Lecture Notes in Electrical Engineering 1112

Ji Su Park
Hiroyuki Takizawa
Hong Shen
James J. Park *Editors*

Parallel and Distributed Computing, Applications and Technologies

Proceedings of PDCAT 2023

Springer

Lecture Notes in Electrical Engineering 1112

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering—quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Editorial Director (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** This series is indexed by EI Compendex and Scopus databases. ****

Ji Su Park · Hiroyuki Takizawa · Hong Shen ·
James J. Park
Editors

Parallel and Distributed Computing, Applications and Technologies

Proceedings of PDCAT 2023

Springer

Editors

Ji Su Park
Department of Computer Science
and Engineering
Jeonju University
Cheonjam-ro, Korea (Republic of)

Hong Shen
School of Applied Sciences
Macao Polytechnic University
Macao SAR, China

Hiroyuki Takizawa
Cyberscience Center
Tohoku University
Sendai, Japan

James J. Park
Department of Computer Science
and Engineering
Seoul National University of Science
and Technology
Seoul, Korea (Republic of)

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-99-8210-3 ISBN 978-981-99-8211-0 (eBook)
https://doi.org/10.1007/978-981-99-8211-0

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

Paper in this product is recyclable.

Message from the PDCAT 2023 General Chair

The 24th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT) is a major forum for scientists, engineers, and practitioners throughout the world to present their latest research, results, ideas, developments, and applications in all areas of parallel and distributed computing. Beginning in Hong Kong in 2000, PDCAT 2023 will be held in Jeju, Korea, after 24 years of successful journey through various countries/regions including Taiwan, Japan, China, Singapore, Australia, New Zealand, and Korea across Asia-Oceania. We are inviting new and unpublished papers.

The conference papers included in the proceedings cover the following topics: PDCAT of Networking and Architectures, Software Systems and Technologies, Algorithms and Applications, and Security and Privacy. Accepted and presented papers highlight new trends and challenges of Parallel and Distributed Computing, Applications and Technologies. We hope readers will find these results useful and inspiring for their future research. Our special thanks go to the Program Chairs: Ji Su Park (Jeonju University, Korea), Hiroyuki Takizawa (Tohoku University, Japan), Hui Tian (Griffith University, Australia), and all Program Committee members and all reviewers for their valuable efforts in the review process that helped us to guarantee the highest quality of the selected papers for the conference.

James J. Park
Hong Shen
PDCAT 2023 General Chairs

Organization

Honorary Chair

Doo-soon Park	SoonChunHyang University, Korea

General Chairs

James Park	SeoulTech, Korea
Hong Shen	Sun Yat-sen University, China

Program Chