks. Nodes in Network B will continue to fail because of the failure of nodes in network A in the third stage. Similar to the analysis of the second step, we can calculate the number of nodes with dependencies in the remaining nodes:

$$N'_{B4} = \frac{2}{3}N_A(\mu_3^2 + 2\mu_3(1 - \mu_3)) + \frac{1}{3}N_A\mu_3 = \frac{1}{2}(3\mu_3 + \mu_3^2)N_B \quad (11)$$

From  $N_{B2}$  to  $N'_{B4}$ , we can obtain

$$N_{B2} - N'_{B4} = \left[1 - \frac{3\mu_3 + \mu_3^2}{2\mu_2}\right]N_{B2} \quad (12)$$

The number of total removed nodes to the original network B is

$$1 - \mu'_2 + \mu'_2 \left[1 - \frac{3\mu_3 + \mu_3^2}{2\mu_2}\right] = 1 - \frac{1}{2}\mu'_1(3 - \mu_3)g_A(\mu'_3) \quad (13)$$

The number of the giant component is

$$N_{B4} = \mu'_4 g_B(\mu'_4)N_B \quad (14)$$

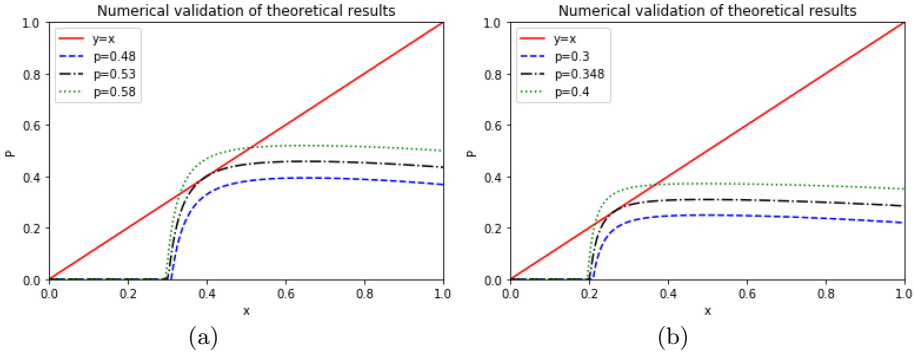


Fig. 1. Theoretical solution

According to the previous derivation process, we can obtain the following recursion relations

$$\begin{cases} \mu'_{2i} = \frac{1}{2}p(3 - \mu_{2i-1})g_A(\mu'_{2i-1}) \\ \mu'_{2i+1} = pg_B(\mu'_{2i}) \end{cases} \quad (15)$$

Now we assume that the fraction of nodes belong to the equal ratio connection 2:1 is  $q$ , so the fraction of nodes belong to the equal ratio connection 1:1 is  $1 - q$ . We can get the iterative equation in the same way.

$$\begin{cases} \mu'_{2i} = \frac{2}{2 - q}p(1 - \frac{1}{2}q\mu_{2i-1})g_A(\mu'_{2i-1}) \\ \mu'_{2i+1} = pg_B(\mu'_{2i}) \end{cases} \quad (16)$$

In the next section, we will solve Eq. 15. And then analyze Eq. 16 in detail.

## 4 Theoretical Solution and Simulation

In this section, we first solve Eq. 15 in this section. Then we verify the theoretical solution obtained through simulation experiments. In the last we will analyze the security of the coupled system.

### 4.1 Critical Threshold Solution

For the cascading failure of the coupled network, although we do not know which step the cascading failure will stopped, the network will not split again when the cascading failure stops. Thus we can get the following equations:

$$\begin{cases} \mu'_{2i} = \mu'_{2i-2} = \mu'_{2i+2} \\ \mu'_{2i+1} = \mu'_{2i-1} = \mu'_{2i+3} \end{cases} \quad (17)$$

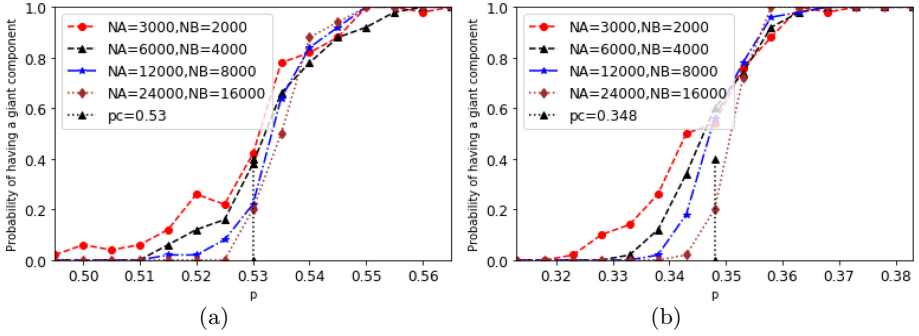


Fig. 2. Numerical validation of theoretical results

In order to facilitate the analysis of iterative formulas for cascading failure, the variable  $x, y$  is defined to satisfy the following equations:

$$\begin{cases} y = \mu'_{2i} = \mu'_{2i-2} = \mu'_{2i+2}, \\ x = \mu'_{2i+1} = \mu'_{2i-1} = \mu'_{2i+3}, \end{cases} \quad (0 \leq x, y \leq 1) \tag{18}$$

Thus, Eq. 18 can be represented by the following equation set

$$\begin{cases} y = p((xg_A(x))^3 - 3xg_A(x) + 3)g_A(x) \\ x = pg_B(y) \end{cases} \tag{19}$$

For Eq. 19, the method of mathematical analysis is difficult to solve. But we can use the method of drawing to find the approximate solution. So we use the following equations to represent Eq. 19.

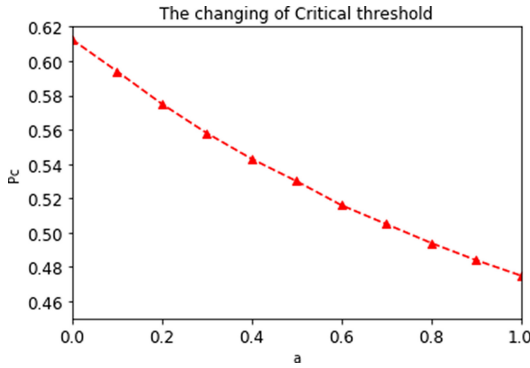
$$\begin{cases} z = x \\ z = pg_B[(\frac{1}{2}p(3 - xg_A(x))g_A(x))g_A(x)] \end{cases} \tag{20}$$

Then we can use the drawing method to solve the equations. As shown in Fig. 1. In Fig. 1, we find the critical threshold of the coupled system when  $\alpha = 4$  and  $\alpha = 6$  respectively. From the two graphs in Fig. 2, we can see that the curve and the straight line have no intersection in the interval  $(0, 1)$  when the value of  $p$  is small. As  $p$  increases, the curve and straight line will be tangency, the value of  $p$  when the two lines are tangency is the critical threshold. When  $p$  continues to increase, the curve and the line will intersect. We can get that the critical thresholds are  $pc = 0.53$  when  $\alpha = 4$ , and  $pc = 0.348$  when  $\alpha = 6$ .

### 4.2 Numerical Simulation

Next, we will verify the theoretical threshold we have obtained through simulation experiments. First we generate two ER networks and then couple the two networks together according to the model established in the second section.





**Fig. 3.** The changing of critical threshold

Then we use random deletion to represent random attacks. In the process of simulating cascade failure, the number of remaining nodes in the network will be saved to the file after each cascade failure. Then we will verify the theoretical threshold by analyzing the data in the file. In order to ensure the accuracy of the experimental results, we have repeated 50 times in each set of experiments.

In order to further verify the theoretical threshold, we take values around the 0.005 interval near the threshold for more detailed verification. In Fig. 3, the abscissa indicates the number of nodes that are not attacked in the initial stage, and the ordinate indicates the probability of existence of the largest connected component. The black arrow in Fig. 3 indicates the critical threshold. From Fig. 3(a) we can see that the curve will get closer to the critical threshold as the number of nodes increases, and the trend of the curve becomes steeper near the critical threshold. We can speculate that when the number of nodes is large enough, the curve will undergo a phase change near the critical threshold. This phenomenon once again shows that the coupled system exhibits a first-order phase transition at the critical threshold.

In Fig. 4, we use another way to indicate that the coupled system changes around the critical threshold. The ordinate represents the proportion of the remaining nodes in the two networks. The abscissa in Fig. 4 has the same meaning as the abscissa in Fig. 3. From the two graphs in Fig. 4 we can see that as the number of nodes increases, the curve approaches the critical threshold. This phenomenon further proves that the theoretical threshold we have obtained is correct. As the value of  $p$  increases, the curves in the graph will gradually overlap. This shows that as the  $p$  increasing, the number of nodes no longer affects the proportion of the remaining nodes when cascading failures stop.

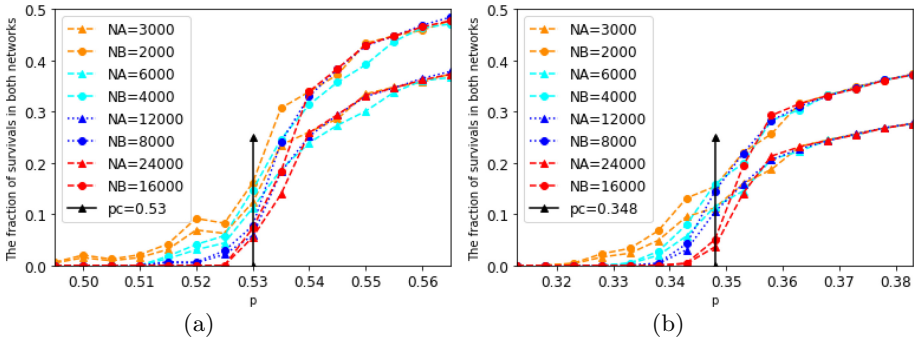


Fig. 4. The fraction of survival in both networks Corresponding

## 5 Conclusion

This paper studies the security analysis of coupled CPS system under complex connections. Our findings propose a method for analyzing complex connection relationships. In this way, we can find the critical threshold of the coupled system under complex connection relations, and we verify the correction of critical threshold by simulation. This method can be applied to some practical coupling systems. However, we have limitations in building models, and we will improve the model in the next step to make it universal. There are still many problems in the research of complex networks, and we need to explore them further.

**Acknowledgments.** This work was supported in part by the National Natural Science Foundation of China (Grant No.61902359, No.61672467 and No.61672468), in part by the Social Development Project of Zhejiang Provincial Public Technology Research (Grant No.2016C33168), in part by Zhejiang Provincial Natural Science Foundation of China (Grant No.LQ19F030010), and in part by the Opening Project of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security (Grant No.AGK2018001).

## References

1. Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. *Nature* **406**(6794), 378–382 (2000)
2. Brummitt, C.D., D’Souza, R.M., Leicht, E.A.: Suppressing cascades of load in interdependent networks. *Proc. Nat. Acad. Sci.* **109**(12), E680–E689 (2012)
3. Buldyrev, S.V., Parshani, R., Paul, G., Stanley, H.E., Havlin, S.: Catastrophic cascade of failures in interdependent networks. *Nature* **464**(7291), 1025–1028 (2010)
4. Dartmann, G., Song, H., Schmeink, A.: Big data analytics for cyber-physical systems. In: *Machine Learning for the Internet of Things*. Elsevier (2019)
5. Dey, P., Mehra, R., Kazi, F., Wagh, S., Singh, N.M.: Impact of topology on the propagation of cascading failure in power grid. *IEEE Trans. Smart Grid* **7**(4), 1970–1978 (2016)

6. Gao, J., Buldyrev, S.V., Stanley, H.E., Havlin, S.: Networks formed from interdependent networks. *Nat. Phys.* **8**(1), 40–48 (2012)
7. Hartmann, T., Fouquet, F., Klein, J., Le Traon, Y., Pelov, A., Toutain, L., Ropitault, T.: Generating realistic smart grid communication topologies based on real-data. In: 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 428–433. IEEE (2014)
8. Huang, Y., Li, B., Liu, Z., Li, J., Yiu, S.M., Baker, T., Gupta, B.B.: Thinoram: Towards practical oblivious data access in fog computing environment. *IEEE Trans. Serv. Comput.* **13**, 602–612 (2019)
9. Ke, S., Zhenxiang, H., Yijia, C.: Review on models of cascading failure in complex power grid. *Power Syst. Technol. Beijing* **29**(13), 1 (2005)
10. Li, T., Li, J., Chen, X., Liu, Z., Lou, W., Hou, T.: NPMML: A framework for non-interactive privacy-preserving multi-party machine learning. *IEEE Trans. Dependable Secure Comput.* (2020). <https://doi.org/10.1109/TDSC.2020.2971598>
11. Li, W., Bashan, A., Buldyrev, S.V., Stanley, H.E., Havlin, S.: Cascading failures in interdependent lattice networks: The critical role of the length of dependency links. *Phys. Rev. Lett.* **108**(22), 228702 (2012)
12. Liu, Z., Li, B., Huang, Y., Li, J., Xiang, Y., Pedrycz, W.: Newmcos: Towards a practical multi-cloud oblivious storage scheme. *IEEE Trans. Knowl. Data Eng.* **32**(4), 714–727 (2019)
13. Manik, D., Rohden, M., Ronellenfitsch, H., Zhang, X., Hallerberg, S., Witthaut, D., Timme, M.: Network susceptibilities: Theory and applications. *Phys. Rev. E* **95**(1), 012319 (2017)
14. Newman, M.: *Networks*. Oxford University Press (2018)
15. Rosato, V., Issacharoff, L., Tiriticco, F., Meloni, S., Porcellinis, S., Setola, R.: Modelling interdependent infrastructures using interacting dynamical models. *Int. J. Crit. Infrastruct.* **4**(1–2), 63–79 (2008)
16. Sun, K.: WAMS-based controlled system separation to mitigate cascading failures in smart grid. In: Stoustrup, J., Annaswamy, A., Chakraborty, A., Qu, Z. (eds.) *Smart Grid Control. Power Electronics and Power Systems*. Springer, Cham (2019)
17. Toft, M.B., Schuitema, G., Thøgersen, J.: Responsible technology acceptance: Model development and application to consumer acceptance of smart grid technology. *Appl. Energy* **134**, 392–400 (2014)
18. Wang, T., Liang, Y., Yang, Y., Xu, G., Peng, H., Liu, A., Jia, W.: An intelligent edge-computing-based method to counter coupling problems in cyber-physical systems. *IEEE Netw.* **34**(3), 16–22 (2020)
19. Xu, G., Liu, J., Lu, Y., Zeng, X., Zhang, Y., Li, X.: A novel efficient maka protocol with desynchronization for anonymous roaming service in global mobility networks. *J. Netw. Comput. Appl.* **107**, 83–92 (2018)
20. Xu, G., Zhang, Y., Sangaiyah, A.K., Li, X., Castiglione, A., Zheng, X.: CSP-E2: An abuse-free contract signing protocol with low-storage TTP for energy-efficient electronic transaction ecosystems. *Inf. Sci.* **476**, 505–515 (2019)
21. Zhu, Y., Yan, J., Tang, Y., Sun, Y.L., He, H.: Joint substation-transmission line vulnerability assessment against the smart grid. *IEEE Trans. Inf. Forensics Secur.* **10**(5), 1010–1024 (2015)



# Automatic Classification Analysis of Tibetan Folk Music Based on Adaboost Algorithm

Ma Ying, Li Kaiyong<sup>(✉)</sup>, and Hou Jiayu

Qinghai University for Nationalities, Xining 810007, China  
2000018@qhmu.edu.com

**Abstract.** Automatic classification of music is a necessary prerequisite for rapid and accurate retrieval of music resources, and its potential application needs are enormous; However, due to the ambiguity of music classification and the diversity of music signals, the study of automatic music classification is still in its infancy; This paper proposes the application of machine learning Adaboost algorithm in the automatic classification of Tibetan folk music, extracts the characteristics of music signals, and uses Adaboost algorithm and decision tree to perform predictive data analysis on Tibetan folk music signals; The results show that the Adaboost algorithm has higher accuracy and better robustness in the classification of Tibetan folk music signals, the accuracy of the Adaboost algorithm is improved by 12.5% compared to the decision tree.

**Keywords:** Machine learning · Adaboost algorithm · Decision tree · Tibetan folk music · Automatic classification

## 1 Introduction

In recent years, the development of computer technology and information processing technology has broadened the way for the classification of music signals. The application of machine learning technology in music signal classification and recognition is also becoming more and more extensive. In terms of audio signal processing, there have been many studies using various machine learning methods from various angles to obtain rich research results, and analyze and study the characteristics of speech signals [1], application of music signal classification and recognition algorithms [2–4], study on classification of speech signals [5], face detection [6, 7], vehicle detection [8] and speech recognition [9], Tibetan Text classification [10] etc., but no reports have been seen in the automatic classification of Tibetan folk music. In this paper, three different types of Tibetan folk music signals will be extracted by speech signal processing to form feature parameter training sets, using machine learning Adaboost algorithm and decision tree to automatically classify the three different kinds of Tibetan folk music signals, analyze the accuracy of Adaboost algorithm in the classification of Tibetan folk music signals.

Fund Projects: Applied basic research project of Qinghai Science and Technology Department (2020-ZJ-709), Key R & D and Transformation Projects of Qinghai Science and Technology Department (2019-GX-170), Multi-source Data Fusion and Applied Research Innovation Team.

© Springer Nature Singapore Pte Ltd. 2020

Y. Xiang et al. (Eds.): SocialSec 2020, CCIS 1298, pp. 379–386, 2020.

[https://doi.org/10.1007/978-981-15-9031-3\\_33](https://doi.org/10.1007/978-981-15-9031-3_33)

## 2 Feature Extraction of Tibetan Folk Music

### 2.1 Pitch Period Extraction

The article performs feature extraction on three types of Tibetan folk music signals, playing and singing, folk songs and Aze, extracts two characteristic parameters of pitch period and formant frequency, a total of 8 samples form the training set of characteristic parameters of music signals.

Pitch period is one of the most important characteristic parameters of speech signal, and it is an important characterization of speech excitation source. Pitch period information is widely used in many fields of signal processing. There are many pitch detection methods: autocorrelation function (ACF) method, average amplitude difference function (AMDF) method, cepstrum method, linear prediction method, etc. The article uses the ACF method to extract the pitch period.

Let the time sequence of the speech signal be  $x(n)$ , after processing the windowed frame, the  $i$  frame speech signal is  $x_i(m)$ , where the subscript  $i$  indicates the  $i$  frame, let each frame length be  $N$ . The short-term autocorrelation function of  $x_i(m)$  is defined as

$$R_i(k) = \sum_{m=1}^{N-k} x_i(m)x_i(m+k) \quad ((1))$$

in the formula,  $k$  is the amount of time delay.

The simulation diagram of the pitch extraction of three Tibetan folk music signals is shown in Fig. 1, 2 and 3.

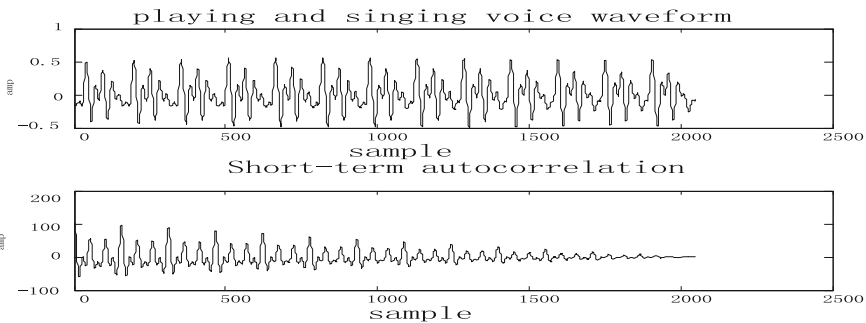
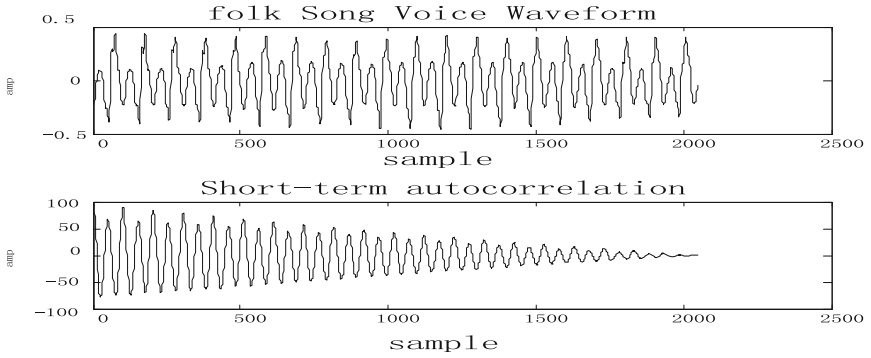


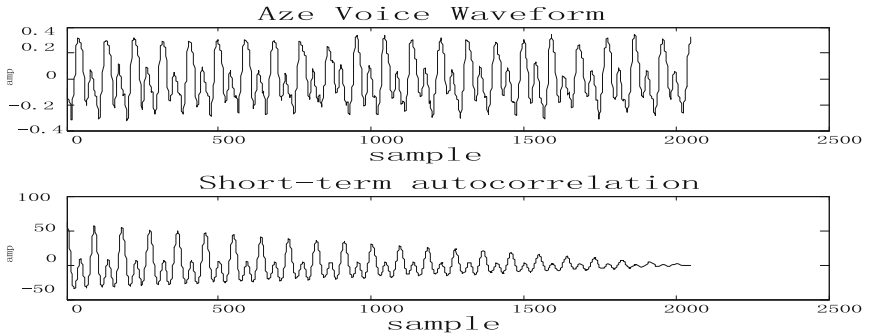
Fig. 1. “Playing and singing” pitch cycle

### 2.2 Formant Frequency Extraction

The formant parameter information is included in the envelope of the speech spectrum, so the formant parameter extraction is to estimate the speech spectrum envelope and use the maximum value in the spectrum envelope as the formant. Commonly used formant estimation methods include cepstrum method and LPC method, etc. The article uses LPC root finding method.



**Fig. 2.** “Folk Song” pitch cycle



**Fig. 3.** “Aze” pitch cycle

The speech channel transfer function is

$$H(z) = \frac{G}{1 - \sum_{i=1}^p a_i z^{-i}} \quad (2)$$

Three formant frequencies are extracted from the three Tibetan folk music signals. The formant frequencies are shown in Fig. 4, 5 and 6.

The sample set of the characteristic parameters of formant frequency and pitch period of three kinds of Tibetan folk music are shown in Table 1.

There are a total of 8 samples in the parameter sample set, and the three formant frequencies  $f_1$ ,  $f_2$ ,  $f_3$  and a pitch period of 8 samples are extracted.

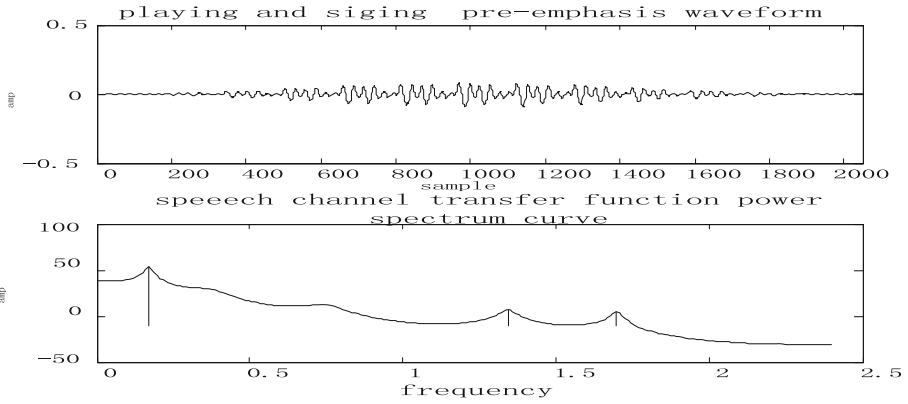


Fig. 4. "Playing and singing" formant frequency

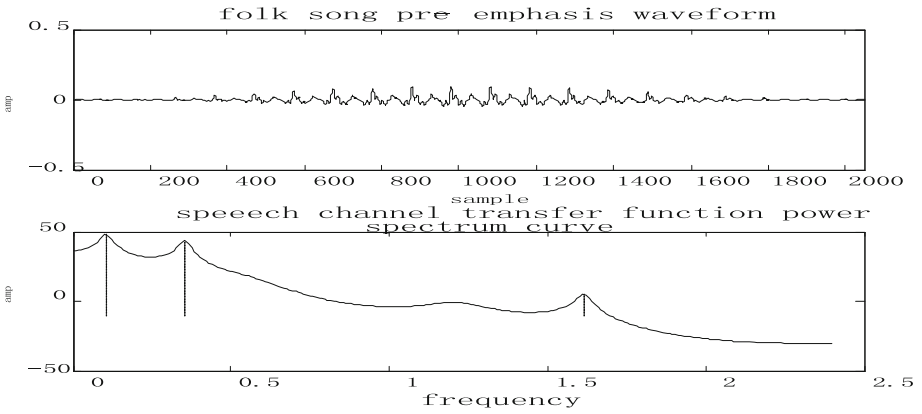


Fig. 5. "Folk song" formant frequency

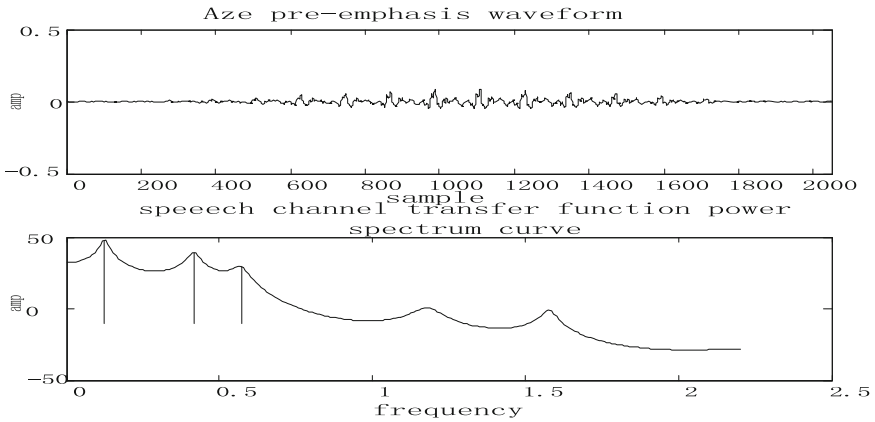


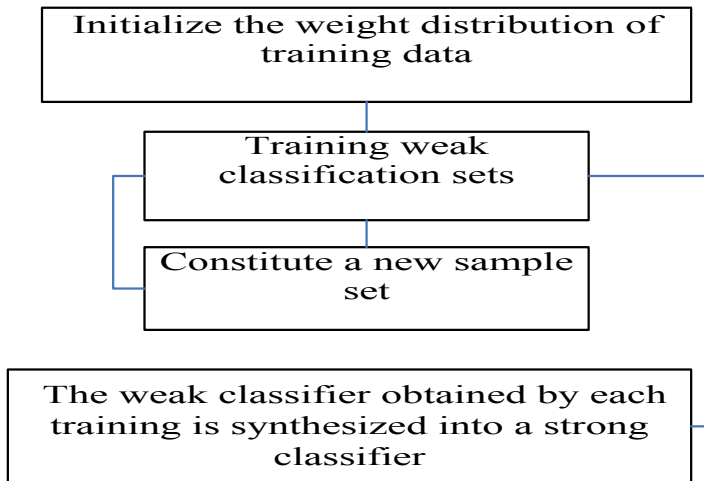
Fig. 6. "Aze" formant frequency

**Table 1.** Characteristic parameter sample set

Sample	Formant f1	Formant f2	Formant f3	Pitch period
1	1704.73	13430.55	16981.95	3.25
2	2031.36	4250.93	16090.06	3.17
3	1521.37	4232.06	15854.55	2.54
4	1044.97	3550.22	16154.64	2.13
5	1136.80	3840.69	15246.69	1.92
6	1582.46	4100.06	15884.06	2.06
7	1265.95	4164.02	5728.21	2.74
8	1102.43	3400.07	5417.27	2.06

### 3 Basic Principles of Adaboost Algorithm

Adaboost algorithm is to get a series of weak classifiers from the sample training set, and then combine these weak classifiers into a strong classifier [11]. The specific algorithm principle is shown in Fig. 7.

**Fig. 7.** Basic principles of Adaboost algorithm

### 4 Basic Principles of Decision Tree Algorithm

Decision tree is a process of classifying data through a series of rules, the generation process of a decision tree mainly includes feature selection, decision tree generation



and pruning; The decision tree based on the Tibetan folk music signal feature parameter training set is shown in Fig. 8.

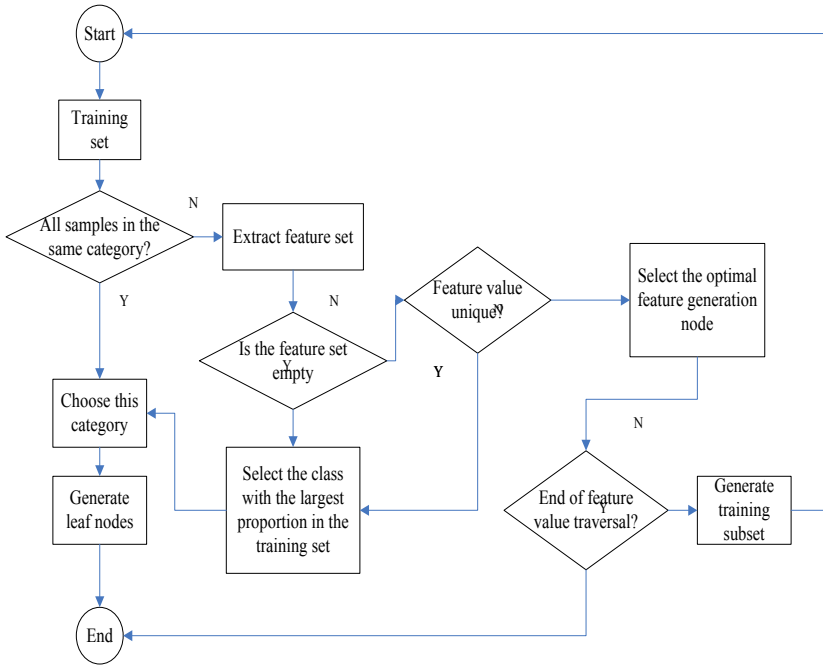


Fig. 8. Principle of training set decision algorithm

### 5 Predictability Data Analysis

By building an algorithm scene model, training and automatic classification of existing samples, analyze the classifier to check the index confusion matrix and accuracy, precision, recall rate and comprehensive evaluation index F1-Score, normally, we expect that the greater the accuracy and recall rate, the better, and the closer the accuracy to 1, the better, when the precision and the recall rate conflict with each other, the classification of the sample set can also be predicted and analyzed through the comprehensive evaluation index F1-Score. The algorithm scene model is shown in Fig. 9.

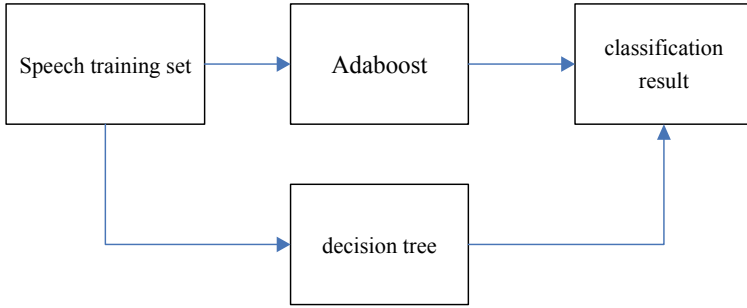
Execute the algorithm, get the confusion matrix as shown in Fig. 10, and analyze the classifier to check the index.

Precision, expressed by P, calculating accuracy

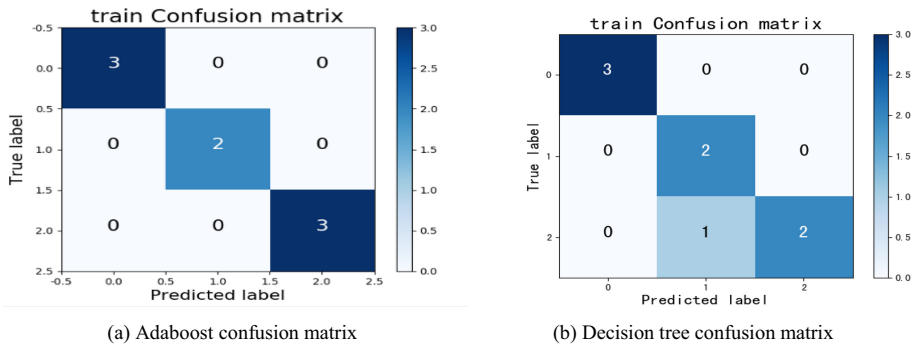
$$P = \frac{TP}{TP + FP}$$

Recall, represented by R, calculate the recall rate

$$R = \frac{TP}{TP + FN}$$



**Fig. 9.** Scene model



**Fig. 10.** Confusion matrix

$$F1-score = 2 \times \frac{P \times R}{P + R}$$

True label in the confusion matrix represents the classification result of the real situation, and Predicted label represents the classification result predicted by the model, the classification of the model prediction pairs is on the diagonal of the confusion matrix, and the accuracy is that the number of samples of the model prediction pair is larger than the total number of samples. Through the obtained Adaboost confusion matrix analysis, the calculation accuracy is 1, the precision is 1, the recall rate is 1, the comprehensive evaluation index F1-Score is 1, and the accuracy of the decision tree confusion matrix analysis is 0.875, the indexes of the training algorithm models of the two algorithms are shown in Table 2, it can be seen that the classification accuracy of the Adaboost algorithm is high.

**Table 2.** Training set model indicators

Detection type	Precision	Recall	F1-Score
Adaboost 1	1	1	1
Adaboost 2	1	1	1
Adaboost 3	1	1	1
Decision tree 1	1	1	1
Decision tree 2	0.6666	1	0.8
Decision tree 3	1	0.6666	0.8

## 6 Summary

Automatic classification has important application value in the field of speech signal processing, and the classification of music signals is today's research boom. This paper extracts the characteristic parameters pitch and formant frequency of three Tibetan folk music signals, forming 8 sample training sets, the machine learning Adaboost algorithm and decision tree algorithm are used to classify the sample set, and the classification results are observed through predictive data analysis, it can be concluded that the Adaboost algorithm is effective for the classification of Tibetan folk music signals, and has high accuracy and good robustness.

## References

1. Liu, Y.: Research on content-based music feature extraction and classification technology. Beijing University of Posts and Telecommunications (2016)
2. Zhaoxing, F., Wei, S., Shuyong, G., Fuzhen, Y.: Application of BP algorithm of PCA-improved RPROP method in music signal classification. *Meas. Control Technol.* **38**(07), 84–88 (2019)
3. Yang, W.: Research on signal classification algorithm in speech/audio hybrid encoder. Wuhan University (2018)
4. Wang, F.: Research on deep learning-based music genres and traditional Chinese musical instrument recognition classification. Nanjing University of Science and Technology (2017)
5. Xing, Y., Liu, L., Cheng, S., et al.: Multi-classification of speech emotion based on Fisher criterion and Adaboost. *Comput. Digit. Eng.* **11**(46), 2197–2229 (2018)
6. Chen, W.: Analysis of a pedestrian detection mode based on AdaBoost. Shandong Industrial Technology (2014)
7. Yan, B.: Research on face detection algorithm based on AdaBoost. *Graph. Image* **3**, 75–77 (2019)
8. Li, X., Zhao, W., Dou, X., et al.: Vehicle detection algorithm based on improved Adaboost + Haar. *Meas. Control Technol.* **38**(2), 42–45 (2019)
9. Yang, S., Zhu, H., Feng, T., et al.: Speech recognition algorithm based on Kaldi. *Comput. Knowl. Technol.* **15**(2), 163–165 (2019)
10. Jia, H.: Research and implementation of tibetan text classification based on AdaBoost Model. Tibet University (2019)
11. Tian, M.: Research on detection methods of Android malicious applications based on machine learning. Beijing Jiaotong University (2019)



# On the Security of a Certificateless Public Verification Scheme for Cloud-Based Cyber-Physical-Social Systems

Jing Wang<sup>1</sup>, Hongjie Zhang<sup>2</sup>, Lishong Shao<sup>3</sup>, Li Li<sup>1</sup>, and Min Luo<sup>1</sup>(✉)

<sup>1</sup> School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China  
mluo@whu.edu.cn

<sup>2</sup> State Grid Ningxia Electric Power Co. Ltd, Yinchuan 750001, China

<sup>3</sup> State Grid Electric Power Research Institute, Nanjing 210003, China

**Abstract.** With the popularity of the cyber-physical-social system, increasing individuals would like to share their data from both cyberspace and physical space with others. This results in more and more data needed to be stored locally. Due to the high price of storing and maintaining those data, the user would like to resort to the cloud storage. Thus, the concept of cloud-based cyber-physical-social (CBCPS) systems are putted forward. In CBCPS systems, the user loses his/her physical control on the data after they are uploaded to the cloud. How to guarantee the data integrity becomes a challenging issue. Recently, Zhang et al. proposed a certificateless public verification (CLPV) scheme to ensure the data integrity in CBCPS systems. Zhang et al. claimed that their CLPV scheme is secure against various attacks and is provably secure in the random oracle model. However, in this paper, we propose a universal attack against Zhang et al.'s CLPV scheme to demonstrate that a malicious cloud server can modify even delete the user's data without being found by the auditor or the user. The security analysis shows that Zhang et al.'s CLPV scheme is not secure enough for the cloud-based cyber-physical-social systems.

**Keywords:** Cloud storage · Certificateless cryptography · Data integrity · Malicious cloud server

## 1 Introduction

The cyber-physical-social (CPS) system is a new promising network system and has attracted much attention from both industrial and academic [1]. By integrating the cyber space, the physical elements and the traditional social networks seamlessly, the CPS system is able to provide many powerful tools (such as the social computing and the social operation) to solve the challenges existing in cyber-physical-social interactions and human-centric technologies development [2].

With the popularity of mobile devices, increasing users in the CPS system would like to store and share data about current status (such as the location, the track and the behavior) collected by their mobile devices [3]. As the evolution of the modern information technology, huge amount of data is generated exponentially in the CPS system. According to the estimation of International Data Corporation (IDC) and EMC, the size of those data will grow from 2,837 exabytes (EB) in 2012 to 40,000 EB in 2020 [4].

Due to the fast growth of data, how to store them becomes an issue that needs to be addressed urgently. In traditional approach, the user needs some hardware devices to build systems and manage the operation, maintenance and update of the system. According to Hasan et al.'s evolution, the cost of managing the storage system is about 5–7 times that of buying hardware devices, i.e. the cost of managing the storage system is seventy-four percent [5]. Due to the huge cost, the traditional approach is uneconomical for the storage of the huge data in the CPSS.

Compared with traditional approach for the storage, the cloud storage has many advantages (such as on-demand self-service, location independent resource polling and transference of risk) [5]. Therefore, the cloud storage can address weaknesses in traditional approach and is suitable for the storage of the data in the CPS system. Due to the integration of the cloud and the CPSS, the cloud-based cyber-physical-social (CBCPS) system was putted forward. After uploading the data to the cloud, the user loses his/her physical control on the data. How to check the data integrity becomes an important challenging issue.

The digital signature or the message authentication code is used in the traditional approaches for ensuring the data integrity. However, those approaches are not suitable for checking the data integrity in the CBCPS system because the verifier has to download all data from the cloud. To address the problem, Ateniese et al. [6] putted forward the concept of the provable data possession (PDP) scheme based on the Public Key Infrastructure (PKI). Later, Ateniese et al. [7] proposed a dynamic PDP scheme. However, Ateniese et al.'s scheme [7] is not able to provide the insert operation. After their pioneering work, a lot of PDP schemes [8] based on the PKI were presented to improve performance or enhance security.

In the above PKI-based PDP schemes, a trusted third-party generates a certificate for each user to establish relation between his/her identity and public key [9]. With the fast growth of the users' number, it is increasingly difficult to effectively manage those certificates [10]. To address the problem, Zhao et al. [11] proposed an identity-based PDP scheme, where the user's identity is his/her public key. After that, several other identity-based PDP schemes [12] were proposed for different applications.

Although the above identity-based PDP schemes are able to address the certificate management problem, they suffer from the key escrow problem, i.e., the key generation center (KGC) generates and knows all users' private keys. The system will be broken totally when the adversary controls the KGC. To overcome the weakness, Wang et al. [13] putted forward the concept of the certificateless provable data possession (CLPDP) scheme. However, Wang

et al.'s CLPDP scheme is unable to provide the data privacy. To enhance security, He et al. [14] proposed an improved CLPDP scheme. Recently, Zhang et al. [15] proposed a certificateless public verification (CLPV) scheme to ensure the data integrity in CBCPS systems. They claimed that their CLPV scheme is able to withstand various attacks. However, this paper will propose a universal attack against their CLPV scheme, i.e., a malicious cloud server is able to modify or delete the data without being found by the third-party auditor or the user. The security analysis shows that Zhang et al.'s CLPV scheme is not secure for practical applications.

The organization of the paper is listed as below. Section 2 gives a brief review of Zhang et al.'s CLPV scheme. Section 3 presets our attack against Zhang et al.'s CLPV scheme. Section 4 analyzes the reason that Zhang et al.'s CLPV suffers from the attack. Section 5 presents some conclusions of the paper.

## 2 Review of Zhang et al.'s CLPV Scheme

We give a brief review of Zhang et al.'s CLPA scheme [15]. Zhang et al.'s CLPA scheme consists of four algorithms, i.e., *Setup*, *Store*, *ChalGen*, *ProGen* and *CheckLog*. The details of the above algorithms are presented below.

*Setup*. This algorithm is executed by the KGC and the user  $U$  to produce the system parameters and the user's private key. Three steps are involved in this algorithm.

Step 1). In this step, KGC produces the system parameters by running the below processes.

- Given a security parameter  $l$ , KGC chooses two groups  $G_1, G_2$  and a bilinear pairing  $e : G_1 \times G_1 \rightarrow G_2$ . KGC also selects a generator  $P$  with a prime order  $q$  of the group  $G_1$ .
- KGC randomly picks  $\lambda \in Z_q^*$  as the system private key and computes the system public key  $P_{pub} = \lambda \cdot P$ .
- KGC picks five secure hash functions  $h, H_1, H_2, H_3$  and  $H_4$ , where  $h : \{0, 1\}^* \rightarrow Z_q^*$ ,  $H_i : \{0, 1\}^* \rightarrow G_1$  and  $i = 1, 2, 3, 4$ .
- KGC publishes the system parameters  $\{G_1, G_2, e, P, q, P_{pub}, h, H_1, H_2, H_3, H_4\}$  and saves  $\lambda$  secretly.

Step 2). In this step, KGC produces  $U$ 's partial private key by running the below processes.

- KGC computes  $Q_{U,0} = H_1(ID_U, 0)$  and  $Q_{U,1} = H_1(ID_U, 1)$ .
- KGC computes  $D_{U,0} = \lambda \cdot Q_{U,0}$  and  $D_{U,1} = \lambda \cdot Q_{U,1}$ .
- KGC sends the partial private key  $D_U = \{D_{U,0}, D_{U,1}\}$  to  $U$  secretly.

Step 3). In this step,  $U$  produces his/her secret value and public key by running the below processes.

- $U$  randomly picks  $x_U \in Z_q^*$  as his/her secret value.
- $U$  computes his/her public key  $X_U = x_U \cdot P$ .

After executing above steps,  $U$ 's signing key and verification key are  $sk_U = \{x_U, D_{U,0}, D_{U,1}\}$  and  $vk_U = \{X_U\}$  respectively.

*Store.* This algorithm is executed by the user  $U$  and the cloud server  $CS$  to produce signatures of a file's blocks and store the file in the cloud. Given a file  $F$ ,  $U$  first divides it into  $n$  blocks, i.e.,  $F = \{m_i\}_{i=1}^n$ .  $U$  randomly picks the file name  $name$  and computes the file's tag  $\tau = name || Sign_{sk_U}(name)$ , where  $Sign(\cdot)$  denotes a secure certificateless signature scheme.  $U$  and  $CS$  run the below three steps.

Step 1).  $U$  produces  $m_i$ 's signature by running the below processes, where  $i = 1, 2, \dots, n$ .

- $U$  randomly produces a one-time-use number  $\Delta_F$  for the file  $F$ .
- $U$  randomly produces a number  $r_i \in Z_q^*$  and computes  $R_i = r_i \cdot P$ , where  $i = 1, 2, \dots, n$ .
- $U$  calculates  $T = H_2(\Delta_F)$ ,  $V = H_3(\Delta_F)$  and  $W = H_4(\Delta_F)$ .
- $U$  calculates  $S_i = m_i \cdot (D_{U,0} + x_U \cdot V) + \tau_i \cdot (D_{U,1} + x_U \cdot W) + r_i \cdot T$ , where  $\tau_i = h(i || name)$ .
- $U$  outputs  $\delta_i = \{R_i, S_i\}$  as  $m_i$ 's signature.

Step 2).  $U$  sends  $\hat{F} = \{F = \{m_i\}_{i=1}^n, \phi = \{\delta_i\}_{i=1}^n, \tau, \Delta_F\}$  to  $CS$ .

Step 3).  $CS$  computes  $Q_{U,0} = H_1(ID_U, 0)$  and  $Q_{U,1} = H_1(ID_U, 1)$  and checks if the equation  $e(\sum_{i=1}^n S_i, P) = e(\sum_{i=1}^n (m_i \cdot Q_{U,0} + \tau_i \cdot Q_{U,1}), P_{pub}) \cdot e(\sum_{i=1}^n (m_i \cdot V) + \sum_{i=1}^n (\tau_i \cdot W), X_U) \cdot e(T, \sum_{i=1}^n R_i)$ , where  $\tau_i = h(i || name)$ . If the equation holds,  $CS$  stores the received data; otherwise,  $CS$  rejects the request.

*ChalGen.* This algorithm is executed by a third-party auditor ( $TPA$ ) to produce a challenging message through running the below five steps.

Step 1).  $TPA$  acquires a hash value  $Bl_t$  of the Bitcoin block [16].

Step 2).  $TPA$  takes  $Bl_t$  as the seed of the pseudorandom bit generator to produce a random number  $\theta$ .

Step 3).  $TPA$  randomly chooses a subset  $I \in \{1, 2, \dots, n\}$  based on the value of  $\theta$ .

Step 4).  $TPA$  randomly chooses a small number  $v_i$  for each  $i \in I$ .

Step 5).  $TPA$  outputs  $\{(i, v_i)_{i \in I}\}$  as the challenging message and sends it to  $CS$ .

*ProGen.* This algorithm is executed by the cloud server  $CS$  to produce a proof of storing the data correctly through running the below two steps.

Step 1).  $CS$  computes  $S = \sum_{i \in I} v_i \cdot S_i$ ,  $R = \sum_{i \in I} v_i \cdot R_i$  and  $v = \sum_{i \in I} v_i \cdot m_i$ .

Step 2).  $CS$  outputs the proof  $\{S, R, v, \Delta\}$  and sends it to  $TPA$ .

*VerPro.* This algorithm is executed by a third-party auditor  $TPA$  to check the correctness of the received proof  $\{S, R, v, \Delta_F\}$  through running the below three steps.

Step 1).  $TPA$  extracts the file's tag  $\tau$  and verifies its validity using  $U$ 's public key.

Step 2).  $TPA$  checks if the equation  $e(S, P) = e(v \cdot Q_{U,0} + \sum_{i \in I} v_i \cdot \tau_i \cdot Q_{U,1}, P_{pub}) \cdot e(v \cdot H_3(\Delta_F) + \sum_{i \in I} v_i \cdot \tau_i \cdot H_4(\Delta_F), X_U) \cdot e(H_2(\Delta_F), R)$  holds, where  $\tau_i = h(i || name)$ . If it does not hold,  $TPA$  confirms the file is not stored correctly and outputs "Reject"; otherwise,  $TPA$  outputs "Accept".

Step 3).  $TPA$  stores  $\{Bl_t, S, R, v, \Delta_F\}$  into the log file  $\Lambda$ .

*CheckLog*. This algorithm is executed by the user  $U$  and third-party auditor  $TPA$  to checks the validity of the log file  $\Lambda$  through running the below four steps.

Step 1).  $U$  randomly selects a subset  $B$  according to Bitcoin blocks.

Step 2).  $U$  produces a group of challenging messages  $I^B = \{\{i^{(1)}, v_i^{(1)}\}_{i \in I^1}, \dots, \{\{i^{(b)}, v_i^{(b)}\}_{i \in I^b}\}$  and sends them to  $TPA$ , where  $b$  denotes the size of the subset  $B$ .

Step 3).  $TPA$  computes  $S^{(B)} = \sum_{j \in B} S^j$ ,  $R^{(B)} = \sum_{j \in B} R^j$  and  $v^{(B)} = \sum_{j \in B} v^j$ .  $TPA$  sends the proof  $\{S^B, R^B, v^B, \Delta_F\}$  to  $U$ .

Step 4).  $U$  checks if the equation  $e(S^B, P) = e(v^B \cdot Q_{U,0} + v \cdot Q_{U,1}, P_{pub}) \cdot e(v^B \cdot H_3(\Delta_F) + v \cdot H_4(\Delta_F), X_U) \cdot e(H_2(\Delta_F), R^B)$  holds. If it does not hold,  $U$  confirms the file stored in the cloud is destroyed and outputs “Reject”; otherwise,  $U$  outputs “Accept”.

### 3 Security Analysis

According to the security model for CLPV schemes, there are two types of adversaries against a CLPV scheme. The type I adversary  $\mathcal{A}_1$  is able to replace any user’s public key with the one chosen by himself/herself, but he/she cannot access the system private key. The Type II adversary  $\mathcal{A}_2$  cannot replace the user’s public key, but he/she is able to access the system private key.

Zhang et al.’s claimed that their CLPV scheme is secure against the above two types of adversaries. In this section, we will show that a malicious cloud server, who has no ability to replace the user’s public key or to access the system private key, is able to modify even delete the user’s data without being found by the third-party auditor or the user.

According to description of Zhang et al.’s CLPV scheme, the cloud server  $CS$  use the signature  $\delta_i = \{R_i, S_i\}$  of the  $i$ th block  $m_i$  to produce the valid proof of the data integrity. To show a malicious cloud server  $CS$  is able to produce a valid proof after modifying or deleting the data, we just need to show it is able to forge a signature  $\delta^* = \{R^*, S^*\}$  of any message  $m^*$ . Suppose  $CS$  wants to change the  $k$ th block  $m_k$  into  $m_k^*$ . It first forges the signature of  $m_k^*$  through the below processes.

- 1).  $CS$  chooses two blocks  $m_i, m_j$  and their corresponding signatures  $\delta_i = \{R_i, S_i\}, \delta_j = \{R_j, S_j\}$ .
- 2).  $CS$  extracts two numbers  $\alpha_i$  and  $\alpha_j$  by solving the below system of linear equations, where  $\tau_i = h(i||name)$ ,  $\tau_j = h(j||name)$  and  $\tau_k = h(k||name)$ .

$$\begin{cases} m_i \cdot \alpha_i + m_j \cdot \alpha_j = m_k^* \\ \tau_i \cdot \alpha_i + \tau_j \cdot \alpha_j = \tau_k \end{cases} \quad (1)$$



According to the Cramer’s Rule [17], the solution of the system of linear equations is

$$\begin{cases} \alpha_i = \frac{\begin{vmatrix} m_k^* & m_j \\ \tau_k & \tau_j \end{vmatrix}}{\begin{vmatrix} m_i & m_j \\ \tau_i & \tau_j \end{vmatrix}} \\ \alpha_j = \frac{\begin{vmatrix} m_i & m_k^* \\ \tau_i & \tau_k \end{vmatrix}}{\begin{vmatrix} m_i & m_j \\ \tau_i & \tau_j \end{vmatrix}} \end{cases} \tag{2}$$

when  $\Delta = \begin{vmatrix} m_i & m_j \\ \tau_i & \tau_j \end{vmatrix} \neq 0$ . *CS* will choose other two blocks  $m_{i'}, m_{j'}$  if  $\Delta$  is zero.

- 3). *CS* computes  $S_k^* = \alpha_i \cdot S_i + \alpha_j \cdot S_j$  and  $R_k^* = \alpha_i \cdot R_i + \alpha_j \cdot R_j$ .
- 4). *CS* outputs  $\delta_k^* = \{R_k^*, S_k^*\}$

Due to  $S_i = m_i \cdot (D_{U,0} + x_U \cdot V) + \tau_i \cdot (D_{U,1} + x_U \cdot W) + r_i \cdot T$ ,  $S_j = m_j \cdot (D_{U,0} + x_U \cdot V) + \tau_j \cdot (D_{U,1} + x_U \cdot W) + r_j \cdot T$ ,  $R_i = r_i \cdot P$  and  $R_j = r_j \cdot P$ , we can get

$$\begin{aligned} S_k^* &= \alpha_i \cdot S_i + \alpha_j \cdot S_j \\ &= \alpha_i \cdot [m_i \cdot (D_{U,0} + x_U \cdot V) \\ &\quad + \tau_i \cdot (D_{U,1} + x_U \cdot W) + r_i \cdot T] \\ &\quad + \alpha_j \cdot [m_j \cdot (D_{U,0} + x_U \cdot V) \\ &\quad + \tau_j \cdot (D_{U,1} + x_U \cdot W) + r_j \cdot T] \\ &= (\alpha_i \cdot m_i + \alpha_j \cdot m_j) \cdot (D_{U,0} + x_U \cdot V) \\ &\quad + (\alpha_i \cdot \tau_i + \alpha_j \cdot \tau_j) \cdot (D_{U,1} + x_U \cdot W) \\ &\quad + (\alpha_i \cdot r_i + \alpha_j \cdot r_j) \cdot T \\ &= m_k^* \cdot (D_{U,0} + x_U \cdot V) \\ &\quad + \tau_k \cdot (D_{U,1} + x_U \cdot W) \\ &\quad + (\alpha_i \cdot r_i + \alpha_j \cdot r_j) \cdot T \end{aligned} \tag{3}$$

$$\begin{aligned} R_k^* &= \alpha_i \cdot R_i + \alpha_j \cdot R_j \\ &= \alpha_i \cdot r_i \cdot P + \alpha_j \cdot r_j \cdot P \\ &= (\alpha_i \cdot r_i + \alpha_j \cdot r_j) \cdot P \end{aligned} \tag{4}$$

and

$$\begin{aligned} e(S_k^*, P) &= e(m_k^* \cdot (D_{U,0} + x_U \cdot V) \\ &\quad + \tau_k \cdot (D_{U,1} + x_U \cdot W) + (\alpha_i \cdot r_i + \alpha_j \cdot r_j) \cdot T, P) \\ &= e(m_k^* \cdot D_{U,0} + \tau_k \cdot D_{U,1}, P) \cdot \\ &\quad e(m_k^* \cdot x_U \cdot V + \tau_k \cdot x_U \cdot W, P) \cdot \\ &\quad e((\alpha_i \cdot r_i + \alpha_j \cdot r_j) \cdot T, P) \\ &= e(m_k^* \cdot Q_{U,0} + \tau_k \cdot Q_{U,1}, \lambda \cdot P) \cdot \\ &\quad e(m_k^* \cdot V + \tau_k \cdot W, x_U \cdot P) \cdot \\ &\quad e(T, (\alpha_i \cdot r_i + \alpha_j \cdot r_j) \cdot P) \\ &= e(m_k^* \cdot Q_{U,0} + \tau_k \cdot Q_{U,1}, P_{pub}) \cdot \\ &\quad e(m_k^* \cdot V + \tau_k \cdot W, X_U) \cdot e(T, R_k^*) \end{aligned} \tag{5}$$

Accord to the above equation, we know that  $\delta^* = \{R^*, S^*\}$  is a valid signature of the message  $m_k^*$ .  $CS$  replaces  $m_k$  and  $\delta_k = \{R_k, S_k\}$  with  $m_k^*$  and  $\delta_k^* = \{R_k^*, S_k^*\}$  respectively. Due to the validity of  $\delta_k^*$ ,  $CS$  is able to produce a valid proof when receiving any challenging message. Therefore, the malicious cloud server  $CS$  is able to modify the user's data without being found by the third-party auditor or the user.

To save space,  $CS$  may want to delete the file without being found by the third-party auditor or the user. It is able to achieve such a goal through the below processes.

- 1).  $CS$  chooses a constant value  $m^*$ . For example,  $CS$  sets  $m^* \leftarrow 0$ .
- 2).  $CS$  sets  $m_i^* = m^*$  and produces a valid signature  $\delta_i^* = \{R_i^*, S_i^*\}$  of  $m_i^*$ , where  $i = 1, 2, \dots, n$ .
- 3).  $CS$  remembers  $m^*$ , replaces  $\delta_i = \{R_i, S_i\}$  with  $\delta_i^* = \{R_i^*, S_i^*\}$  and deletes the file  $F = \{m_i\}_{i=1}^n$ .

Because  $\delta_i^* = \{R_i^*, S_i^*\}$  is a valid signature of the message  $m_i^* = m^*$ ,  $CS$  can use  $m^*$  and  $\delta_i^*$  to produce a valid proof when receiving any challenging message. Therefore, the malicious cloud server  $CS$  is able to delete the user's data without being found by the third-party auditor or the user.

## 4 Discussions

According to the process of generating  $m_i$ 's signature  $\delta_i = \{R_i, S_i\}$ , we know that  $S_i$  is a linear combination of  $(D_{U,0} + x_U \cdot V)$ ,  $(D_{U,1} + x_U \cdot W)$  and  $T$ , i.e., no direct relation exists among their coefficients, where  $S_i = m_i \cdot (D_{U,0} + x_U \cdot V) + \tau_i \cdot (D_{U,1} + x_U \cdot W) + r_i \cdot T$ ,  $R_i = r_i \cdot P$  and  $\tau_i = h(i||name)$ . This is the reason that Zhang et al.'s CLPV scheme suffers from the universal attack.

To overcome such a serious security weakness, a straightforward approach is destroy the linear relation existing among  $(D_{U,0} + x_U \cdot V)$ ,  $(D_{U,1} + x_U \cdot W)$  and  $T$  when computing  $S_i$ . A simple method to achieve such a goal is to replace  $\tau_i = h(i||name)$  with  $\tau_i' = h(i||name||R_i)$ . Due to  $R_i$ 's change in each block, the modification is able to overcome the security weakness. However, this results in heavy communication costs when the third-party auditor checks the data integrity because the cloud server has to sends  $\{R_i\}_{i=1}^n$  to the third-party auditor. We hope this security weakness could be overcome through other method in near future.

## 5 Conclusions

Recently, Zhang et al. [15] proposed an efficient certificateless public verification scheme for cloud-based cyber-physicalsocial systems to ensure the integrity data stored in the cloud. Although they demonstrated that their scheme is secure against various attacks, this paper points out that a malicious cloud server is able to modify even delete the data without being found by the third-party auditor or the user. Therefore, Zhang et al.'s scheme is not secure for practical applications.

**Acknowledgment.** The work was supported by the Science and Technology Project of State Grid Corporation of China (Research and Application of Life Cycle Management and Control Technology for Power Monitoring System User Behavior (2019–2020)).

## References

1. Xiong, G., et al.: Cyber-physical-social system in intelligent transportation. *IEEE/CAA J. Autom. Sin.* **2**(3), 320–333 (2015)
2. Liu, Z., Yang, D., Wen, D., Zhang, W., Mao, W.: Cyber-physical-social systems for command and control. *IEEE Intell. Syst.* **4**, 92–96 (2011)
3. Ning, H., Liu, H., Ma, J., Yang, L.T., Huang, R.: Cybermatics: cyber-physical-social-thinking hyperspace based science and technology. *Future Gener. Comput. Syst.* **56**, 504–522 (2016)
4. Cui, L., Yu, F.R., Yan, Q.: When big data meets software-defined networking: SDN for big data and big data for SDN. *Netw. IEEE* **30**(1), 58–65 (2016)
5. Hasan, R., Yurcik, W., Myagmar, S.: The evolution of storage service providers: techniques and challenges to outsourcing storage. In: *Proceedings of the 2005 ACM Workshop on Storage Security and Survivability*, pp. 1–8. ACM (2005)
6. Ateniese, G., et al.: Provable data possession at untrusted stores. In: *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 598–609. ACM (2007)
7. Ateniese, G., Di Pietro, R., Mancini, L.V., Tsudik, G.: Scalable and efficient provable data possession. In: *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, p. 9. ACM (2008)
8. Ateniese, G., et al.: Remote data checking using provable data possession. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **14**(1), 12 (2011)
9. Erway, C.C., Küpçü, A., Papamanthou, C., Tamassia, R.: Dynamic provable data possession. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **17**(4), 15 (2015)
10. Chen, H.C.H., Lee, P.P.C.: Enabling data integrity protection in regenerating-coding-based cloud storage: theory and implementation. *IEEE Trans. Parallel Distrib. Syst.* **25**(2), 407–416 (2014)
11. Zhao, J., Chunxiang, X., Li, F., Zhang, W.: Identity-based public verification with privacy-preserving for data storage security in cloud computing. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **96**(12), 2709–2716 (2013)
12. Wang, H.: Identity-based distributed provable data possession in multicloud storage. *IEEE Trans. Serv. Comput.* **8**(2), 328–340 (2015)
13. Wang, B., Li, B., Li, H., Li, F.: Certificateless public auditing for data integrity in the cloud. In: *2013 IEEE Conference on Communications and Network Security (CNS)*, pp. 136–144. IEEE (2013)
14. He, D., Zeadally, S., Wu, L.: Certificateless public auditing scheme for cloud-assisted wireless body area networks. *IEEE Syst. J.* **12**, 64–73 (2015)
15. Zhang, Y., Chunxiang, X., Shui, Y., Li, H., Zhang, X.: SCLPV: secure certificateless public verification for cloud-based cyber-physical-social systems against malicious auditors. *IEEE Trans. Comput. Soc. Syst.* **2**(4), 159–170 (2015)
16. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
17. Hedman, B.A.: An earlier date for “cramer’s rule”. *Historia Math.* **26**(4), 365–368 (1999)



# Advertising Strategy for Maximizing Profit Using CrowdSensing Trajectory Data

Kaihao Lou<sup>1</sup>, Shuqiu Li<sup>1</sup>, Funing Yang<sup>2,3(✉)</sup>,  
and Xingliang Zhang<sup>4</sup>

<sup>1</sup> Department of Computer Science and Technology, Jilin University, Changchun, China

<sup>2</sup> School of Management, Jilin University, Changchun, China  
yfn@jlu.edu.cn

<sup>3</sup> Applied Technology College, Jilin University, Changchun, China

<sup>4</sup> China Mobile Group Jilin Co., Ltd Network Management Center, Changchun, China

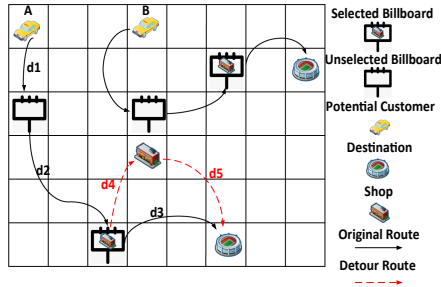
**Abstract.** Out-door billboard advertising is a traditional method to attract potential customers for making commercial profits, which represent the income from attracted customers' consumption minus the cost of billboards. Existing billboard selection strategies usually prefer to select the billboards with a large flow of customers without considering many factors, such as customers' preferences and detour distance. In this paper, a billboard selection optimization problem is formulated to find the appropriate billboards so that advertisers could obtain best commercial profits. First, we adopt the semi-markov model to predict customers' mobility by using crowdsensing trajectory data. Then, with the consideration of customers' preferences and detour distance, two advertising strategies are proposed to address the billboard selection problem for two situations. In the end, we conduct extensive simulations based on the widely-used real-world trajectory: *epfl*. The results of simulations demonstrate that our advertising strategies could achieve the superior commercial profits compared with the state-of-the-art strategies, which could match the analysis of theory.

**Keywords:** Billboard advertising · Semi-markov model · Mobile crowdsensing · Optimization

## 1 Introduction

Out-door billboard is one of the most important methods for advertising. It is necessary to select appropriate billboards for advertisers to obtain the best commercial profits. There are limitations of traditional billboard advertising strategies, which are listed as follows: 1) Existing strategies usually prefer to choose the billboards with a large flow of customers which are highly competitive and

have high costs, while it may not achieve equivalent business returns. 2) Existing strategies may ignore the distance between the billboards and the shops, which could affect the possibility of customers being attracted. Hence it is difficult to select billboards, which could achieve the best profit for advertisers, and we attempt to propose a solution to address those limitations.



**Fig. 1.** The illustration of the gridded map for billboard selection.

With the development of mobile devices and mobile network, it is possible to recruit some mobile users to take a common sensing task through their smartphones, which is called Mobile CrowdSensing (MCS) [3]. In fact, it provides us a new solution to collect customers profiles, i.e., traveling trajectories and preferences [12]. By using those data collected from MCS, we can choose appropriate billboards to advertise so that we can overcome the mentioned limitations.

As shown in Fig. 1, there are billboards in different areas. Passing potential customers may be attracted to the shop when they see the advertisements on the billboards so that the advertisers could obtain the profit. Advertisers can only choose a limited number of billboards due to the limited budget. Hence, our problem can be summarized as selecting billboards from candidate billboards, which could maximize the profits for advertisers with limited budget. In order to address the billboard selection problem, we formulate it into the probabilistic maximum coverage problem which is NP-hard. We propose two practical advertising strategies with consideration of customers’ preferences and their detour distance. The main contributions of this paper are summarized as follows:

- We formulate this billboard selection problem as the NP-hard problem to select appropriate billboards for advertisers to obtain the best commercial profit.
- We propose two advertising strategies taking not only the customers’ preferences but also their mobility into consideration with limited budget. The approximate ratios for these two advertising strategies are  $(1 - \frac{1}{e})$ .
- We conduct extensive simulations based on real-world trajectory: *epfl*. The results show that compared with other strategies, our advertising strategies could achieve better profits for advertisers.

## 2 Related Work

There have been many works on advertising strategy. In [8], Liu *et al.* propose a system, which uses taxi Trajectories to help select the locations of billboards. In [2], Einziger *et al.* propose an algorithm that allows conflict-free, near-optimal advertising selection with low computational complexity to select advertisements to broadcast at each access point so as to maximize revenue. In [5], Huang *et al.* propose a strategy to maximize the coverage of advertisements with consideration of individuals' interests and mobility patterns.

There have also been some works focusing on mobile crowdsensing. In [1], Cheung *et al.* propose an algorithm for calculating the optimal user decision-making under general conditions by using dynamic programming method, and deduces a closed decision-making criterion for the special but practical situation of non-discounted remuneration. In [4], Gong *et al.* focus on the path planning and task assignment problem in mobile crowdsensing, so that total task quality can be maximized with constraints of user travel distance budgets. In [9], Marjanović *et al.* propose an edge computing architecture, which is suitable for large-scale MCS services by putting the main MCS functions in the reference MEC architecture.

## 3 Problem Formulation

### 3.1 System Model

Considering there is an advertising system which is composed of a crew of potential customers, denoted by the set  $U = \{u_1, u_2, \dots, u_n\}$  and also a set of candidate billboards:  $V = \{v_1, v_2, \dots, v_m\}$ . The areas in the map can be represented as  $L = \{l_1, l_2, \dots, l_h\}$ . The cost of candidate billboards is denoted by  $C = \{c_1, c_2, \dots, c_m\}$ , respectively. Moreover, all the potential customers' preferences are denoted by the set  $A = \{a_1, a_2, \dots, a_j\}$ . Hence, without loss of generality, we denote potential customer  $u_i$ 's preferences as  $A_{u_i} \subseteq A$ , while the advertisement  $T$  could match some preferences of customers, which are  $A_T \subseteq A$ .

Each potential customer  $u_i$  starts moving from his initial location, and heads for his destination. When  $u_i$  sees the advertisement on his way, he may decide to go to the shop if the advertisement matches his preferences and the detour distance  $d_i(t)$  is not too far. We assume that if a potential customer  $u_i$  decides to go to the shop, then the advertisers will obtain the profit which is denoted by  $f_i$ . The attracted customers are denoted as  $U_{attracted}$ .

### 3.2 Problem Description

With the limit of budget which is denoted by  $B$ , we attempt to choose a set of billboards which can be denoted as  $S = \{s_1, s_2, \dots, s_k\}$  from  $V$  to do advertising. If a billboard  $v_i$  is selected to  $S$ , then it will deliver advertisement content to those potential customers whose locations are in its area until the deadline.

Hence our purpose is to find the best advertising strategy meeting the following optimization problem:

$$\begin{aligned} \text{Maximize } F &= \sum_{i=1}^{U_{\text{attracted}}} f_i - B \\ \text{s.t. } \sum_{j=1}^k c_j &\leq B, \forall s_j \in S, S \subseteq V \end{aligned} \quad (1)$$

where  $F$  is the total profits for advertisers from billboard advertising. In this paper, we assume that, when a customer is attracted to the shop after he sees the advertisement, then the advertisers will obtain profit from the customer and each customer who has been attracted could create the same profit for advertisers, which is denoted by  $f$ .

### 3.3 NP-Hard Proof

Before solving the above optimization problem, we first attempt to prove that the billboard selection problem is NP-hard, which is shown as follows:

*Proof.* First of all, we want to formulate this problem in Eq. 1 as the probabilistic set coverage problem, which includes a collection of sets  $X = \{X_1, X_2, \dots, X_m\}$  with the corresponding costs  $c = \{c_1, c_2, \dots, c_m\}$ .  $X_i$  consists of a lot of elements, which is denoted as  $O = \{O_1, O_2, \dots, O_n\}$ . The associate possibilities that the elements can be covered are denoted as  $p = \{p_1, p_2, \dots, p_n\}$  and the associated weights are denoted as  $W = \{w_1, w_2, \dots, w_n\}$ . The target is to select a subcollection of  $X$  with constraint of a given budget  $B$  to maximize the number of covered elements.

Then, we reconsider the billboard selection problem in this paper. We can regard the potential customers as the elements set. The probabilities that the potential customers decide to go to the shop when they see the advertisement can be regarded as  $p$ . The profit that the advertiser gets from each customer can be considered as  $W$ . Moreover, we can consider the billboard set we need to choose as  $X$ , and their cost is  $c$ . It is obvious that the probabilistic set covering problem is NP-hard, so that the billboard selection problem is also NP-hard.

## 4 System Overview

### 4.1 Mobility Prediction

First of all, we attempt to predict each potential customer's locations so that we can select the appropriate billboards to improve the effect of advertising. It is not difficult to map each customer's traces into a square area in a plane region like Fig. 1, especially when area is small [7]. Each customer's trace could be converted into a sequence of grids and billboards' locations can be converted into fixed grids. In this paper, we use the semi-markov model [11] to predict the

customers' mobility. One of the most important equation of semi-markov,  $Z(\cdot)$  is defined by Eq. (2).

$$\begin{aligned}
 Z_u(l_i, l_j, T) &= P(S_u^{n+1} = l_j, t_u^{n+1} - t_u^n \leq T | S_u^0, \dots, S_u^n; \\
 &\quad t_u^0, \dots, t_u^n) \\
 &= P(S_u^{n+1} = l_j, t_u^{n+1} - t_u^n \leq T | S_u^n = l_i)
 \end{aligned}
 \tag{2}$$

where  $Z_u(l_i, l_j, t)$  is the probability that the customer  $u$  will move from his current grid  $l_i$  to the grid  $l_j$  at or before time  $T$  when he moves next time.  $S_u^k$  represents the customer  $u$ 's  $k$ -th location during his moving and its corresponding arrival time is denoted as  $t_u^k$ . The grid that the customer will enter in the next time unit is related to his current grid, which can be obtained from the customer's historical trace records. Then, we can define another key equation  $Q(\cdot)$ , denoted by Eq. (3).

$$Q_u(l_i, l_j, T) = \begin{cases} \sum_{k=1}^L \sum_{t=1}^T (Z_u(l_i, l_k, t) - Z_u(l_i, l_k, t - 1)) \cdot \\ Q_u(l_k, l_j, T - t), & i \neq j \\ 1 - \sum_{k=1, k \neq i}^L Z_u(l_i, l_k, T) + \\ \sum_{k=1, k \neq i}^L \sum_{t=1}^T (Z_u(l_i, l_k, t) - Z_u(l_i, l_k, t - 1)) \cdot \\ Q_u(l_k, l_i, T - t), & i = j \end{cases}
 \tag{3}$$

It is easy to find that the potential customers cannot move from one grid to another when  $T = 0$ , so we can get  $Q_u(l_i, l_i, 0) = 1$  and  $Q_u(l_i, l_j, 0) = 0$  ( $i \neq j$ ). Next, we can use the following equation to calculate the expectation of customers' passing any grid  $l_j$  before deadline:

$$P^{l_j}(u) = 1 - \prod_{t=0}^T (1 - Q_u(l_i, l_j, t))
 \tag{4}$$

### 4.2 Customer Preferences Level

After considering customers' mobility, in order to determine the utility of each billboard, we need to measure a potential customer  $u_i$ 's preference level for an advertisement  $T$ , which is denoted as  $P_{prefer}(u_i)$ . The preference level  $P_{prefer}$  can be calculated by the following equation:

$$P_{prefer}(u_i) = \frac{A_{u_i} \cap A_T}{A_{u_i}}
 \tag{5}$$

Obviously, if the advertisement's attributes  $A_T$  could match all the customer's preference  $A_{u_i}$ , then  $P_{prefer} = 1$  which means the potential customer would be likely to go to the shop by the factors of preferences.



### 4.3 Detour Distance

As we mentioned, one of the important factors is the detour distance which is denoted as  $d(u_i)$ . As we can see from the Fig. 1, the original path for the customer (A) is  $d1 + d2 + d3$ . When the customer (A) sees the advertisement, he will decide whether to go to the shop. If he decides to go to the shop, the path to the shop is  $d4$ , and the path from the shop to his original destination is  $d5$ . The detour distance is calculated as follows:

$$d(u_i) = \begin{cases} \min_{v_j \in V, v_j \in S} d4 + d5 - d3, & \text{if } \exists v_j, v_j \in S \\ \infty, & \text{otherwise} \end{cases} \quad (6)$$

During the customer’s path, the customer may see a lot of billboards which have been chosen to advertise, so he will decide if go to the shop after he sees the current billboard. Then the detour distance should be calculated from the current billboard to the customer’s destination. Then we need to measure how the detour distance affects the possibility that the customers decide to go to the shop. The equation is shown as follows:

$$P_{detour}(u_i) = 1 - \frac{d(u_i)}{D_{max}} \quad (7)$$

where  $D_{max}$  is a predefined constant and  $P_{detour}(u_i)$  represents the detour distance level which affects the possibility that the customer will go to the shop. In this paper, we set  $D_{max}$  as the maximum diagonal length in the selected area.

### 4.4 Billboard Utility

In this part, we look into how to calculate billboard utility, which is denoted as  $F(v_j)$ . The billboard utility is the expectation that how many customers that the billboard  $v_j$  could attract. First we need to calculate the probability that the customer will be attracted to the shop after he sees the advertisement, the equation is shown as follows:

$$P_{attract}^{v_j}(u_i) = \alpha P^{l_j}(u_i) \times \beta P_{prefer}(u_i) \times \gamma P_{detour}(u_i) \quad (8)$$

where  $l_j$  is the grid that the billboard  $v_j$  located.  $\alpha$ ,  $\beta$  and  $\gamma$  are relative weights where  $\alpha + \beta + \gamma = 1$ . The probability that the customer will be attracted to the shop could be affected by the different billboards that the customer sees. Therefore, it is necessary to determine the influence that different billboards to the same potential customer, which can be denoted as follows:

$$P_{attract}(u_i) = 1 - \prod (1 - P_{attract}^{v_j}(u_i)), \forall v_j \in S \quad (9)$$

where the  $P_{attract}(u_i)$  is the total probability that the customer  $u_i$  will be attracted to the shop after he sees current billboard with consideration of different billboards impact. Then the utility of a billboard for a specific advertisement can be calculated as follows:

---

**Algorithm 1.** Advertise Strategy For Constant Cost (ASFCC)

---

**Input:** : number of billboards  $k$ , a set of billboards  $V$   
**Output:** : the selected billboard set  $S$

- 1:  $S \leftarrow \emptyset$ ;
- 2:  $F \leftarrow 0$ ;
- 3: **for**  $i = 1$  **to**  $k$  **do**
- 4:      $v_h \leftarrow \arg \max_{v_h \in V \setminus S} F_{S \cup v_h}$
- 5:      $S = S \cup v_h$ ; update  $F$
- 6:      $V = V \setminus v_h$
- 7: **return** the selected billboard set  $S$ .

---

$$F(v_j) = [1 - \prod_{i=1}^n (1 - P_{attract}^{v_j}(u_i))] \times f, v_j \in V, u_i \in U \tag{10}$$

Then we can get the total utility of the billboard set, which is shown in Eq. 11:

$$F = f \times \sum P_{attract}(u_i) - B, \forall u_i \in U \tag{11}$$

## 5 Advertising Strategy

In this section, we propose two advertising strategies for the billboard selection problem in two situations when the billboards have same cost and different cost.

### 5.1 Same Cost for Each Billboard

First, we consider the situation that each billboard has the same cost. In this situation, we can convert the budget restriction into the billboard quantity restriction where we need to select a billboard set, which could maximize the profit for advertisers with the constraint of billboard number  $k$ . The detailed greedy algorithm is shown in Algorithm 1.

We have proposed a greedy advertising strategy to address the above NP-hard problem. According to [13], we can confirm that  $F$  is a submodular function, which can be summarized as that consider there are two arbitrary node sets  $S_1$  and  $S_2$ ,  $S_1 \subset S_2$ , and  $\forall v_k \in V \setminus S$ , the submodular property holds, i.e.,  $F_{S_1 \cup v_k} - F_{S_1} \geq F_{S_2 \cup v_k} - F_{S_2}$ . The bound can also be derived from [13], which is  $(1 - \frac{1}{e})$ .

### 5.2 Different Cost for Each Billboard

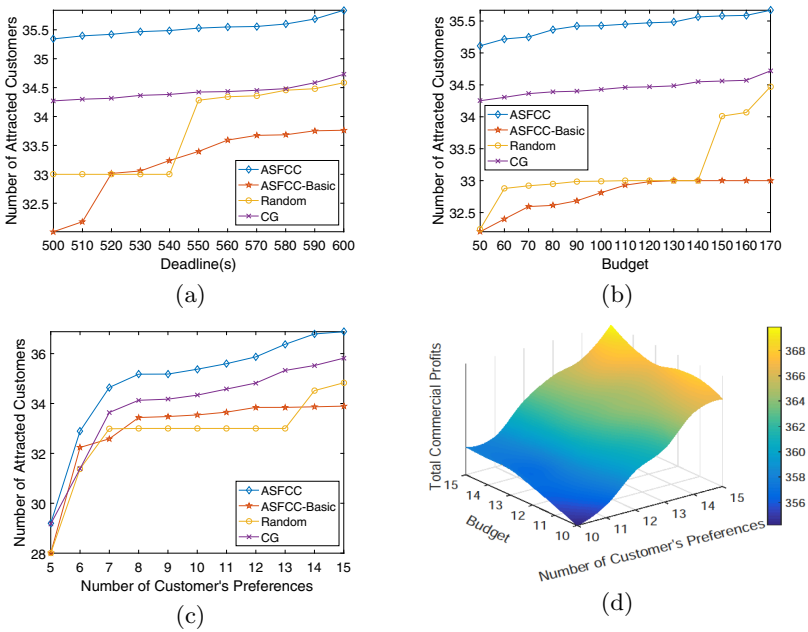
Now, we attempt to propose the advertising strategy for the situation that each billboard has different cost. The detailed algorithm is shown in Algorithm 2. By now, we have proposed another greedy advertising strategy to address the above NP-hard problem when each billboard has different cost. According to [6], we can get  $F(S_j) \geq (1 - \frac{1}{e})F(S_{opt})$ ,  $k \geq 3$  which represents that when  $k \geq 3$ , the approximate ratio for this algorithm is  $(1 - \frac{1}{e})$ .

**Algorithm 2.** Advertise Strategy For Different Cost (ASFDC)

**Input:** : number of billboards  $k$ , a set of billboards  $V$ , cost set for each billboard  $C$ , total budget  $B$

**Output:** : the selected billboard set  $S$

- 1:  $S_1 \leftarrow \arg \max\{F(S_{temp}) \mid |S_{temp}| < k, S_{temp} \subseteq V, \text{ and } c(S_{temp}) \leq B\}$ ;
- 2:  $S_2 \leftarrow \emptyset$ ;
- 3: **for**  $S_{temp} \subseteq V, |S_{temp}| < k, \text{ and } c(S_{temp}) \leq B$  **do**
- 4:     **while**  $V \setminus S_{temp} \neq \emptyset$  **do**
- 5:          $v_h \leftarrow \arg \max_{v_h \in V \setminus S_2} \frac{F(v_h)}{c(v_h)}$
- 6:         **if**  $c(S_{temp}) + c_{v_h} \leq B$  **then**
- 7:              $S_{temp} = S_{temp} \cup v_h$
- 8:         **if**  $F(S_{temp}) > F(S_2)$  **then**
- 9:              $S_2 \leftarrow E$
- 10: **if**  $F(S_1) > F(S_2)$  **then**
- 11:     **return** the selected billboard set  $S_1$ .
- 12: **else**
- 13:     **return** the selected billboard set  $S_2$ .



**Fig. 2.** Performances on *epfl trace set*, when the billboard costs are constant.

## 6 Performance Evaluation

### 6.1 The Simulation Settings and Traces

In this paper a real-world dataset: *epfl trace set* [10], is adopted to test the advertising strategy's performances. In the *epfl trace set*, about 500 taxis' GPS coordinates are included which are collected over 30 days in the San Francisco Bay Area. Each taxi in this dataset is equipped with a GPS receiver and sends location-update to a central server which is fine-grained so that we can accurately interpolate node positions between location-updates.

First of all, we process the dataset by filtering out some abnormal users including those with discontinuous traces or remote locations. Next, we can match these traces into a map area and convert it into gridded map which can be processed by Baidu Map. We randomly select 35 billboards located in different areas to be the candidate billboards. The  $\alpha$ ,  $\beta$  and  $\gamma$  in Eq. 8 are set to  $\frac{1}{3}$ . The preference of each customer is randomly generated to reduce the difficulty. The total preference type  $|A| = 20$ . The deadline in this simulation is set from 500 to 600. The cost for each billboard is set to 10 when the cost is constant and the cost is set from 10 to 20 when the cost is different. The budget is set from 50 to 170 when the cost is constant and it is set from 100 to 200 when the cost is different. The number of each customer's preferences is set from 5 to 15 in our simulation. We repeat our simulation 10,000 times, taking the average as the final result.

### 6.2 Algorithms for Comparison

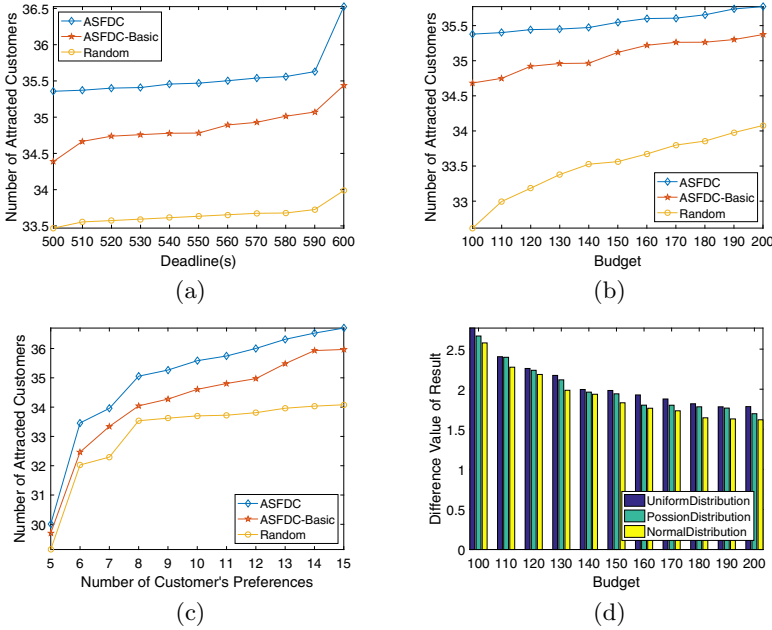
In order to determine the performance of our advertising strategies for two situations, we compare the ASFCC with ASFCC-Basic, Random and Capped Greedy (CG) [14] when the cost of each billboard is same. ASFCC-Basic would select the billboards which have the largest utilities, and Random would randomly select the billboards to advertise. The Capped Greedy (CG) would select the billboards, which could maximize the total utility without consideration of customers' preferences. When the cost of each billboard is different, we compare ASFDC with ASFDC-Basic and Random, where ASFDC-Basic would choose the billboards which have the largest utilities to advertise. In this paper, we use the number of attracted customers to measure the performances of different strategies.

### 6.3 Simulation Results on Constant Billboard Cost

When the cost for each billboard is same, we compare our advertising strategy ASFCC with ASFCC-Basic, Random and Capped Greedy (CG) on *epfl*. The results of simulation are shown in Fig. 2.

As we can see from Fig. 2(a), obviously, the result of ASFCC is the best which means that the billboards selected by ASFCC could attract the most potential customers and the advertisers could get the best profit. It is clear that

in Fig. 2(b), the number of attracted customer by Random is least while other strategies could attract more customers where ASFCC’s performance is the best. Next, in Fig. 2(c), with the growth of the number of customer’s preferences, the number of attracted customers is getting larger in which the performance of Random is the worst while the performance of ASFCC is the best. As shown in Fig. 2(d), the results could demonstrate the analysis for Fig. 2(b) and Fig. 2(c).



**Fig. 3.** Performances on *epfl trace set*, when the billboard costs are different.

Finally, we conduct the simulations to determine the correctness of the approximate for ASFCC. As we can see from Table 1, the results of ASFCC when the deadline is from 500 to 550 are obviously larger than  $(1 - \frac{1}{e})Optimal$  which are consistent with our theoretical analysis.

### 6.4 Simulation Results on Different Billboard Cost

In this part, we conduct the simulations to test performance of ASFDC compared with other two strategies when the costs of billboards are different. The results are shown in Fig. 3.

As shown in Fig. 3(a), it is not difficult for us to find that the billboards selected by ASFDC could attract more potential customers which also means the profit for advertisers could be better. As shown in Fig 3(b), the results of these three strategies improved significantly with the increase of the budget

**Table 1.** Results on *epfl*, when the billboard costs are constant.

Algorithm	Deadline					
	500	510	520	530	540	550
Optimal	37.06	37.27	37.71	37.85	37.87	37.91
$(1 - \frac{1}{e})$ Optimal	23.42	23.56	23.84	23.92	23.94	23.96
ASFCC	35.34	35.39	35.42	35.48	35.52	35.54

**Table 2.** Results on *epfl*, when the billboard costs are different.

Algorithm	Deadline					
	500	510	520	530	540	550
Optimal	37.38	37.41	37.59	37.62	38.02	38.71
$(1 - \frac{1}{e})$ Optimal	23.63	23.65	23.76	23.78	24.03	24.47
ASFDC	35.37	35.40	35.44	35.45	35.47	35.54

where our strategy ASFDC can attract more potential customers to the shop so that the advertisers could get more profit. As we can see from Fig. 3(c), ASFDC could achieve better results than ASFDC-Basic which are much better than Random as the number of customer’s preferences getting larger. We also test the difference between ASFDC and Random when the probability distribution of billboard cost is different, which is shown in Fig. 3(d). It is not difficult to find that the difference between the results from different distribution is small, but when the distribution is uniform, ASFDC works better.

Finally, we evaluate the correctness of the approximate for ASFDC by the simulation results on *epfl trace set* where the deadline is set from 500 to 550. The results are shown in Table 2. Compared with the results of  $(1 - \frac{1}{e})$ optimal, we can easily see that the results of ASFDC are larger which matches the analysis of our theory.

## 7 Conclusion

In this paper, two advertising strategies are proposed in order to maximize the profit for advertisers in two situations. First, we grid the map area to make sure that each billboard locates in a grid. Next we predict the customers’ mobility patterns by semi-markov model. Then we calculate the utility of each billboard. Finally, we propose two advertising strategies in this paper to decide which billboards to select for two different situations. We perform the extensive simulations based on the widely-used real-world trajectory: *epfl*. The results show that, our advertising strategies could bring best profit for advertisers compared with other advertising strategies in two situations.

## References

1. Cheung, M.H., Hou, F., Huang, J.: Delay-sensitive mobile crowdsensing: algorithm design and economics. *IEEE Trans. Mob. Comput.* **17**(12), 2761–2774 (2018). <https://doi.org/10.1109/TMC.2018.2815694>
2. Einziger, G., Chiasserini, C.F., Malandrino, F.: Scheduling advertisement delivery in vehicular networks. *IEEE Trans. Mob. Comput.* **17**(12), 2882–2897 (2018). <https://doi.org/10.1109/TMC.2018.2829517>
3. Ganti, R.K., Ye, F., Lei, H.: Mobile crowdsensing: current state and future challenges. *IEEE Commun. Mag.* **49**(11), 32–39 (2011). <https://doi.org/10.1109/MCOM.2011.6069707>
4. Gong, W., Zhang, B., Li, C.: Location-based online task assignment and path planning for mobile crowdsensing. *IEEE Trans. Veh. Technol.* **68**(2), 1772–1783 (2019). <https://doi.org/10.1109/TVT.2018.2884318>
5. Huang, M., Fang, Z., Xiong, S., Zhang, T.: Interest-driven outdoor advertising display location selection using mobile phone data. *IEEE Access* **7**, 30878–30889 (2019). <https://doi.org/10.1109/ACCESS.2019.2903277>
6. Khuller, S., Moss, A., Naor, J.: The Budgeted Maximum Coverage Problem (1999)
7. Lin, M., Hsu, W.J., Lee, Z.Q.: Predictability of individuals' mobility with high-resolution positioning data. In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp 2012*, pp. 381–390. ACM, New York (2012). <https://doi.org/10.1145/2370216.2370274>
8. Liu, D., et al.: Smartadp: visual analytics of large-scale taxi trajectories for selecting billboard locations. *IEEE Trans. Visual Comput. Graphics* **23**(1), 1–10 (2017). <https://doi.org/10.1109/TVCG.2016.2598432>
9. Marjanović, M., Antonić, A., Žarko, I.P.: Edge computing architecture for mobile crowdsensing. *IEEE Access* **6**, 10662–10674 (2018). <https://doi.org/10.1109/ACCESS.2018.2799707>
10. Piorkowski, M., Sarafijanovic-Djukic, N., Grossglauser, M.: CRAWDAD dataset EPFL/mobility (v. 2009–02-24), February 2009. <https://crawdad.org/epfl/mobility/20090224>, <https://doi.org/10.15783/C7J010>
11. Wang, E., Yang, Y., Wu, J., Liu, W., Wang, X.: An efficient prediction-based user recruitment for mobile crowdsensing. *IEEE Trans. Mob. Comput.* **17**(1), 16–28 (2018). <https://doi.org/10.1109/TMC.2017.2702613>
12. Wang, L., Yu, Z., Yang, D., Ma, H., Sheng, H.: Efficiently targeted billboard advertising using crowdsensing vehicle trajectory data. *IEEE Trans. Ind. Inform.*, 1 (2019). <https://doi.org/10.1109/TII.2019.2891258>
13. Yang, Y., Xu, Y., Wang, E., Lou, K., Luan, D.: Exploring influence maximization in online and offline double-layer propagation scheme. *Inf. Sci.* **450**, S0020025518302287 (2018)
14. Zheng, H., Wu, J.: Placement optimization for advertisement dissemination in smart city. *IEEE Trans. Netw. Sci. Eng.*, 1 (2018). <https://doi.org/10.1109/TNSE.2018.2805768>



# A Method for Community Partition Based on Information Granularity

Tianchu Hang<sup>1</sup>, Yang Bai<sup>2</sup>(✉), and Guishi Deng<sup>3</sup>

<sup>1</sup> College of Letters and Science, University of California, Berkeley,  
CA 94704, USA

<sup>2</sup> School of Information Engineering, Eastern Liaoning University, Dandong 118003, China  
by1997@163.com

<sup>3</sup> Institute of Systems Engineering, Dalian University of Technology, Dalian 116024, China

**Abstract.** The social network community partition is conducive to obtaining hidden and valuable knowledge and rules, which is currently a hot research perspective. Traditional community mining often analyzes network structure information from a static point of view, but ignores the analysis of individual actors' initiative, which limits the construction of community concept model and the effect of community partition. This article argues a method for community partition based on information granularity. First, we optimize the social relationship model by using the link prediction method and establish the similarity model of user social relationship. Second, aiming at the deficiency of K-means clustering algorithm and the defect of high dimension and sparsity of data, the principle of information granularity is introduced in user clustering analysis, and membership degree and generalized equivalence relation of user equivalence relation are given respectively. On this basis, we propose a social community partition method based on the information granularity. Finally, experiments show that, because of the effective integration of the important information of users' social relations and the introduction of information granularity method, the proposed model obtains better I index, and Dunn index evaluation results compared with K-means.

**Keywords:** Social system · User similarity · Community partition · Information granularity

## 1 Introduction

With the rapid development of Web2.0 and social media, Facebook, Twitter, Weibo, Renren, WeChat and other social networks have become an indispensable online platform for people's daily communication and exchange activities. Social networks are the reflection of social activities in the real world [1]. Therefore, in social networks, the interaction between people forms a complex user relationship. In essence, this relationship is not only generated by two users who interact directly, such as FOAF [2], butterfly effect [3] and other principles in the research of complex networks, which show that although there is no apparent connection between the two uses, in fact, may have a myriad of relationships. For example, in Weibo, users form user interaction relationships



through “comments”, “forwarding” and other interactive behaviors, and the subjectivity and real-time nature of these interaction relationships often better reflect the strength of the real relationship between users, and also make social networks show diversity and weighting. Analyzing the user relationship of social network can promote the theoretical research of network evolution, and also has important significance for the application research of personalized recommendation, community partition, community discovery, etc.

The user relationship of social network originates from interpersonal network, which is a self-organizing topological system, reflecting the inherent relationship between people formed in the real world by the interest preference. User’s interest preference can be obtained not only from the inherent attributes and the generated content of the individual, but also through the interaction between individuals. For example, the hot council caused by network events temporarily brings users together. Its core is the emergence of user interest in a specific time, which can form a dynamic community on the network. Considering the general characteristics of user network relationship, building a good social network model is the key, which has an important impact on the efficiency and accuracy of community mining algorithm.

At present, through the analysis of the topological structure of the network graph, the idea of community partition divides the network into tight and loose groups, forming a user centered community structure. The development of community and the social network of users interact and promote each other. Clustering is one of the main technologies of data mining and usually used in community partition. It divides data objects into multiple clusters, which makes the objects in the same cluster have higher similarity and the objects in different clusters have higher dissimilarity. Among the clustering algorithms, K-means clustering is a widely used algorithm, which has the advantages of simplicity, rapidity, scalability and high efficiency. One disadvantage of K-means clustering is that it needs to set the number of clusters in advance, but in the past, we didn’t know how many clusters should be divided for a given data set, which led to the lack of accuracy of the algorithm. Another disadvantage of K-means clustering is that it is relatively sensitive to the initial value. Once the initial value is not well chosen, the clustering result is easy to fall into local minimum. In addition, with the continuous expansion of social network data set size, the increase of data dimensions makes the data sparser, and the distance gap between the data is gradually narrowing. Many algorithms can achieve better clustering results on low-dimensional data, but the performance in processing high-dimensional data is unsatisfactory, which will cause “dimension disaster”. At the same time, due to noise features and redundant features with the increase of clustering, the effectiveness of clustering algorithm is greatly reduced.

Aiming at the deficiency of K-means clustering algorithm and the high dimension and sparsity of data, this paper applies the principle of information granularity to the algorithm of user community partition, gives the membership degree and generalized equivalence relation of user equivalence relation, and proposes a community partition based on information granularity algorithm CPIG (community partition based on information granularity). According to the user’s attention and interaction information, the algorithm establishes the similarity degree of user’s social relationship, optimizes the

model with the link prediction method, and realizes the community partition with the cluster analysis method of information granularity.

## 2 Related Work

### 2.1 User Relationship Network Description

The user relationship of social network constitutes a kind of social network structure. The link prediction method is a powerful assistant tool to analyze the social network structure [4], which can be used to discover the potential relationship between people. Link prediction refers to the prediction of the possibility of link between two nodes in a network that have not yet generated a connection edge through known network structure information [5]. Its analysis process is usually based on graph theory. For example, use triples like  $G = (V, E, W)$  represent a user relationship network, as shown in Fig. 1. Among them,  $V = \{x, y, a, b, c, d, e\}$  represents the user's node set,  $E$  represents the edge set between nodes,  $W$  represents the weight set on the edge between nodes with relationship similarity between user nodes. For example, the weight between  $x$  and  $e$  is similarity  $s_{xe}$ , and the similarity between other user nodes is omitted in Fig. 1. There are directly connected edges between  $x$  and  $e$ , and but there is no directly connected edge between  $x$  and  $y$ . That is to say, the similarity calculation of the relationship between users and users cannot obtain direct similarity. However, through observation, we can see that there are many common nodes between  $x$  and  $y$ , so we can speculate that there is a certain correlation between them. The problem of similarity prediction between  $x$  and  $y$  can be transformed into the problem of network node link prediction, that is, whether there is an implied edge between  $x$  and  $y$  and how much the weight of this edge can be predicted, and then judge whether it will be included in the basic information recommended to. That is to realize link prediction based on similarity.

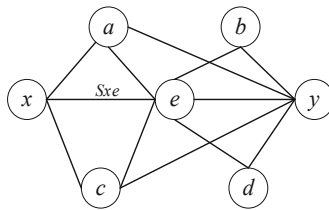


Fig. 1. Diagram of user relationship networks

### 2.2 Link Prediction Method of Weighted Network

Most of the research methods of link prediction are aimed at unauthorized networks, and only a few extend to weighted networks. Lv [6] puts forward weighted prediction method of WRA link based on RA, as shown in formula (1).

$$S_{xy} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w_{xz}^\alpha + w_{zy}^\alpha}{s(z)} \tag{1}$$

Among them,  $w_{xz}$  represents the weight between  $x$  and  $z$ ,  $s(z)$  is the strength of  $z$ . When  $\alpha = 0$ , it means that the graph is an unauthorized graph, and When  $\alpha = 1$ , it means that the graph is a simple weighted graph. Therefore, if Eq. (1) is used as the prediction method of weighted graph, it is necessary to find the appropriate value  $\alpha$  according to the actual measurement of different data sets. Bai [7] introduces the weight value on the proposed RALP and gives the WRALP index, as shown in formula (2).

$$S_{xy} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w_{xz} + w_{zy}}{s(z)} + \varepsilon \sum_{(i,j) \in I_{x \rightarrow y}} \frac{(w_{xi} + w_{ij})(w_{ij} + w_{jy})}{s(i)s(j)} \quad (2)$$

By analyzing the existing weighted network link prediction methods, we can see that both WRA and WRALP adopt edge-weight summing way in predicting nodes and their common neighbors. Here, as long as the edge weights of any prediction node and their common neighbors are larger, the contribution of their common neighbors to the link prediction results is also larger. Corresponding to the actual social network, it can be understood that as long as any one of the two users interacts with their common neighbors frequently, it is considered that the two users are more likely to know each other, which is not always consistent with the real social network. At the same time, it is also a difficult problem to determine the optimal value  $\alpha$  for WRA due to the data set, which often takes a lot of computing time. Zhao [8] puts forward rWRA, a weighted network link prediction method based on the weight similarity of trusted path, and obtains the highest accuracy of similarity prediction on most data sets, as shown in formula (3).

$$S_{xy} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w_{xz} \cdot w_{zy}}{s(z)} \quad (3)$$

### 2.3 Information Granularity Principle

Granularity comes from the physical concept, which originally refers to the “average measurement of particle size”. Scholars combine the concept of granularity with information technology to express the “average measurement of information thickness”. In a general sense, a domain or a “particle” can be represented by a set of propositions like  $E = \{g_1, g_2, \dots, g_n\}$ , and  $g_i$  represents the granularity of information, which is a measure of different levels of information and knowledge refinement [9]. Information granularity is to divide a large amount of complex information into many simple information blocks according to its characteristics and performance, which is similar to the basic idea of clustering analysis. Therefore, scholars combine the two to carry out granularity clustering calculation, mainly used for clustering analysis. In essence, the process of clustering analysis is to define an equivalent relationship between sample points, any sample point belonging to the same class is regarded as equivalent, which determines a partition of the sample point set, corresponding to a class of clustering, and considers that the sample points in the class have similar properties. If it is necessary to describe the relationship between sample points more precisely, the membership scale (threshold) setting is reduced, and the sample point set presents a “fine” outline, so as to obtain “fine” class; otherwise, “coarse” class. we define the selection of similarity and similarity threshold, as shown in definition 1.

- Definition 1: Let  $R$  be an equivalence relation and  $X$  be the domain of the problem. For  $\forall x, y \in X$ , if  $xR_1y \Rightarrow xR_2y$  is satisfied, it is called that  $R_1$  is finer than  $R_2$  a ratio. It is recorded as  $R_1 \leq R_2$ .

Definition 1 is proposed from the aspect of attribute granularity. Generally, the smaller the attribute granularity is, the fewer the number of attributes it contains, and the coarser the corresponding equivalence relationship is. According to this equivalence relation, the coarser the attribute set is divided, so the coarser the attribute granularity is, on the contrary, the finer the attribute granularity is. The principle of information granularity is adopted in clustering results analysis. If the granularity value is larger, the number of clusters will be smaller, and the clustering results will be coarser. otherwise, the smaller the granularity value is, the more the number of clusters will be, and the more detailed the clustering results will be. Therefore, using information granularity to cluster, we can divide objects of different granularity according to the specific requirements of the problem.

### 3 Link Prediction of User Social Relationship

#### 3.1 Relevant Definitions

In order to describe the similarity algorithm among user nodes, the related formal definition and explanation are given firstly: set weighted social network graph  $G = (V, E, W)$ , node set  $V$ , edge set  $E$ , edge weight set  $W$ , if  $x, y \in V$ ,  $w_{xy}$  is the weight between  $x$  and  $y$ . In particular, for unauthorized networks, the default of  $w_{xy}$  is 1.

- Definition 2: Node Strength. Set  $G = (V, E, W)$ ,  $x, y \in V$ ,  $\Gamma(x) \in V$  as the neighbor node of  $x$ . Define the strength of the node  $x$  as formula (4):

$$s(x) = \sum_{y \in \Gamma(x)} w_{xy} \tag{4}$$

- Definition 3: Edge Weight Strength. Let  $G = (V, E, W)$ ,  $x, y \in V$ , define the edge-weight strength between  $x$  and  $y$  [10] as formula (5):

$$sw_{xy} = \frac{w_{xy}}{s(x) + s(y) - w_{xy}} \tag{5}$$

$sw_{xy}$  Represents the proportion of the weights between  $x$  and  $y$  in the sum of the weights of all their neighbors' connected edges. Equation (5) refers to the definition of node strength in Eq. (4). For example,  $s(x)$  is equal to the sum of the weights of all the connected edges. In particular, for an unauthorized network, the node strength is the degree of the node.

### 3.2 A Weighted Graph Link Prediction Algorithm PWRALP

Referring to the weight calculation method of the trusted network from Zhao [11], taking the edge weight product of the user and the common neighbor as the contribution value of the common neighbor, and then extending to the edge weight strength product of the third-order path of the local path similarity, we propose the PWRALP (product weighted RALP) index based on the edge weight product, as shown in formula (6).

$$Sim_{PWRALP}(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w_{xz} \cdot w_{zy}}{s(z)} + \varepsilon \sum_{(i,j) \in I_{x \rightarrow y}} sw_{xi} \cdot sw_{ij} \cdot sw_{jy} \quad (6)$$

The symbols in the formula (6) are described in the previous relevant definitions. In particular, if  $w \notin [0, 1]$ , standardize  $w$  to  $w'$ ,  $w' = e^{-\frac{1}{w}}$ ,  $\varepsilon$  as an adjustable parameter, which value is usually 0.001.

### 3.3 User Similarity Model Based on PWRALP

Through the edge weight formed by the intensity of interaction, a weighted user relationship network is formed. The most important feature of social network is the social relationship between users, such as expressing the friend relationship between users in the form of “concern, fans, circle” on the network application platform. As the platform provides a good media intermediary function, the interaction among friends will be enhanced by “like”, “comment”, “forward” and “share”, which will help to strengthen the tightness between users. Some group norms formed by interaction make different people have some common identity, which forms a meaningful network group [12]. Therefore, the interaction between users is an effective information to study user relationship mining.

The user interaction of social network can be divided into weak association interaction and strong association interaction [13]. Weak association operation refers to the implicit interaction between users, such as often paying attention to the same page, or using the same website application together; strong association operation refers to the more direct interaction between users, such as forwarding, likes and so on. The forwarding behavior expresses the user’s recognition of the forwarded content, and the user’s intention is expressed more clearly. Therefore, this paper measures the degree of correlation between users based on the forwarding behavior between users, as shown in formula (7).

$$\sigma_{xy} = \frac{u_{xy}}{\sum_{k=1}^n u_{xy} + 1} \quad (7)$$

Among them,  $\sigma_{xy}$  represents the similarity between user  $x$  and  $y$ ,  $u_{xy}$  represents the number of comments about  $x$  on  $y$ . Generally, the similarity of pairs  $x$  on  $y$  is different from that of pairs  $y$  on  $x$ . Therefore, the set interactive similarity  $x$  on  $y$  like  $Sim(x, y)$ , as shown in formula (8)

$$Sim(x, y) = \frac{\sigma_{xy} + \sigma_{yx}}{2} \quad (8)$$

The main purpose of user similarity prediction algorithm is to predict the similarity between two users who do not have similarity links on the surface when the data is sparse, so as to generate similarity links between the two users and enrich the data set so as to solve the problem caused by data sparsity. Social network has homogeneity. The more the number of links between users in the network (the greater the node strength), the more frequent the user's interaction behavior (the greater the edge weight strength), and the greater the probability of links among users [14], which is also the problem to be solved by user similarity prediction.

## 4 Process of Community Partition Based on Information Granularity (CPIG)

### 4.1 Generation of Initial Equivalence Relation Set

Calculate the initial equivalence relationship of users.  $D$  is the complete set of users, recorded as  $D = \{u_i\}, i = 1, 2 \dots n$ . The initial equivalence relationship of user  $u_i$  is  $R_i = \{\{[u_i]\}, \{D - [u_i]\}\}$ , where  $[u_i] = \{u_j | Sim(u_i, u_j) \geq \beta_1\}, i, j = 1, 2 \dots, n$ , and  $Sim(u_1, u_j)$  is the similarity model of users  $u_i$  and  $u_j$ ,  $\beta_1$  is the similarity threshold.

### 4.2 Initial Community Partition

The initial equivalence set  $R^*$  is further processed to generate a generalized equivalence relation set  $R'$  and realize community partition.

Let  $R_i$  and  $R_j$  be two initial equivalence relations,  $R_i' = \{\{[u_i]\}, \{D - [u_i]\}\}$  be the generalized equivalence relations of user  $u_i$ , where  $[u_i] = \cup\{u_j | \mu(R_i, R_j) \geq \beta_2\}, i, j = 1, 2 \dots, n$ ,  $\mu(R_i, R_j) = \begin{cases} \frac{|[u_i] \cap [u_j]|}{|[u_i] \cup [u_j]|} & \text{intersection} \\ 0 & \text{No intersection} \end{cases}$ ,  $\beta_2$  is the threshold of membership degree of equivalence relations.

### 4.3 Merging

After the initial community partition is generated, similar users may be divided into different classes. In order to avoid this wrong partition, it is necessary to further merge the classes to get the optimized community partition. Each initial class is regarded as a large granularity and each granularity is merged according to the partition method of information granularity. First, calculate the similarity between each class and other classes, and then merge the two classes with the largest similarity. Repeat the process until the similarity is less than a certain threshold. Let  $C_i$  and  $C_j$  be two initial classes,  $\forall u_x \in C_i, \forall u_x \in C_j$  and the calculation of class similarity  $Sim(C_i, C_j)$  is as shown in formula (9).

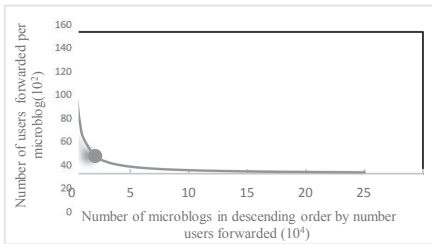
$$Sim(C_i, C_j) = \eta_{ij} \frac{|[u_i] \cap [u_j]|}{|[u_i] \cup [u_j]|} \tag{9}$$

where  $\eta_{ij} = \begin{cases} 1, [u_i] \cap [u_j] \neq \emptyset \\ 0, [u_i] \cap [u_j] = \emptyset \end{cases}$ . After calculating the similarity between classes, we can merge the two classes with the largest similarity until the similarity is less than a certain threshold.

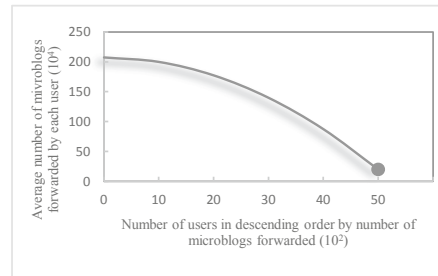
There are two advantages of above method: on the one hand, the integration of the user social relations and the comprehensive similarity of the label theme realizes the comprehensive evaluation mechanism of the user's individual and social attributes, so that the users in the community not only have a high degree of interest similarity, but also a high degree of user social relationship tightness. On the other hand, the information granularity method is used to Row user clustering and clustering merging can set the interval so that it can achieve self-optimization in a certain step size.

## 5 Experiment and Analysis

Our experiments with real data collected from Sina Weibo through web crawler technology, with a total number of 150728 users. Make statistics on the relevant information of the data set, including the text content and release time of microblog, the list of users' concerns, and the interactive information of users' forwarding. Analyze the user interaction information and make a statistical chart of the user forwarding microblog, as shown in Fig. 2 and Fig. 3.



**Fig. 2.** Distribution of forwarding users per microblog



**Fig. 3.** Distribution of the number of microblogs forwarded by each user

In Fig. 2, only a few microblogs have a large number of forwarding users, while 97% of microblogs have fewer than 20 forwarding users, indicating that the interactive data is sparse. In Fig. 3, the distribution of users' microblog forwarding number follows a long tail distribution. The users whose average forwarding number is less than 20 times account for 67% of the total users, and these users need to be filtered out. Continue to clean the data set, and the statistical information of the experimental data obtained is shown in Table 1.

**Table 1.** Statistical results of experimental data

Number of users	Number of microblogs	Average number of original microblogs
7531	864177	114.7

## 5.1 Link Prediction Experiment

According to formula (8), the similarity of user interaction is calculated (in the experiment, the forwarding interaction is the representative), the user interaction network is constructed, and the link prediction based on the weighted graph of interaction relationship is carried out according to formula (6).

**Evaluating Indicator.** We use indicators *AUC* and *Precision* to measure the prediction accuracy of the algorithm. *AUC* is the probability that the score value of the edge in the test set is higher than the score value of a non-existent edge randomly selected. The specific method is as follows: make  $n$  independent comparison, randomly select one edge from the test set each time and compare it with the non-existent edge randomly selected. If the score value of the edge in the test set is greater than the score value of the non-existent edge, add one point and the cumulative number is; if the score value of two edges is greater than the score value of the non-existent edge, add one point and the cumulative number is  $n'$ . If the scores are equal, 0.5 points will be added, and the cumulative number is  $n''$ . Thus, *AUC* is defined as  $AUC = (n' + 0.5n'') / n$ . So, if all the scores are randomly generated, then  $AUC = 0.5$ . Therefore, through  $AUC > 0.5$  we can measure the accuracy of the algorithm. Indicator *Precision* refers to the proportion of accurate prediction in the previous  $L$  prediction links, that is  $Precision = m / L$ , which means if  $m$  of the top  $L$  links are in the test set, then  $m$  links are predicted accurately. Therefore, the greater the *Precision* value is, the more accurate the prediction result is.

**Experimental Steps and Result Analysis.** When the interaction similarity is used as the variable weight value, the link prediction is made for the social relationship network. Three algorithms WRA, WRALP and PWRALP (proposed in this paper) are used to do link prediction experiments to compare the accuracy of prediction. We divide the original data set 100 times randomly and get the training set with 90% links and the test set with 10% links.

In AUC index evaluation method, 500, 1000 and 3000 random sampling comparisons are conducted respectively, and the prediction accuracy results are shown in Table 2. The scores of all possible edges that do not exist in the training set are calculated, and the network with 7531 nodes and 12752 edges is obtained by deleting the data of isolated points. The number of node pairs is  $7531 \times 7530 / 2 - 12752 = 28341463$ . The method of random sampling is adopted in the selection of user nodes, and each extraction of three numbers is carried out independently. The edge of node relationship is considered as one edge if two user nodes have “forwarding” relationship. In the evaluation method *Precision*, only the prediction link with score value in front has the recommended significance. The values  $L$  we set are 500, 1000 and 2000. Take the first  $L$  links in the prediction result and compare it with the test set. See Table 3 for the prediction accuracy result.



**Table 2.** AUC results

	n = 500	n = 1000	n = 3000
WRA	0.5694	0.7491	0.8685
WRALP	0.5748	0.7586	0.8751
PWRALP	0.5798	0.7675	0.8853

**Table 3.** Precision results

	L = 500	L = 1000	L = 2000
WRA	0.3277	0.6018	0.8360
WRALP	0.4421	0.6294	0.8572
PWRALP	0.4478	0.6389	0.8726

It can be seen from Table 2 that the values of these three indicators *AUC* are very close, and it is impossible to explain which algorithm is better. The reason is that in different scale data sets, the probability of two nodes having a common neighbor is different. Compared with small and medium-sized networks, the probability of large-scale networks is lower, which leads to little difference in the value *AUC* of link prediction method based on common neighbor. Therefore, it is not appropriate to test large-scale data sets only by *AUC*. The three algorithms in Table 2 are randomly divided into 10 training sets and test sets respectively, and the calculated average value is taken as the experimental results of each group, where  $\varepsilon = 0.001$ . The prediction accuracy of PWRALP is higher than that of WRA and WRALP. Therefore, the weighted method proposed in this paper achieves the best prediction results under different recommended lengths.

**5.2 CPIG Experiment**

**Evaluating Indicator.** Index *I* refers to the maximum distance between the current class and the center of the class in all classes to measure the degree of separation between classes, while using the sum of the distance between each point in the class and the center of the class to measure the degree of tightness within the class [14]. The influence of clustering number on the index  $1/m$  can be eliminated by introducing coefficient. The core idea of index *I* is to find a maximum value to make the clustering result optimal. Set the number of clusters as *m*, and the definition of the index is as follows formula (10).

$$I_m = \left[ \frac{1}{m} \frac{\sum_{x \in D} d(x, c)}{\sum_{i=1}^m \sum_{x \in c_i} d(x, c_i)} \max_{1 \leq i < j \leq m} d(c_i, c_j) \right]^p \tag{10}$$

Index DUNN is an internal evaluation mechanism. The evaluation results only depend on the clustering data itself. The purpose of DUNN is to distinguish whether the clustering is compact (i.e. whether the variance in the same cluster is small enough) and properly separated (whether the mean value between different clusters is relatively far away). For a given cluster, the higher DUNN is, the better the result is. Let the number of clusters be *m*, and the definition of this index is as follows formula (11).

$$DI_m = \frac{\min_{1 \leq i < j \leq m} \delta(c_i, c_j)}{\max_{1 \leq k \leq m} \Delta k} \tag{11}$$

Where  $\delta(c_i, c_j)$  represents the distance measurement between clusters, and  $\Delta k$  represents the distance within clusters. See formula (12) and formula (13) for specific

definitions.

$$\delta(c_i, c_j) = \min(\text{dist}(x_i, x_j)), x_i \in c_i \cap x_j \in c_j \tag{12}$$

$$\Delta k = \max(\text{dist}(x_i, x_j)), x_i \in c_\tau \cap x_j \in c_\tau \tag{13}$$

**Experimental Procedure and Result Analysis.** According to formula (8), we do the experiment to divide information granularity community. by considering that experimental data set, the social relationship in this paper is represented by the forwarding interaction relationship. Therefore, the parameter  $\gamma = 0, \mu = 0.1$ . It is calculated that  $\alpha = 0.6, \beta = 0.4$  from the formula (13), we can adjust the granularity of class partition through adjusting the parameters  $\beta_1$  and  $\beta_2$ .

**Table 4.** Indexes I and DUNN of two clustering algorithms

Indexes	Algorithms	$\beta_1 = 0.5$	$\beta_1 = 0.5$	$\beta_1 = 0.5$	$\beta_1 = 0.5$	$\beta_1 = 0.5$	$\beta_1 = 0.5$	$\beta_1 = 0.5$
		$\beta_2 = 0.2$	$\beta_2 = 0.3$	$\beta_2 = 0.4$	$\beta_2 = 0.5$	$\beta_2 = 0.6$	$\beta_2 = 0.7$	$\beta_2 = 0.8$
I	CP-K-means	0.014	0.016	0.024	0.028	0.03	0.032	0.032
	CPIG	0.017	0.025	0.031	0.036	0.041	0.035	0.032
DUNN	CP-K-means	0.141	0.157	0.176	0.182	0.195	0.216	0.229
	CPIG	0.153	0.179	0.218	0.229	0.247	0.223	0.21

K-means is chosen to do a comparative test with CPIG proposed in this paper. Take the values  $\beta_1$  and  $\beta_2$  of different combinations and different particle sizes. See Table 4 for the results of indexes I and DUNN.

It can be seen from Table 4 that the algorithm in this paper is generally better than K-means algorithm for I index and Dunn index in the same perspective. Next, under the parameter setting of  $\beta_1 = 0.5$  and  $\beta_2 = 0.6$ , select K-means as the algorithm of comparison test. Due to the large size of the original data set, data sets of different sizes are selected for testing. The matrix is tested with the data of  $200 \times 200, 200 \times 400, 200 \times 600, 200 \times 800, 200 \times 1000, 200 \times 1200, 200 \times 1400$ . The results are shown in Fig. 4 and Fig. 5. It can be seen that, with the increase of data, the algorithm in this paper is better than K-means algorithm by analyzing the comparison between I index and Dunn index.

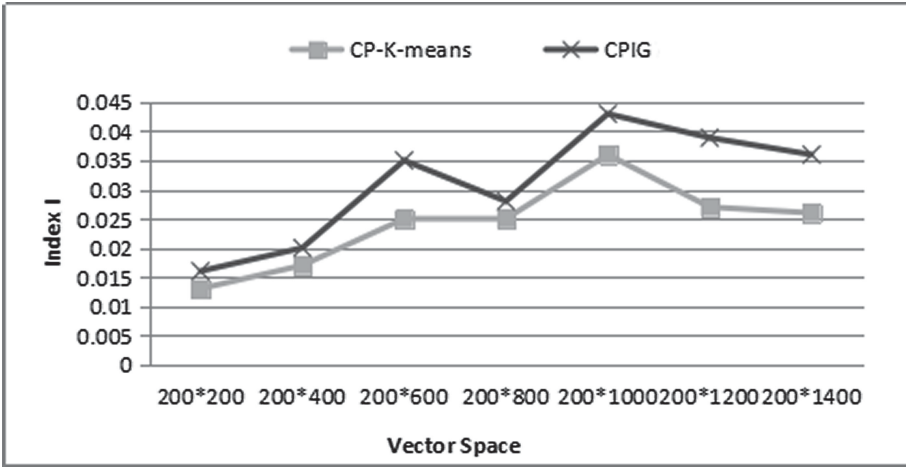


Fig. 4. I index under  $\beta_1 = 0.5$ ,  $\beta_2 = 0.6$

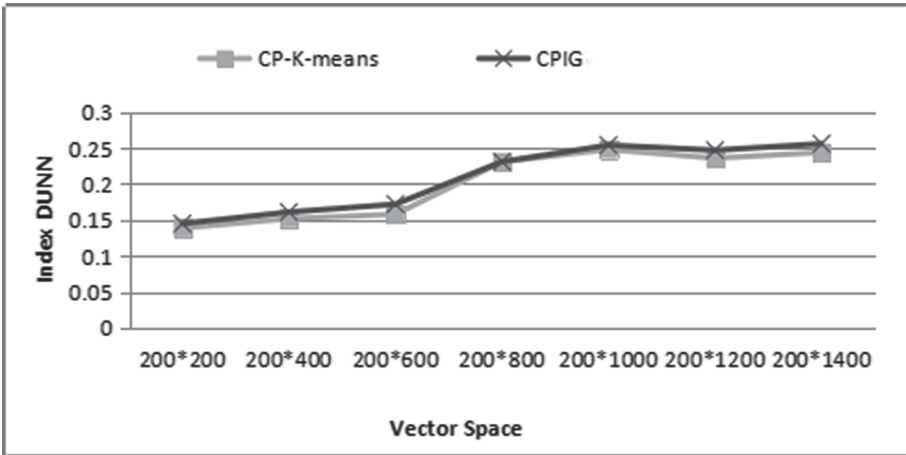


Fig. 5. DUNN index under  $\beta_1 = 0.5$ ,  $\beta_2 = 0.6$

## 6 Conclusion

In this paper, the node index and path index are combined, and the trusted path weight calculation method is introduced. The PWRALP weighted similarity index based on the edge weight product is proposed. The comparison experiment shows that this method can achieve higher prediction accuracy. In the follow-up study, we will analyze the directionality of social networks, and combine with the results of this study to make a comprehensive analysis of the prediction of user relationship links of social networks. According to the principle of information granularity, because of the high dimension of feature vector and the high sparsity of data, an efficient storage and calculation method of user feature vector is designed, which reduces the spatial complexity of data storage,

is superior to the traditional K-means clustering algorithm, and effectively solves the problems brought by the high dimension and sparsity of data.

As future work, we plan to extend our clustering algorithm to the semantic level for corresponding to meaningful topic domains. We also hope to propose a method for web services recommendation for further evaluate our work.

**Acknowledgement.** This work was supported by the National Natural Science Foundation of China (Grant No. 71372083) and university level Doctoral Research Initiation Fund Project (Grant No. 2019BS025). The authors thank the anonymous reviewers for their helpful suggestions.

## References

1. Li, B., Chen, Z.-g., Huang, R., et al.: Link prediction friends recommendation algorithm for online social networks named JAFLink. *J. Chin. Comput. Syst.* **38**(8), 1741–1745 (2017)
2. Diaz Perez, M.: Redes sociales en Internet: aplicacion FOAF (Friend-of-a-Friend) Social networks in Internet: FOAF application (Friend-of-a-Friend). *Rev. Cubana Informacion Ciencias Salud* **15**(6), 1–7 (2007)
3. Zhang, F., Si, G.-y., Luo, P.: A survey for rumor propagation models. *Complex Syst. Complex. Sci.* **6**(4), 1–11 (2009)
4. Jiang, M.-s., Ge, J.-f., Chen, L.: Parameter selection algorithm for link prediction in networks with node attributes. *J. Chin. Comput. Syst.* **38**(6), 1278–1283 (2017)
5. Lv, L.-y.: Link prediction on complex networks. *J. Univ. Electron. Sci. Technol. China* **39**(5), 651–661 (2010)
6. Lv, L.-y., Zhou, T.: Link prediction in weighted networks: the role of weak ties. *EPL* **89**(1), 18001 (2010)
7. Meng, B., Hu, K., Tang, Y.: Link prediction based on a semi-local similarity index. *Chin. Phys. B* **20**(12), 498–504 (2011)
8. Zhao, J., Miao, L.-l., Yang, J., et al.: Prediction of links and weights in networks by reliable routes. *Sci. Rep.* **5**, 1–15 (2015)
9. Zhou, T., Lü, L., Zhang, Y.C.: Predicting missing links via local information. *Eur. Phys. J. B* **71**(4), 623–630 (2009)
10. Guo, J.-f., Liu, M.-m., Xu, L.: Link prediction based on similarity of nodes of multipath in weighted social networks. *J. Zhejiang Univ. (Eng. Sci.)* **50**(7), 1347–1352 (2016)
11. Zhao, J., Miao, L.-l., Yang, J., et al.: Prediction of links and weights in networks by reliable routes. *Sci. Rep.* **5**, 1–15 (2015)
12. Backstrom, L., Kumar, R., Marlow, C., et al.: Preferential behavior in online groups. In: *International Conference on Web Search and Data Mining*, vol. 28, pp. 117–128. ACM (2008)
13. Zhang, X., Yu, Z.-w., Liang, Y.-j., et al.: Community development method based on interactive similarity. *Comput. Sci.* **41**(4), 215–218 (2014)
14. Yan, F., Zhang, M., Tan, Y.-w., et al.: Community discovery based on actors' interests and social network structure. *J. Comput. Res. Dev.* **47**(z1), 357–362 (2010)



# Forward Calculation for Improving the Sensitivity of Multiple Perturbations in Magnetic Induction Tomography Based on Brain Tissue Structure

Yi Lv<sup>(✉)</sup>

Shenyang Aerospace University, Shenyang 110136, Liaoning, China  
lvyi\_julia@126.com

**Abstract.** Magnetic Induction Tomography (MIT) is a non-invasive and contactless electromagnetic imaging method, which is especially suitable for medical monitoring. The low sensitivity of the central region of the measured target has always restricted the application of MIT in practice. In this paper, based on the analysis of the biological structure of human brain, the sensitivity of the brain structure to the forward problem is studied. The experimental parameters are certain, the measurement model is established, and the perturbation conductivity value is calculated and compared in the case of multiple perturbations to the target object under different conditions. The result shows that, under the ideal condition of parameter setting, adding perturbations to the target region can improve the sensitivity within the region can provide a new way of thinking for the research of improving the sensitivity in MIT.

**Keywords:** Magnetic induction tomography · Sensitivity · Forward problem · Perturbation

## 1 Introduction

Magnetic induction tomography is a technique which use excitation coils to produce the eddy current field in the tissue and use detection coils to detect it and imaging the internal conductivity of tissue. The magnetic field generated by MIT easily penetrates the human tissue such as skull. Compared to the existing medical image technique such as CT, MRI, PET and ultrasound, it is non-radiation, inexpensive, without implanting drugs in the human body, portable, and long-time monitoring easily. Therefore, the research on the lab experimental model is the key for MIT to apply in medical human measurement.

In 2004, Scharfetter simulate eddy current problems with two dimension model of an edema in a human head model with MIT [1]. The edema is simulated as a spherical perturbation on the spherical and high conductivity 0.162 S/m and frequency is 100 kHz. Prove that it is possible to detect edema with MIT, although he didn't describe whether it tends to H or I. But we can infer from the conductivity it point to H. In 2007, Schafetter used MIT to image 2D edema in the brain [2]. In 2009, H. Griffiths team used MIT to

forward modeling and imaging 3D haemorrhagic cerebral stroke. They use 12 tissues, the operating frequency is 10 MHz, the conductivity of stroke is nearly blood [3, 4]. Later, they also used the same model to computer with frequency-difference MIT [5]. 2010, Peyton team began to use simulation to access the feasibility of detecting a hemorrhagic type stroke with 3D model. They chose the frequency of 10 MHz and the stroke they chose the same as bloods [6–8]. In 2012, Jorge Caeiros used 4 layers of 3D human head tissue to establish a phantom to simulate haemorrhage [9]. He use 1 MHz frequency and the conductivity which is the same as blood, 0.8 S/m. In 2017, Zhili Xiao established 2D human brain model with 6 layers tissue to simulate the haemorrhagic stroke and imaging [10, 11]. The frequency was chosen for 1 MHz and 10 MHz, the stroke conductivity is individually 0.822 S/m and 1.097 S/m.

This paper analyzes the electromagnetic characteristics of biological tissue and structure features of the human brain, a certain experimental parameters, in according to different situation to put forward the target object disturbance was added to improve the sensitivity of the method, under the condition of disturbance conductivity value change more than a large amount of experiments, ideally obtained parameter setting, adding disturbance in the target area can improve the sensitivity of regional internal, offers a new way to improve the sensitivity.

## 2 Materials and Methods

### 2.1 Tissue Structure of Human Brain

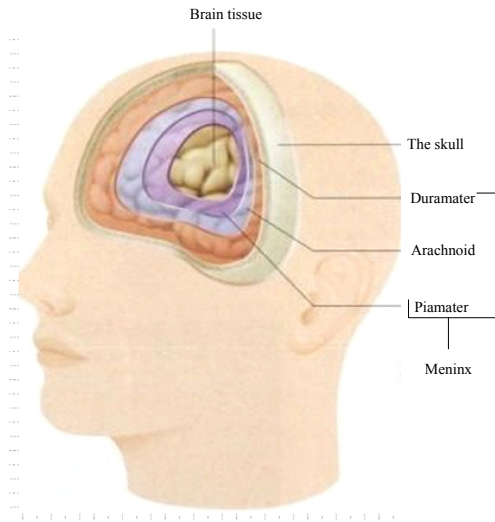
Human brain tissue is a biological conductor with a certain impedance. As can be seen from Table 1, different biological tissues have different impedance. Most of the lesions in craniocerebral tissue occur in the brain, and some also involve cerebrospinal fluid in contact with the cerebral cortex. The electrical impedance changes significantly during the lesions. For example, for a concussion or atrophy of a brain muscle, abnormal tissue resistance is twice as high as normal brain tissue. For cerebral hemorrhage, the blood resistance is  $1.5 \Omega \cdot \text{m}$ , and the heterotropic tissue is about a quarter of the normal tissue in the brain. For brain tumors, the tumor site impedance is  $80 \Omega \cdot \text{m}$ , and the aberrant tissue was 13 times larger than the normal tissue in the brain [19–21].

Human brain accounts for 2% of body weight, but requires 20% of human blood supply. Together with the spinal cord, the brain not only regulates unconscious activities, but also coordinates most conscious movements. The brain is the part that generates consciousness and enables people to think and learn [12, 13]. The structure of the brain is intricate and complex. In order to conduct relevant studies on the basis of the model in MIT, and to take the scalp tissue into consideration, the section structure of the brain is simplified as shown in Fig. 1.

The brain has a variety of protective layers, the first and most important being the skull. There are three layers of protective membranes between the skull and the brain's gray matter: the dura mater attached to the skull, the subdural arachnoid and the pia mater attached to the surface of the brain. Cerebrospinal fluid (CSF), located between the arachnoid membrane and the pia meninges, also has a protective effect. Brain tissue floats in CSF. There are two types of brain tissue: gray matter and white matter. The outer gray matter is the cerebral cortex, with an average thickness of 3–5 mm. There are islands

**Table 1.** Electrical impedance for brain and other tissue.

Tissues	Conductivity ( $\Omega \cdot m$ )
The bone	166
Fat	21–28
The gray matter	2.8
The white matter	6.8
Blood	1.5
The plasma	0.66
Cerebrospinal	0.65



**Fig. 1.** Cross-section structure of the human brain.

formed by gray matter deep inside the brain. The deep surface of the cerebral cortex is the white matter that constitutes the brain’s internal parenchyma, mainly composed of nerve fibers.

Since the head of a person is approximately a sphere, the concentric circle model can be used for simulation study of two-dimensional MIT research. The layers are 1, 2, 3 and 4. For the 1-tier model, only the brain is included; For the 2-tier model, including the brain and scalp; For the 3-layer model, including the brain, skull and scalp; For the 4-layer model, the brain, cerebrospinal fluid, skull and scalp are included. When studying the electrical characteristics of human head, since the head structure is approximately spherically symmetric, concentric spherical models with various complexities can be established, such as 3-layer model, 4-layer model, and isotropic multi-layer spherical model, etc. It is assumed that the medium of each layer is uniform and isotropic [14, 15].

### 2.2 The Forward Problem of MIT

The forward problem is to compute the measured signals with the given setup including geometry, distribution of dielectric properties, the operating frequency and coil excitation current. The Maxwell's equations for MIT with time-harmonic fields and linear materials can be written in the form [16]:

$$\begin{aligned} \nabla \times E &= -i\omega\mu H, \quad \nabla \times H = J_s + E(\sigma + i\omega\varepsilon), \\ \nabla \cdot \varepsilon E &= \rho, \quad \nabla \cdot \mu H = 0. \end{aligned} \tag{1}$$

Here, E and H are the electric and magnetic fields,  $\mu$  is the permeability,  $\rho$  is the electric charge density,  $J_s$  is the current sources. The term  $iE\omega\varepsilon$  corresponds to the displacement current which can be ignored when the operating frequency is under 100 kHz and can't be ignored when the frequency is above 100 kHz. By application of the temporal gauge  $E = -i\omega A, B = \nabla \times A$ , the resulting A-formulation of the vector wave equation when it is under 100 kHz and above 100 kHz are individually as follows [23–25]:

$$\nabla \times (\mu^{-1} \nabla \times A) + i\omega\sigma A = J_s \tag{2}$$

$$\nabla \times (\mu^{-1} \nabla \times A) + (i\omega\sigma - \omega^2\varepsilon)A = J_s \tag{3}$$

Once the vector potential is obtained, the induced voltage can be calculated as the line integral of the tangential components of A along a sensor coil as [26]:

$$V = \oint_{Coil} E \cdot dl = -i\omega \oint_{Coil} A \cdot dl \tag{4}$$

Where dl is length element of the coil.

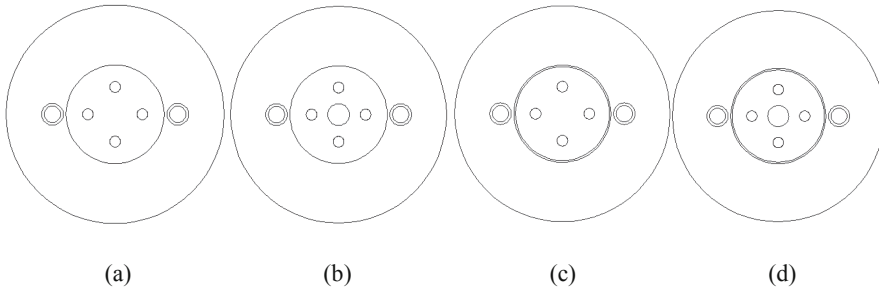
The current of the excitation coil is set to  $I_0$ , thus the sensitivity of induced voltage to conductivity is [17, 18],

$$\frac{\partial V_{ij}}{\partial \sigma_k} = -\omega^2 \frac{\int A_i \cdot A_j dv}{I_0} \tag{5}$$

### 2.3 Experimental Measurement Model

Conductivity value close to the person brain hemorrhage, for example, the conductivity of target and area 0.2 S/m, was largely the hematoma in the middle of the small circular area of 0.8 S/m, high electrical conductivity shielding is 1.6 S/m, adding in the target area for different conductivity value of disturbance objects, aims to change the conductivity values observed among changes in order to improve the sensitivity, sensitivity experiments measurement model is shown in Fig. 2. In each figure, the left ring is the excitation coil and the right ring is the detection coil. The inner diameter is 15 mm and the outer diameter is 20 mm. The great circle between the two coils is similar to the cross-section of the human head, and its radius is 90 m. The small circles in the middle of Fig. 2(b) and (d) are assumed to be centered tumors with a radius of 20 mm. In Fig. 2(c) and (d), the thickness of the annular shielding similar to that of CSF is 3 mm. For the convenience of calculation, the shielding circle is set at the outermost with a radius of 200 mm. The excitation frequency is set at 10 MHz and the current is 0.5 A.





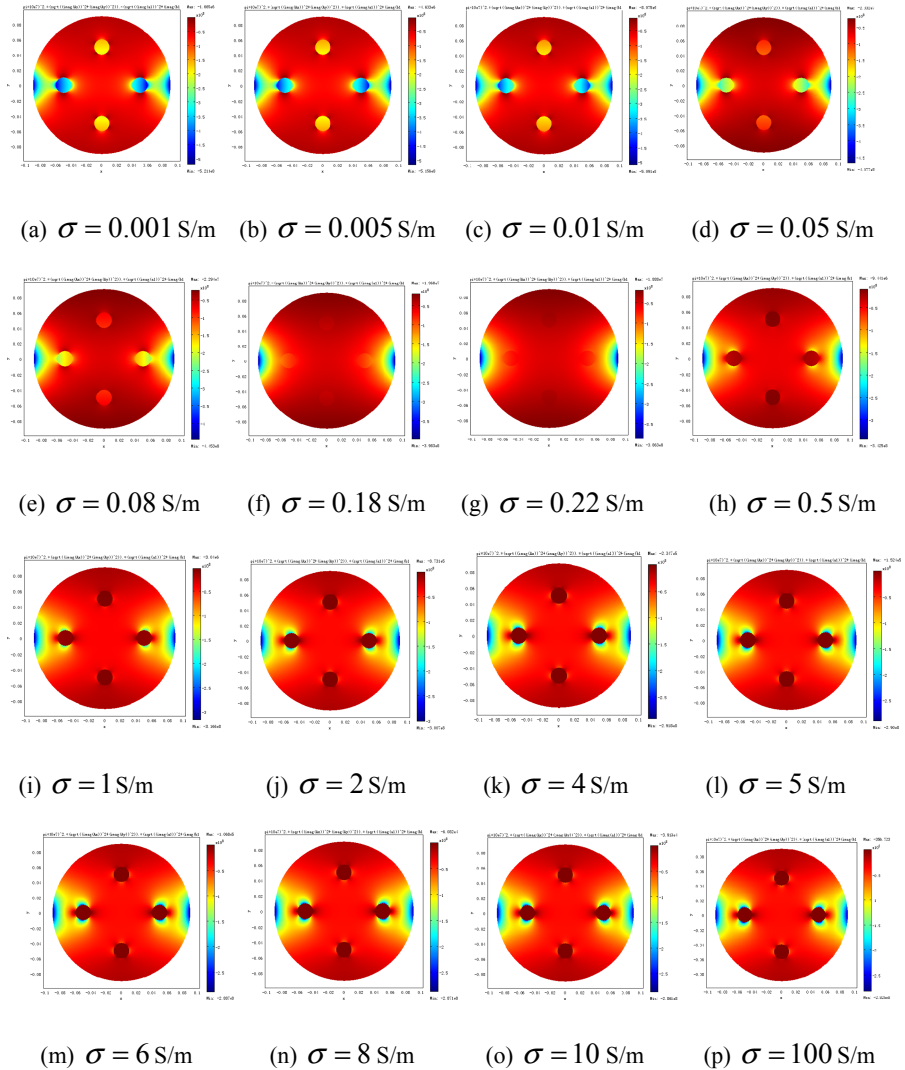
**Fig. 2.** Measurement model with four perturbation for different structure. (a) uniform measurement model; (b) one perturbation in the middle; (c) shielding measurement model; (d) a model with both one perturbation and shielding.

### 3 Experimental Results and Analysis

According to Fig. 2(a), the conductivity values of the added edge disturbance in the target area were set as 0.00 S/m, 0.005 S/m, 0.0 S/m, 0.08 S/m, 0.18 S/m, 0.22 S/m, 0.5 S/m, 1 S/m, 2 S/m, 4 S/m, 5 S/m, 6 S/m, 8 S/m, 10 S/m and 100 S/m, respectively, for sensitivity calculation, as shown in Fig. 3.

For the uniform background detection region, when four small perturbations were added to the region, the sensitivity of the region changed from the appearance when no perturbations were added. It can be seen from the Fig. 4, figure (a), (b) and (c) similar to that of the conductivity values corresponding to 0.001 S/m and 0.005 S/m and 0.01 S/m, relative to the background of the 0.2 S/m, the sensitivity of the edge disturbance area relatively obvious, the disturbance area near the two coil sensor sensitivity is high and far away from the sensor coil two disturbance area is lesser, high sensitivity of the area figure (d) and (e) are similar in appearance. The conductivity values correspond to 0.05 S/m and 0.08 S/m, respectively. With the increase of the conductivity, the sensitivity of the edge disturbance decreases, and the sensitivity region also decreases. Figure (f) and (g) are similar in appearance, and the conductivity corresponds to 0.18 S/m and 0.22 S/m, respectively. At this time, the sensitivity of the edge disturbance region becomes very insignificant, which is integrated into the background conductivity. The main reason is that the background conductivity is very close to the disturbance when the background conductivity is 0.2 S/m, and the sensitivity region is still not strong. Figure (h)–(p) has a similar appearance, that is, after the edge conductivity is greater than 0.5 S/m, until 100 S/m, the sensitivity of the edge disturbance region becomes extremely small, and the area close to and far from the coil is not large. Except for the disturbance around the coil, a small part of the high sensitivity region is generated, and the relatively high sensitivity region of the center increases.

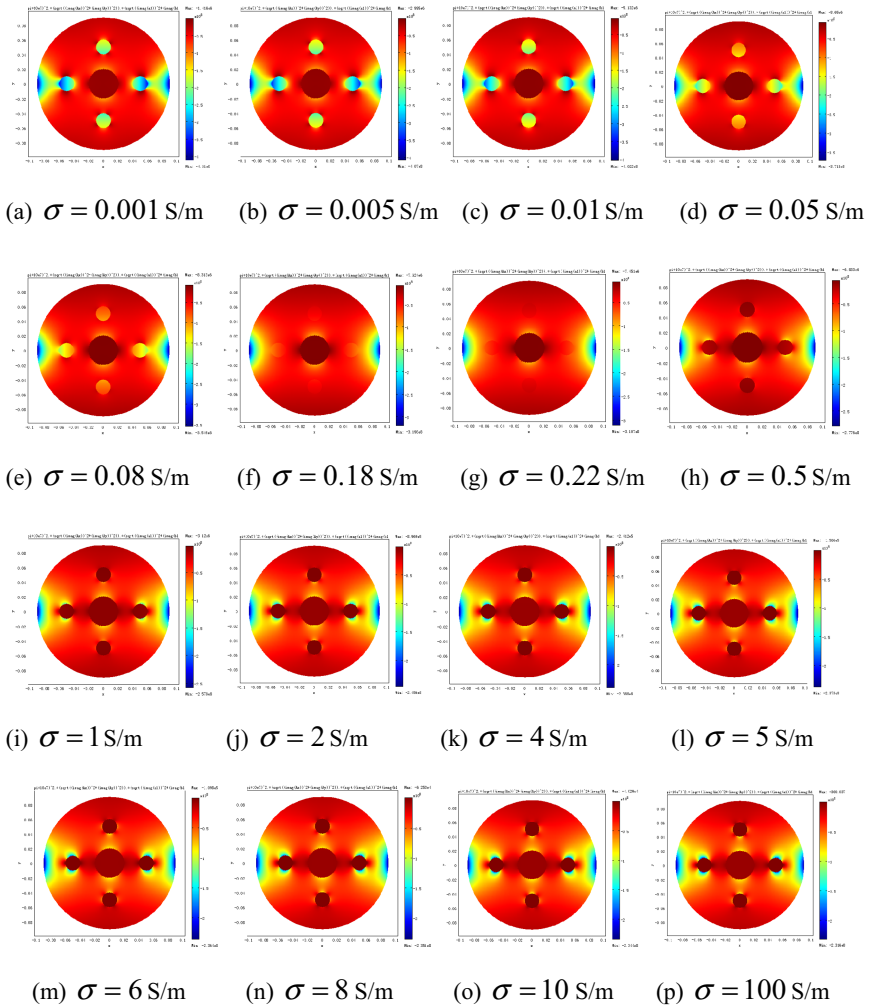
For Fig. 2(b), add an intermediate disturbance of 0.8 S/m when the background conductivity is 0.2 S/m, and then add the corresponding edge disturbance in Fig. 3 for sensitivity calculation, as shown in Fig. 4.



**Fig. 3.** Sensitivity map about uniform measurement model with four perturbation for different conductivity.

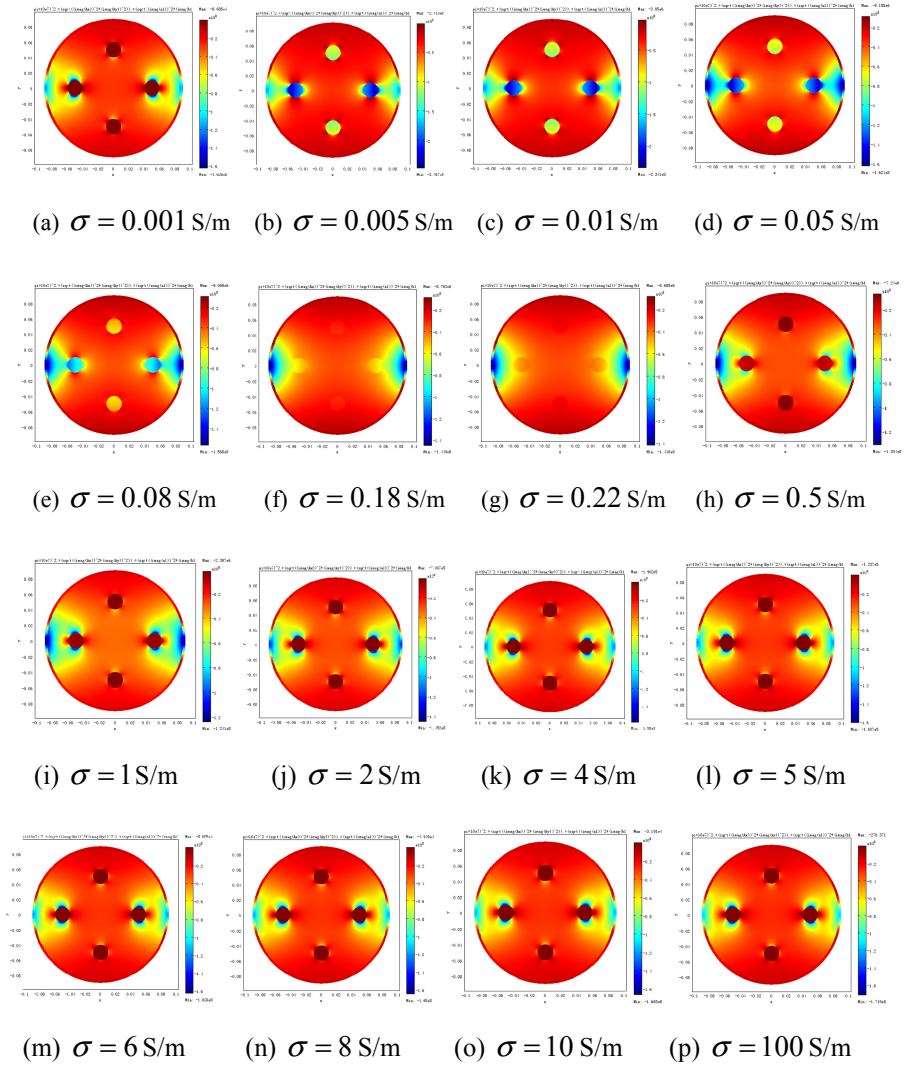
From the appearance of Fig. 4, its change rule is similar to that of Fig. 3. The difference is that in this case, due to the occurrence of intermediate disturbance, the high sensitivity region in the middle of the detection region increases, and the high sensitivity region at the edge of the intermediate disturbance is driven.

For Fig. 2(c), edge shielding with an conductivity value of 1.6 S/m was added under 0.2 S/m of background conductivity, and the corresponding edge disturbance in Fig. 3 was added for sensitivity calculation, as shown in Fig. 5.



**Fig. 4.** Sensitivity map about one perturbation with  $\sigma = 0.8$  S/m in the middle measure model with four perturbation for different conductivity.

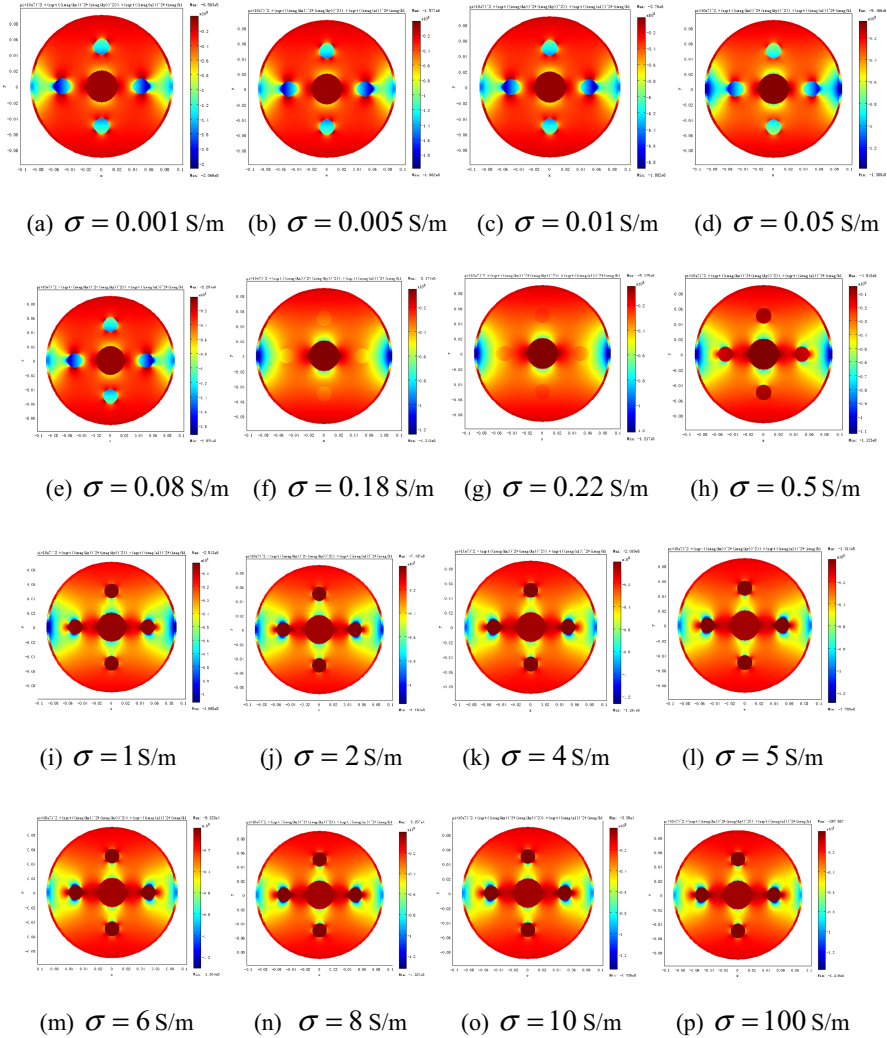
As can be seen from Fig. 5, the sensitivity region has changed. On the whole, the sensitivity region has been enlarged in all cases, and its appearance is similar to that in the case of no shielding. In figure (a), the conductivity of the four edges is 0.001 S/m. The results show that the sensitivity of the added edge disturbance is no different whether it is close to the coil or far away from the coil. The values are all relatively low, but the sensitivity around the edge disturbance close to the coil is slightly improved. Figure (b)–(e) is similar. The conductivity values of the edge disturbance are 0.005 S/m, 0.01 S/m, 0.05 S/m and 0.08 S/m, respectively. The sensitivity of the edge disturbance area close to the coil is relatively high, while the sensitivity far away from the coil is relatively low, and the conductivity at 0.08 S/m is obviously slightly lower than other case. Figure (f)



**Fig. 5.** Sensitivity map about one perturbation with  $\sigma = 1.6$  S/m in the middle measure model with four perturbation for different conductivity.

and (g) are similar. Since they are similar to the value of the background conductivity, the edge disturbance is not obvious. However, due to the appearance of shielding, the disturbance close to the coil has a certain value, and the difference can be seen. Figure (h)–(p) is similar. The conductivity ranges from 0.5 S/m to 100 S/m. The sensitivity of the edge disturbance region is not significantly different whether it is close to or far from the coil.

For Fig. 2(d), the intermediate disturbance with the conductivity value of 0.8 S/m and the edge shielding with the conductivity value of 1.6 S/m were added under the background conductivity of 0.2 S/m, and the corresponding edge disturbance in Fig. 3 was added for sensitivity calculation, as shown in figure.

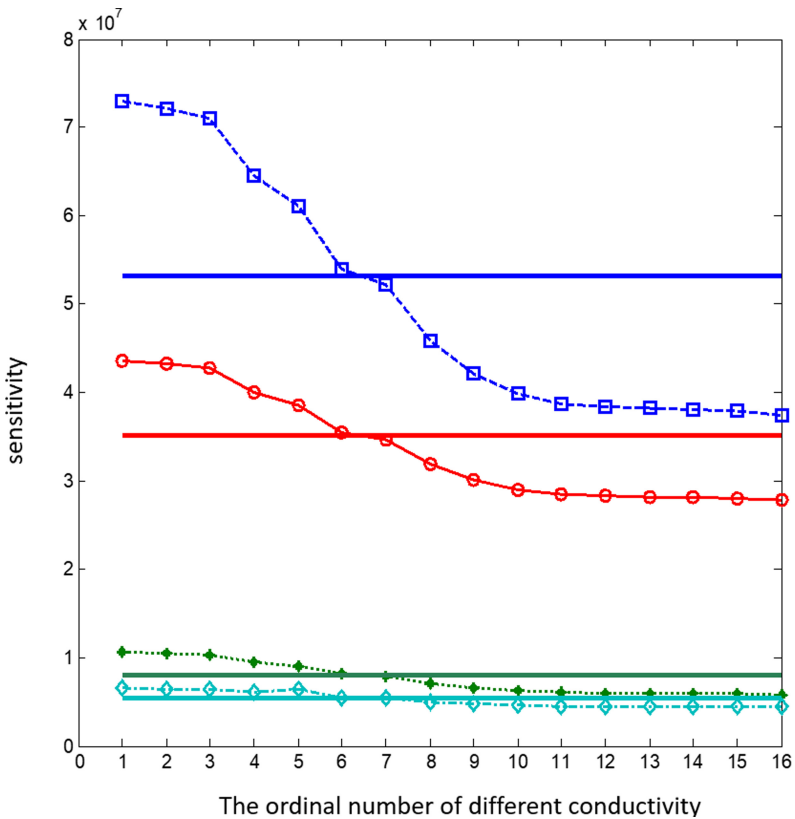


**Fig. 6.** Sensitivity map about one perturbation with  $\sigma = 0.8$  S/m in the middle with shielding with  $\sigma = 1.6$  S/m measurement model with four perturbation for different conductivity.

As can be seen from Fig. 6, figure (a)–(e) is very similar, that is, when the conductivity of the edge disturbance is less than 0.08 S/m, the sensitivity of the whole region is not clearly distinguished, and the sensitivity of the edge disturbance close to the coil

is slightly greater than that far away from the coil. The appearance of the edge disturbance drives the sensitivity close to the coil. Figure (f) and (g) are similar. Although the conductivity of the edge disturbance is similar to that of the background, it can be distinguished due to the presence of shielding, and the sensitivity of the region near the center is slightly higher than that near the edge. Figure (h)–(p) is very similar, and the overall sensitivity is improved when it is less than 0.5 S/m. In particular, the two disturbance regions far away from the coil form a highly sensitive region with the disturbance in the middle.

Figure 7 shows the center sensitivity corresponding to different disturbance conductivity values, including four cases of no edge disturbance in the target region and four cases corresponding to the addition of edge disturbance.



**Fig. 7.** Sensitivity in the center for different conductivities. (The ordinal 1–16 corresponding to 0.010 S/m, 0.005 S/m, 0.01 S/m, 0.05 S/m, 0.08 S/m, 0.18 S/m, 0.22 S/m, 0.5 S/m, 1 S/m, 2 S/m, 4 S/m, 5 S/m, 6 S/m, 8 S/m, 10 S/m, 100 S/m respectively)

As you can see in Fig. 7, the uniform background conditions, with the edge of the four small circular disturbance conductivity value increases, the sensitivity of the detection area center is gradually reducing, the maximum value appeared on the disturbance

of 0.001 S/m. When the edge of the conductivity value is less than even background conductivity 0.2 S/m, the center is greater than the sensitivity without disturbance when the value of the edge, but greater than 0.2 S/m, the center of the sensitivity value is less than the undisturbed. Therefore, the sensitivity of the central region can be improved by adding edge disturbance to the detection area, but the added disturbance must be less than the background conductivity, otherwise the sensitivity will become lower, and when the edge conductivity is less than 0.01 S/m, the increase of the sensitivity will be very small. For a uniform background conductivity of 0.2 S/m, the addition of an edge disturbance of 0.01 is a good choice, increasing approximately 33.9%. For the case where the intermediate disturbance is added, the variation rule of the central sensitivity is basically the same as that of the uniform background, but the variation range is smaller. After four edge disturbances of 0.01 S/m were added, the intermediate sensitivity was improved by 29.6%. For edge and thin shield, the change law and uniform background are similar, but with the advent of shielding, slightly smaller than its center of sensitivity value changes in homogeneous background. For the both center disturbance and the edge block's case, the overall change area remains the same, but the smaller amplitude, the maximum perturbation on the fringe of 0.001 S/m.

## 4 Conclusion

Overall, four kinds of cases, the uniform background 0.2 S/m for setting the edge disturbance of watershed, when conductivity is greater than the background, adding the edge disturbance instead reduces the sensitivity, when less than the background conductivity, the center has improved sensitivity, and the approximate inverse proportion change appears, however, when the conductivity small to 0.01 S/m, while reducing the value of the conductivity of edge can also improve the sensitivity, but increased modestly, background of 0.2 S/m on the edge of the disturbance of the conductivity set to 0.01 S/m. Under uniform background, if the edge is added with a thin shielding with high conductivity, the center sensitivity should be reduced, but if the detection area is added with edge disturbance, the center sensitivity can be improved. When a circular disturbance is detected in the center of the region, the addition of edge disturbances is still improved, but the amplitude is not so obvious.

Magnetic induction imaging is a new technique which can be used in the noninvasive and noninvasive biological tissue. However, due to the existing problems, this technique has not been used in the actual medical treatment, and is limited to the laboratory model research. There is still a wide space for further discussion and research in terms of both detection signals and image reconstruction. MIT's research work can be further carried out from the following aspects. Reconstruction image can't reflect well the measured object is one of the key factors of the amount of data is not enough, limited to the space is limited, and the number of detection sensor will be restricted, how to increase in the number of objective conditions in limited Numbers, and can guarantee the stability of the measurement of the MIT to practical application must solve the problem. Detecting coil sensor signal detection by the commonly used two methods detect amplitude and phase detection, because of MIT's own signal, the characteristics of current phase detection method, mainly due to the characteristics of the eddy current signal itself in the MIT,

high requirement to the phase detection device, so the cost and technical requirements are very strict, in as much as possible to reduce cost and technical difficulties, on the basis of development is suitable for the high precision phase MIT detection device is particularly important.

## References

1. Hollaus, K., Magele, C., Merwa, R., Scharfetter, H.: Numerical simulation of the eddy current problem in magnetic induction tomography for biomedical applications by edge. *Elements* **40**(2), 623–626 (2004)
2. Merwa, R., Scharfetter, H.: Magnetic Induction Tomography: a feasibility study of brain oedema detection using a finite element human head model. In: Scharfetter, H., Merwa, R. (eds.) 13th International Conference on Electrical Bioimpedance and the 8th Conference on Electrical Impedance Tomography. IFMBE Proceedings, vol. 17, pp. 480–483. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-73841-1\\_124](https://doi.org/10.1007/978-3-540-73841-1_124)
3. Zolgharni, M., Ledger, P.D., Griffiths, H.: Forward modelling of magnetic induction tomography: a sensitivity study for detecting haemorrhagic cerebral stroke. *Med. Biol. Eng. Comput.* **47**, 1301 (2009). <https://doi.org/10.1007/s11517-009-0541-1>
4. Zolgharni, M., Ledger, P.D., Armitage, D.W., Holder, D.S., Griffiths, H.: Imaging cerebral haemorrhage with magnetic, induction tomography: numerical modeling. *Physiol. Meas.* **47**(12), 1301–1304 (2009)
5. Zolgharni, M., Griffiths, H., Ledger, P.D.: Frequency-difference MIT imaging of cerebral haemorrhage with a hemispherical coil array: numerical modeling. *Physiol. Meas.* **31**(8), 111–114 (2010)
6. Dekdouk, B., Ktistis, C., Armitage, D.W., Peyton, A.J.: Assessing the feasibility of detecting a Hemorrhagic type stroke using a 16 channel Magnetic Induction System. *J. Phys: Conf. Ser.* **222**(1), 1–4 (2010)
7. Dekdouk, B., Yin, W.L., Ktistis, C., Peyton, A.J.: A method to solve the forward problem in magnetic induction tomography based on the weakly coupled field approximation. *IEEE Trans. Bio-med. Eng.* **57**(4), 914–921 (2010)
8. Feldkamp, J.R., Quirk, S.: Coil geometry effects on scanning single coil magnetic induction tomography. *Phys. Med. Biol.* **62**, 7097–7113 (2016)
9. Caeiros, J., Martins, R.C., Gil, B.: A new image reconstruction algorithm for real-time monitoring of conductivity and permeability changes in magnetic induction tomography. In: 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 6239–6242 (2012)
10. Xiao, Z.L., Tan, C., Dong, F.: Brain tissue based sensitivity matrix in hemorrhage imaging by magnetic induction tomography. In: 2017 IEEE International and Measurement Technology Conference (2017)
11. Xiao, Z.L., Tan, C., Dong, F.: Effect of inter-tissue inductive coupling on multi-frequency imaging of intracranial hemorrhage by magnetic induction tomography. *Measure. Sci. Technol.* **28**(8), 084001 (2017)
12. Stawichi, K., Gratkowske, S., Komorowski, M., Pietruszewicz, T.: A new transducer for magnetic induction tomography. *IEEE Trans. Magn.* **45**(3), 1832–1835 (2009)
13. Riedel, C., Golombeck, M., Von Saint-George, M., Dossel, O.: Data acquisition system for contact-free conductivity measurement of biological tissue. In: Proceedings of the IFMBE:EMBE, Vienna, Austria, pp. 86–87 (2002)
14. Geddes, L.A., Baker, L.E.: The specific resistance of biological material - a compendium of data for the biomedical engineer and physiologist. *Med. Biol. Eng.* **5**(1), 271–293 (1967)



15. Netz, J., Forner, E., Haagemann, S.: Contactless impedance measurement by magnetic induction - a possible method for investigation of brain impedance. *Physiol. Meas.* **14**(1), 263–471 (1993)
16. Zolgharni, M., Ledger, P.D., Armitage, D.W., Holder, D.S., Griffiths, H.: Imaging cerebral haemorrhage with magnetic, induction tomography: numerical modeling. *Physiol. Measure.* **30**(6), 187–200 (2009)
17. Yin, W., Peyton, A.J.: Sensitivity formulation including velocity effects for electromagnetic induction system. *IEEE Trans Instrum. Measure.* **46**(5), 1172–1175 (2010)
18. Roth, Y., et al.: Transcranial magnetic stimulation of deep brain region: principles and methods. *Adv. Bio. Psychiatry* **23**, 204–224 (2007)
19. Wagner, T., et al.: Non-invasive human brain stimulation. *Ann. Rev. Biomed. Eng.* **9**, 204–224 (2007)
20. Christ, A., et al.: The virtual family—development of surface-based anatomical models of two adults and two children for dosimetric simulations. *Phys. Med. Biol.* **55**, 23–38 (2010)
21. Korjensky, A.V., Cherepenin, V.A.: Measuring system for magnetic induction tomography. In: *Proceedings of the 10th International Conference on Electrical Bio-impedance*, Barcelona, pp. 365–368 (1998)
22. Scharfetter, H., Lackner, H.K., Rosell, J.: Magnetic induction tomography: hardware for multi-frequency measurements in biological tissues. *Physiol. Meas.* **22**, 131–146 (2001)
23. Horeh, L., Gilad, O., Romsauerova, A., Mcewan, A., Arridge, S.R., Holder, D.S.: Stroke type differentiation by multi-frequency electrical impedance tomography – a feasibility study (2005)
24. Gencer, N.G., Tek, M.N.: Electrical conductivity imaging via contactless measurements. *IEEE Trans. Med. Imaging* **18**(7), 617–627 (1999)
25. Dowrich, T., Blochet, C., Holder, D.: In vivo bioimpedance changes during haemorrhagic and ischaemic stroke in rats: towards 3D stroke imaging using electrical impedance tomography. *Physiol. Meas.* **3**, 765–784 (2016)
26. Holder, D.S.: Electrical Impedance Tomography of brain function. In: *2008 World Automation Congress*, pp. 1–6 (2008)



# Pair-Wise Convolution Network with Transformers for Sequential Recommendation

Jiangpeng Shi<sup>1</sup>, Xiaochun Cheng<sup>2</sup>, and Jianfeng Wang<sup>3</sup>(✉)

<sup>1</sup> School of Life Sciences, Shanxi Datong University, Datong, China  
jiangpeng\_shi@163.com

<sup>2</sup> Department of Computer Science, Middlesex University, London, UK  
xiaochun.cheng@gmail.com

<sup>3</sup> School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China  
wjf739@gmail.com

**Abstract.** Sequential recommendations seek to employ the sequence of interactions between users and commodities to predict their next behavior based on the behavior they have recently made. Previously, some recommendation systems have been built on Markov chains and recurrent neural networks (among others). However, these methods have many limitations that they emphasize too much sequence change to fully emphasize the correlation between adjacent items; Besides, they generally ignore the influence of contextual information. To solve the shortcomings of the existing sequential recommendations, we try to model the relationship between items, get an effective representation of sequential features, and capture complex sequence correlations. Specifically, we propose a pair-wise convolution network with transformers for the sequential recommendation. The two-dimensional convolution networks encodes the sequence into a three-dimensional tensor and learns the relationships of features between the sequences. We adopt a residual connection to prevent the gradient from disappearing and solve the loss of feature information. The experimental results show that our method is superior to various advanced sequential models on sparse and dense data sets and different evaluation indicators.

**Keywords:** Sequential recommendation · Convolutional attention network · Pair-wise convolution

## 1 Introduction

In the present information explosion era, there are more and more data on the Internet, which greatly enrich the content of the Internet. Due to the extensive use of internet technologies, social media platforms, and e-commerce systems (such as Tiko, Amazon, or Netflix) are used more and more frequently in people's lives. People's habit of obtaining information from nothing, from

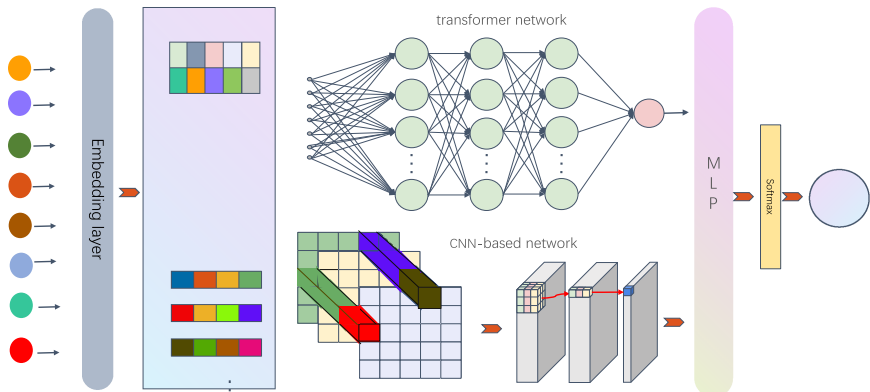
quantitative change to qualitative change, has undergone tremendous changes. At the same time, Internet users not only put forward higher requirements for the speed and real-time information acquisition but also have an increasing demand for personalized and precise search. Therefore, a large number of clicks, purchase interactions, and other user feedback are also generated in these systems. Taking Tiko as an example, users may watch thousands of short videos every week. Faced with massive amounts of information, they have become dazzled and unable to make decisions. It is difficult for people to find the information that is most suitable for them from a large amount of information. The recommendation system has been widely applied in online service to improve the quality of search engines and the accuracy of their result lists. The main purpose of the recommendation system is to infer users' preferences for items based on user interaction behavior and assist users to filter out their desired products. Therefore, the recommendation plays an important role in thereby improving user search efficiency and improving information overload, such that this helps the decision-making process of selecting the best product design for manufacture.

In the past decade, many efforts have been made to develop general recommendation methods, such as model-based methods, deep learning-based methods include Markov models and recurrent neural network models. For example, Liu et al. [7] proposed a new short-term attention/memory priority model, which can obtain the user's general interest from the long-term memory of the conversational context, while considering The user's current interest comes from the short-term memory of the last click. However, this model ignores the dynamic changes in users' long-term preferences over time and often uses static features to model users' long-term preferences. They integrate user-item or item-item interactions in a linear fashion, thereby limiting the capabilities of the model. Ying et al. [13] proposes a two-layer attention network that combines long-term and short-term user interests for comprehensive modeling. Yuan et al. [16] proposed a CNN model to learn item sequence features, add a residual mechanism to the deep CNN, and stitch with user embedding in the last layer, and finally input MLP for prediction. However, the user's interest pattern is personalized. Most of the existing sequence recommendation models usually only consider the user's recent interaction behavior independently, without obtaining the real internal connection between the user's continuous behavior, and most algorithms can only model the user's historical behavior linearly, and it is difficult to have a flexible sequence. Therefore, how to obtain the potential dependencies between users' historical behaviors and how to accurately obtain user interests have become key research issues in personalized recommendation systems.

To solve the above limitations, we propose a pair-wise convolution network with transformers for a sequential recommendation (Fig. 1), which considers the relative relationship between any two items. Specifically, the Transformer-based semantic layer models the user's historical sequential behavior and decodes the target product to obtain the user's preferences; the fusion layer captures the linear and non-linear correlation between the user's preferences and the target

product. To understand user preferences more dynamically, the model introduces users' long-term preferences and current users' short-term interests. This paper believes that each user has different preferences for each commodity. The attention mechanism in transformer automatically assigns weights to items to dynamically capture long-term and short-term interests. At the same time, the layer is designed to adaptively combine long-term and short-term preferences, which pays attention to the different effects of different users on each product. The experimental results show that the performance of this method exceeds that of all baseline models.

A person has a mature and stable value system to see and understand the world, and he has his preferences and judgments about what he comes into contact with, and the behavioral decisions that people usually make are often based on long-term preferences. However, in the real world, many factors affect user decisions. Long-term preferences and short-term preferences together determine people's decision-making behavior. Therefore, the recommendation is based on the long-term preference and the short-term preference as the contextual information. The contributions of this article are summarized as follows: 1. Considering the context information in the sequence table, we make full use of the attention mechanism to make similar items more weighted. The weighted results make them closer and closer in space, modeling users' short-term hobbies, and making a more accurate user experience. Products Recommended. 2. This paper adopts CNN-based network results to capture the relationship between items through the window sliding of the convolution kernel, models the commodity sequence that best reflects what customers need, and analyzes the user's search intention understanding. 3. Considering the time sequence of the user sequence, we use the transformer to model the user's click sequence and make recommendations.



**Fig. 1.** The framework of the proposed method

The remainder of the paper is organized as follows. In Sect. 2, gives an overview of the literature and a critical review of research in the area of this work. In Sect. 3, we present the methods for predicting potential needs and wants, the different aspects and improvements are presented as sections in this chapter, which are value chain for predicting potential customer needs and wants. In Sect. 4, the data analysis and results for each dataset are shown, and the obtained results are summarized. In Sect. 5, the conclusion and future work are discussed.

## 2 Related Work

### 2.1 Sequential Recommendation

More recent attention has focused on the provision of sequential recommendations. The most common one is based on Markov's recommendation, which can effectively capture the local sequence model. Rendle et al. [9] proposed a Factorized Personalized Markov Chains based Markov chain to predict the next item. However, this approach performs best on sparse data sets. He et al. [2] proposed high-order Markov Chains to model pairwise user-item and item-item interactions. Tang et al. [10] models the sequence as a two-dimensional space and adopts 2D CNN to convolute the embedding items to predict the next item. However, the higher-order MC based methods need to be specified rather than being chosen adaptively. In many research methods, many of the methods are based on RNN to predict the next items. Yu et al. [14] proposed a dynamic recurrent basket model based on RNN for the next-basket recommendation. HidasiK et al. [4] proposed to use Gated Recurrent Unit (GRU) to model sequential behavior for the session-based recommendation. Since these RNN-based methods take the state from the last step and current action as their input, these dependencies make RNN-based methods less efficient (i.e., higher model complexity). It may be because these complex models require large amounts of data to capture long-term patterns, i.e., easily overfitting in high-sparsity settings.

### 2.2 Deep Learning Models

Many studies focus on predicting the next item as well as improve services by providing reliable insight into what customers need and want. To date, several studies have investigated the search recommendation algorithm. These studies have made important progress in modeling user behavior sequences. The main research work is based on the convolutional neural network and attention mechanism, such as Caser [10] and NextItNet [15]. Caser [10] as the most typical CNN, however, the standard CNN architecture and max-pooling operation of Caser were not well-suited to model long-range user sequences. NextItNet [15] proposed to model the user interaction-item sequence by stacking CNN layers to increase the receptive field of higher layer neurons, The results show that the NextItNet is more effective than the RNN model to recommend the top-n

items. Based on this, many methods are on the extension. At the same time, the good results are shown by the self-attention model based on the transform in the field of SRS. DIN considers the influence of different items in the behavior sequence on the current predicted item by introducing an attention mechanism; DIEN solves the shortcoming that DIN cannot capture the dynamic changes in user interests. DIN [18] and DIEN [17] do not consider the session information in the user's historical behavior, because the behavior in each session is similar, and the difference between different sessions is very large. The BST [1] model uses the Transformer model to capture the associated features of each item in the user's historical behavior sequence. However, the computational cost of the self-attention mechanism is higher than the CNN structure of the superimposed expansion.

### 3 Method

This paper proposes the pair-wise convolution network with the transformers model, which combines long-term preferences and short-term preferences to provide users with recommendations. This article combines long-term and short-term interest to extract relevant information from users' recent historical interaction behaviors (click, browse, purchase). The system can automatically collect the required data in the implicit feedback scenario, which not only effectively alleviates the problem of data dispersion, but also provides users with a smoother and more comfortable experience.

#### 3.1 Behavior Sequence Layer

At present, deep learning algorithms have made great progress in the recommendation system, and all have good effects such as FM [8], DeepFM [6], Wide & Deep [3,11]. The general method is converted discrete high-dimensional features to fixed-length continuous features through feature embedding operations, and then perform feature extraction through multiple connections, and finally activate function to obtain the predicted recommendation probability. The user's interest characteristics have obvious diversity and local activation. Due to the related characteristics of user interests, if users have different tendencies for different commodities. Therefore, the attention mechanism is introduced to model interest. The most basic interest extraction method is the same as the traditional deep model method, which mainly includes a feature embedding layer, a multilayer perceptron, and an pooling layer and connection layer.

Embedding, as a commonly used vector representation of deep learning, can extract out the multi-dimensional latent features of commodities. The user (ID and its attributes), items (ID and its attributes), and contextual information are uniformly expressed as feature vectors as input to predict the target score value. The data is represented as a feature vector to the target value, each feature vector is represented as a hidden vector, and the interaction between all non-zero features is considered in the hidden space. The neural network can be used

directly. We use word2vec to make a queue with historical data of behavior to form label expansion to get similar items.

In this sequence, we elaborate on the sequence recommendation problem as follows. We mark the set of users as  $U$  and the set of products at  $I$ . For each user, there will be a sequence of items  $S$  corresponding to the user. We embed the user and the product into two matrices, respectively, where  $d$  is the latent dimension representation, then the embedding matrix is expressed as follows:

$$E^{(i,t)} = \begin{bmatrix} e_{S_{t-L}^i} \\ \vdots \\ e_{S_{t-2}^i} \\ e_{S_{t-1}^i} \end{bmatrix} \tag{1}$$

Where  $e_{S_{t-L}^i}$  is the embedding vector. The items and users were embed into two matrices  $E_I \in \mathbb{R}^{|I| \times d}$  and  $E_U \in \mathbb{R}^{|U| \times d}$ , where  $d$  is the latent dimensionality. For user  $u$ , we retrieve the input embedding matrix by looking up the previous  $L$  items in the item embedding matrix. The role aims to transform the original ID behavior sequence into an embedding behavior sequence. Inspired by transformer [1] and the 2D-CNN network [12], we convolve the three-dimensional pair-wise sequence to extract features. The convolution process is as follows,

$$c_i^k = \phi_c (E_{i:i+h-1} \odot F^k) \tag{2}$$

where  $F^k$  is the convolution kernel. The residual network was used to fuse the  $k$  features and the final features are expressed as follows,

$$c^k = [c_1^k c_2^k \cdots c_{L-h+1}^k] \tag{3}$$

We concatenate the outputs of the convolutional layers and feed them into a fully-connected neural network layer to get more high-level and the features  $\tilde{c}^k$ :

$$\tilde{c}^k = \sum_{l=1}^L \tilde{F}_l^k \cdot c^k \tag{4}$$

### 3.2 Interest Extraction Layer

The interest extraction layer is used to extract user interest by simulating the process of user interest migration. It contains a semantic layer and a feature extraction layer, the semantic layer is the main structure of the model, which mainly includes the self-attention based on the multi-head attention mechanism. First, the network learns the dependence of each item in the behavior sequence through self-attention to obtain user semantic characteristics. Then, the feature of the target is decoded to the user’s semantic feature to obtain the user’s

semantic preference by the attention. Next, the semantic layer will be described in detail, such as the following formula.

$$\tilde{m} = \text{self-attention}(m_{in}) \quad (5)$$

where  $\tilde{m}$  represents user semantic features, self-attention represents self-attention mechanism, and  $m_{in}$  represents user's input features. The attention decodes target item features and user semantic features to obtain user semantic preferences  $m_o$ , as follows:

$$m_o = \text{Attention}(I_t, \tilde{m}, \tilde{m}) \quad (6)$$

$$\tilde{m} = \text{Attention}(m_{in}, m_{in}, m_{in}) \quad (7)$$

Among them,  $\tilde{m}$  represents the user's semantic preference.  $I_t$  represents the features of the target item, The attention is formula as:

$$U = \text{Attention}(m_t, \tilde{m}, \tilde{m}) = \text{Softmax}\left(\frac{I_t \tilde{m}^T}{\sqrt{D}}\right) \tilde{m} \quad (8)$$

But there is a multi-head parallel mechanism in the transformer. We connect the attention of each head into multi-head attention, and then the multi-head attention is the following:

$$U = \text{Concat}(head_1, \dots, head_i, \dots, head_n) \quad (9)$$

In order to further increase the nonlinearity of the model, we adopt a feedforward neural network. It is defined as follows:

$$\begin{aligned} m_o &= FFN(U_o) \\ &= \text{Normalize}(\text{Conv1D}(\text{Conv1D}(m_o)) + m_o) \end{aligned} \quad (10)$$

Normalize means normalization to solve the problem of vanishing gradient, Conv1D means a one-dimensional volume network. The two-layer convolutional neural network performs two nonlinear mappings of  $m_o$ . At the same time, to prevent the loss of original information, a residual connection method is adopted.

### 3.3 Interest Fusion Layer

In order to provide better generalization capabilities for the entire model, this article adds a fusion layer to the semantic layer. The fusion layer learns the correlation between the target product features generated from the embedding layer and the user's semantic preferences obtained from the semantic layer and merges the two models through the output of the last hidden layer of the multi-layer perceptron. Therefore, at the fusion layer, this article uses a simple linear function to capture the interaction between user preferences and item features:

$$\begin{aligned} z_0 &= \text{Concat}(m_o, I_t) \\ z_1 &= \text{ReLU}(W_1^T z_0 + b_1) \\ &\dots \\ h_2 = z_l &= \text{ReLU}(W_l^T z_{l-1} + b_l) \end{aligned} \quad (11)$$



Among them,  $W_l$  and  $b_l$  are respectively the weight matrix and offset vector of the hidden layer of the  $i$ -th layer. Combine the feature vector of the product and the output of the multilayer perceptron as the input of the output layer:

$$h = \text{Concat}(h_1, h_2) \hat{y} = W' \begin{bmatrix} h \\ m_o \end{bmatrix} + b' \quad (12)$$

The main function is to stimulate the interest evolution process related to the current target advertisement by adding an attention mechanism based on the interest extraction layer. Recently developed methods focus on designing structures of MLP for better information extraction. The objective function used in the base model is the cross-entropy loss function defined as:

$$L = -\frac{1}{N} \sum_{(x,y) \in S} (y \log \hat{y} + (1-y) \log(1-\hat{y})) \quad (13)$$

where  $S$  is the training set of size  $N$ ,  $y \in 0, 1$  as the label,  $\hat{y}$  is the output of the network after the softmax layer, representing the predicted probability of sample being clicked.

## 4 Experiment

### 4.1 Datasets and Experiment Data

To verify the effective performance of our method, we adopted two basic data sets for the experimental method. **Gowalla**: this website is a social network website with time frame sequence. This data set contains implicit feedback through user-venue check-ins. **MovieLens**: this data set is a collaborative filtering algorithm widely used in recommendation systems. We use the ml1m data set, which consist of one million user ratings to verify the performance of our algorithm.

The method of processing data is similar to the previous method [12]. For all data sets, we regard user comments and ratings as implicit feedback of users, and divide behavior sequences according to time frames. During the processing, the products and users were discarded that fewer than five related behavioral. We divide the data set into training set, validation set and test set. Among them, in the behavior sequence, the most recent action is used as the test set, the second most recent action is used as the verification set, and the rest is used as the training set. The processed data set is shown in Table 1.

**Table 1.** Statistics of the datasets.

Dataset	#users	#items	avg. #act. per user	avg. #act. per item	#actions
ML-1M	6.0K	3.4K	165.50	292.06	0.993M
Gowalla	13.1K	14.0K	40.74	38.12	0.533M

## 4.2 Comparison Methods

We have selected the following mainstream sequence recommendation algorithms as our comparison algorithm.

Factorized Markov Chains (FMC) [9]: Based on the Markov chain algorithm, we use two product embeddings to decompose the product conversion matrix, and generate a recommendation sequence based on the last viewed product,

Factorized Personalized Markov Chains (FPMC) [9]: It is an extension based on the Markov model. FPMC sets up a Markov chain for each shopping sequence. The established Markov chain not only learn the long-term preferences of users but also capture the demands of user that provide personalized recommendations for each user. Therefore, this model has a strong advantage in modeling sequences.

GRU4REC [5]: The method used recurrent neural networks for conversation recommendation tasks, using recurrent neural networks to model conversation sequences. The method treats each user's feedback sequence as a session.

Convolutional Sequence Embeddings (Caser) [10]: The first application of convolutional neural network in sequence model, convolution from the vertical and horizontal dimensions respectively to capture high-order Markov chains, this method has achieved good results.

Convolutional Neural Networks (CosRec) [12]: The sequence between items is encoded into a three-dimensional tensor in a pair-wise manner, and a 2D convolutional neural network is used to learn local features.

## 4.3 Implementation Details

The data preprocessing and training in this article run on the Ubuntu operating system, the graphics card is NVIDIA GTX 1080, the memory is 32 GB, and the integrated development environment is PyCharm. The data preprocessing mainly is Python 3.7, and the related extension libraries are Numpy, Pickle scipy, etc. to support large-scale file data to reading and simple matrix operations. At the same time, in order to further improve training efficiency, the open-source deep learning framework is pytorch1.1 during training, and CUDA10.0 was introduced to the GPU acceleration.

We used 10 convolutional blocks. For the ml1m dataset, the dimension of  $d$  is set to 20, the length of  $l$  is 10, and  $T$  is set to 2. For the gowalla dataset, the dimension of  $d$  is set to 100, and the length of  $l$  is 10,  $T$  is set to 3. We use two self-attention blocks. The optimizer is the adam optimizer, the learning rate is set to 0.000005, and the batch size is 512. The dropout rate of turning off neurons is 0.2 for MovieLens-1m and 0.5 for the other datasets due to their sparsity.

## 4.4 Evaluation Metrics

The precision rate indicates how many positive samples are predicted correctly among the samples whose predictions are positive, and the recall rate indicates

how many positive samples are correctly predicted among all the positive samples. We evaluate our model with  $\text{precision@n}$ ,  $\text{recall@n}$  and mean average precision, as shown in the Formula 14 and Formula 15.

$$\text{Prec@N} = \frac{|R \cap \hat{R}_{1:N}|}{N} \quad (14)$$

$$\text{Recall@N} = \frac{|R \cap \hat{R}_{1:N}|}{|R|} \quad (15)$$

#### 4.5 Recommendation Performance

Table 2 shows the experimental results of five comparison methods and our algorithm on two data sets. It can be concluded from the table that our algorithm obtains the best performance on all data sets, which shows the effectiveness of the proposed algorithm. In addition, from observing the comparison method, it is found that FMC and FPMC perform poorly on some data sets. Both FMC and FPMC deal with fully parameterized transition graphs, and their premise is that users and commodities have independent parameters, and the calculation of each parameter does not consider the influence of other parameters. However, in the personalized recommendation process, we need to decompose the transferred three-dimensional matrix to break the independence between parameters and estimation, so that the mutual influence between similar users, products, and transfer situations can be considered. At the same time, it is observed that

**Table 2.** Performance comparison on the four data sets.

Dataset	Metric	FMC	FPMC	GRU4Rec	Caser	CosRec	Ours
<i>ML - 1M</i>	MAP	0.0687	0.1053	0.1440	0.1507	0.1883	<b>0.1895</b>
	Prec@1	0.1280	0.2022	0.2515	0.2502	0.3308	<b>0.3308</b>
	Prec@5	0.1113	0.1659	0.2146	0.2175	0.2831	<b>0.2818</b>
	Prec@10	0.1011	0.1460	0.1916	0.1991	0.2493	<b>0.2506</b>
	Recall@1	0.0050	0.0118	0.0153	0.0148	0.0202	<b>0.0204</b>
	Recall@5	0.0213	0.0468	0.0629	0.0632	0.0843	<b>0.0837</b>
	Recall@10	0.0375	0.0777	0.1093	0.1121	0.1438	<b>0.1438</b>
Gowalla	MAP	0.0229	0.0764	0.0580	0.0928	0.0980	<b>0.1894</b>
	Prec@1	0.0517	0.1555	0.1050	0.1961	0.2135	<b>0.3374</b>
	Prec@5	0.0362	0.0936	0.0721	0.1129	0.1190	<b>0.2790</b>
	Prec@10	0.0281	0.0698	0.0782	0.0571	0.0884	<b>0.2473</b>
	Recall@1	0.0064	0.0256	0.0155	0.0310	0.0337	<b>0.0206</b>
	Recall@5	0.0257	0.0722	0.0529	0.0845	0.0890	<b>0.0834</b>
	Recall@10	0.0402	0.1059	0.0826	0.1223	0.1305	<b>0.1436</b>

the performance of recommendation algorithms based on convolutional neural networks (Caser and CosRec) is better than traditional recommendation algorithms (FMC and FPMC), which shows that CNN-based can effectively model the interaction between users and items. Compared with Coser and our method, our algorithm has better performance, which shows that the attention mechanism is effective for mining the short-term and long-term intentions of historical users for modeling user and project interaction. Compared with all recommendation methods, the performance of our algorithm is better in solving the dynamic attention mechanism. Our model will assign different weights to different candidate items according to the user's interaction sequence and the user's historical score. Items are often highly related to candidate items. As can be seen from the data in the table, in the user interaction sequence, the attention mechanism can assign different weights to different products, which is higher than other types of product activation weights.

Figure 2 and Fig. 3 show the performance of our method on Gowalla and ml1m datasets, respectively. In each figure, the horizontal axis represents the number of training sessions, and each table shows the performance indicators of six tests. We can see from Fig. 2 that as  $N$  increases, precision gradually decreases. In addition, the performance reaches its best around twenty-eight rounds, and the performance will not increase as the number of rounds increases. Recall and precision are just the opposite. With the increase of  $N$ , the recall gradually becomes smaller. When the number of rounds reaches fourteen, the recall will not increase and the performance tends to be stable. Figure 3 shows the performance of our method on the ml1m data set. In general, the performance increases with the increase in the number of rounds. Although there are

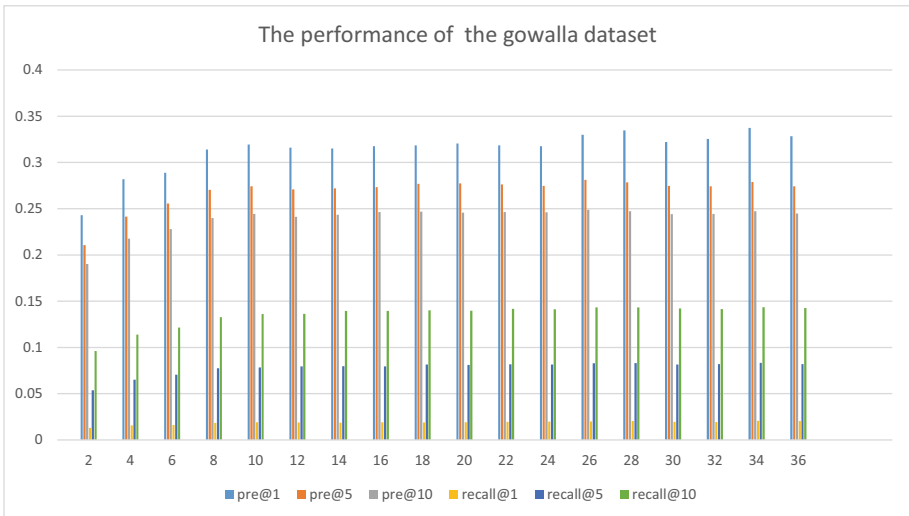


Fig. 2. The performance of gowalla data with various epoch.

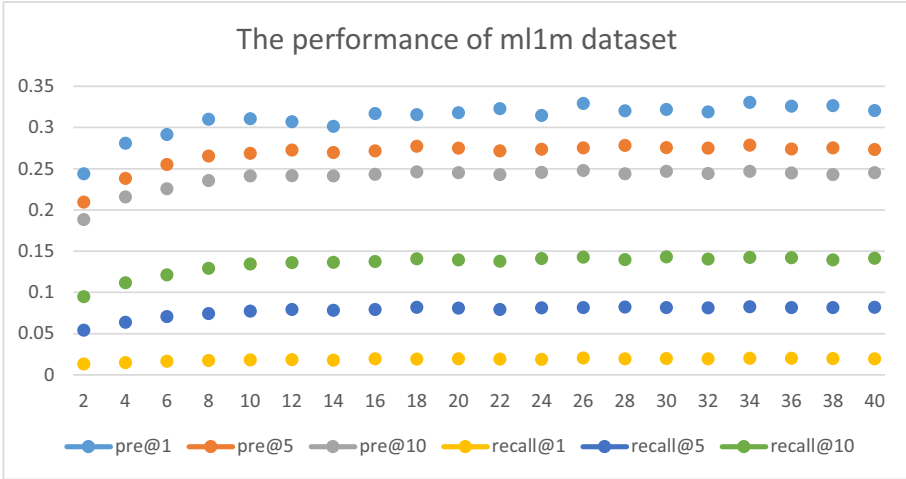


Fig. 3. The performance of gowalla data with various epoch.

fluctuations in a small range, the overall performance tends to be stable. The performance leveled off after the 26-th round. From the performance of the total extraction volume on the two data sets of Fig. 2 and Fig. 3, the precision increases with the increase of  $N$ , and the recall decreases with the increase of  $N$ .

## 5 Conclusion

This paper analyzes the personalized recommendation model. In order to make recommendations for users' hobbies more accurately, the current more accurate CNN-based network is selected as the basis for optimization and improvement, and finally, a feature-based weight extraction is proposed. In terms of data analysis and processing, this article adopts a public data set. The data is first analyzed for the defect value, and the original data is sampled and balanced to ensure the accuracy of later training. This paper designs a new feature weight extraction model, which assigns different weights to different important items in the user interaction sequence, and more accurately simulates the user's real purchase situation. In this paper, a corresponding feature weight extraction module is designed to further enhance the accuracy of model recommendation. Based on the original deep interest network, the corresponding advantages of the feature weight extraction model are introduced, and a more accurate recommendation effect is achieved through experimental verification. In the future research process, we will study multi-modal recommendation, take into account the multi-semantic user's intention understanding of the combination of text and pictures, and combine the current research situation to continue to research.

## References

1. Chen, Q., Zhao, H., Li, W., Huang, P., Ou, W.: Behavior sequence transformer for e-commerce recommendation in Alibaba. CoRR abs/1905.06874 (2019)
2. He, R., McAuley, J.J.: Fusing similarity models with Markov chains for sparse sequential recommendation. In: ICDM, pp. 191–200 (2016)
3. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.: Neural collaborative filtering (2017)
4. Hidasi, B., Karatzoglou, A.: Recurrent neural networks with top-k gains for session-based recommendations. In: CIKM, pp. 843–852 (2018)
5. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. In: ICLR (2016)
6. Liu, F., Guo, W., Guo, H., Tang, R., Ye, Y., He, X.: Dual-attentional factorization-machines based neural network for user response prediction. In: WWW, pp. 26–27 (2020)
7. Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: STAMP: short-term attention/memory priority model for session-based recommendation. In: KDD, pp. 1831–1839 (2018)
8. Mao, X., Mitra, S., Swaminathan, V.: Feature selection for FM-based context-aware recommendation systems. In: ISM, pp. 252–255. IEEE Computer Society (2017)
9. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: WWW, pp. 811–820 (2010)
10. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: WSDM, pp. 565–573 (2018)
11. Xu, J., Shi, J., Yao, Y., Zheng, S., Xu, B., Xu, B.: Hierarchical memory networks for answer selection on unknown words. In: Calzolari, N., Matsumoto, Y., Prasad, R. (eds.) COLING, pp. 2290–2299. ACL (2016)
12. Yan, A., Cheng, S., Kang, W., Wan, M., McAuley, J.J.: CosRec: 2D convolutional neural networks for sequential recommendation. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, 3–7 November 2019, pp. 2173–2176 (2019)
13. Ying, H., et al.: Sequential recommender system based on hierarchical attention networks. In: IJCAI, pp. 3926–3932 (2018)
14. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A dynamic recurrent model for next basket recommendation. In: SIGIR, pp. 729–732 (2016)
15. Yuan, F., et al.: Future data helps training: modeling future contexts for session-based recommendation. In: WWW, pp. 303–313 (2020)
16. Yuan, F., Karatzoglou, A., Arapakis, I., Jose, J.M., He, X.: A simple convolutional generative network for next item recommendation. In: Culpepper, J.S., Moffat, A., Bennett, P.N., Lerman, K. (eds.) WSDM, pp. 582–590. ACM (2019)
17. Zhou, G., et al.: Deep interest evolution network for click-through rate prediction. In: AAAI, pp. 5941–5948 (2019)
18. Zhou, G., et al.: Deep interest network for click-through rate prediction. CoRR abs/1706.06978 (2017)

## Author Index

- Bai, Yang 407
- Chang, Liang 340
- Chen, Deng 355
- Chen, Jinyin 72
- Chen, Meng 263
- Chen, Tong 120
- Cheng, Xiaochun 433
- Cheng, Zishuai 20
- Choo, Kim-Kwang Raymond 3
- Cui, Baojiang 20
- Deng, Guishi 407
- Duan, Gonghao 355
- Feng, Shuo 245
- Fenggang, Lai 62
- Fu, Junsong 20
- Fu, Shaojing 97, 278
- Gang, Zhao 224
- Gao, Min 164
- Guo, Jie 203
- Guo, Yue 318
- Guo, Yun 318
- Han, Jianmin 329, 369
- Hang, Tianchu 407
- He, Debiao 3
- He, Liqiang 178
- He, Yifei 340
- Hu, Guyu 85
- Hu, Zhaolong 329, 369
- Huang, Xinyu 340
- Huo, Zhixin 263
- Iqbal, Muhammad Munwar 132
- Jay Guo, Y. 189
- Jia, Chunfu 41
- Jia, Nan 278
- Jiang, Haifeng 153
- Jiayu, Hou 379
- Jing, Du 62
- Jing, Zhengjun 189
- Ju, Jianping 355
- Jun-Feng, Tian 109
- Kaiyong, Li 379
- Kan, Zhe 369
- Khadam, Umair 132
- Kuang, Xiaohui 143, 234, 245, 254
- Li, Chenyang 340
- Li, He 120
- Li, Li 3, 387
- Li, Long 120
- Li, Min 318
- Li, Peng 303
- Li, Qing 153
- Li, Shuhao 34
- Li, Shuqiu 395
- Li, Yuxi 303
- Li, Zheng 41
- Lianfang, Wang 224
- Lin, Xiang 72
- Liu, Can 329
- Liu, Chong 153
- Liu, Jiqiang 120, 340
- Liu, Lu 234
- Liu, Ren Ping 189
- Liu, Tianyu 215
- Liu, Wei 355
- Liu, Yanhong 263
- Lou, Kaihao 395
- Lu, Jianfeng 329
- Lu, Jingwen 203
- Lu, Liu 224
- Lu, Qian 52
- Lu, Xiuqing 52
- Luo, Min 3, 387
- Luo, Yuchuan 97
- Lv, Yi 420
- Ma, Chao 215
- Mostarda, Leonardo 132

- Niu, Wenjia 120, 340  
 Ouyang, Wenlei 41  
 Pan, Zhang 62  
 Pang, Ling 254  
 Peng, Hao 329, 369  
 Peng, Xiaowei 52  
 Qi, Zhihan 178  
 Qiao, Xueming 263  
 Qiu, Weidong 203  
 Ren, Xiaorui 34  
 Rui, Kong 62  
 Rui-Fang, Guo 109  
 Shao, Lishong 387  
 Shi, Jiangpeng 433  
 Song, Bo 189  
 Song, Minglu 120  
 Sun, Ruiqi 263  
 Tan, Yu-an 245  
 Tang, Chaogang 153  
 Tang, Jianyin 355  
 Tang, Peng 203  
 Tong, Endong 120, 340  
 Ullah, Farhan 132  
 Wang, Bingyan 34  
 Wang, Dong 215  
 Wang, Hao 203  
 Wang, Jianfeng 433  
 Wang, Jing 387  
 Wang, Juan 245  
 Wang, Shuoru 120  
 Wang, Sikai 355  
 Wang, Yahui 41  
 Wang, Zhifei 178  
 Wang, Zhiqiang 34  
 Wei, Fengqi 178  
 Wei, Wei 355  
 Wei, Xianglin 85, 153  
 Wu, Huaming 153  
 Wu, Zhi 215  
 Xian, Hequn 52  
 Xiao, Yu 62  
 Xiaohui, Kuang 224  
 Xie, Shengxu 85  
 Xing, Changyou 85  
 Xu, Gang 178  
 Xu, Jian 303  
 Xu, Li 164  
 Xu, Ming 278  
 Xu, Zifeng 303  
 Yan, Ming 234  
 Yang, Funing 395  
 Yang, Jin 355  
 Yang, Kai 254  
 Yang, Luming 97  
 Yang, Tao 34  
 Ye, Wang 224  
 Yi, Zhou 62  
 Ying, Ma 379  
 Yu, Jianping 143  
 Yu, Xiao 245  
 Yuan, Ke 41  
 Zeng, Yingming 41, 97, 318  
 Zhang, Bo 234, 254  
 Zhang, Dunjie 72  
 Zhang, Fengqian 289  
 Zhang, Guomin 85  
 Zhang, Hongjie 387  
 Zhang, Ruyun 143, 234  
 Zhang, Xiangkun 263  
 Zhang, Xingliang 395  
 Zhang, Xinxin 164  
 Zhang, Yanan 215  
 Zhang, Yanduo 355  
 Zhang, Yuan 289  
 Zhang, Yunru 3  
 Zhang, Zihang 355  
 Zhangchi, Ying 62  
 Zhao, Dandan 329, 369  
 Zhao, Gang 143, 234, 245, 254  
 Zhao, Quanyu 289  
 Zheng, Chaohui 355  
 Zhou, Fucai 303  
 Zhou, Qian 189  
 Zhou, Ying 340  
 Zhu, Dongjie 263  
 Zhu, Weiyi 263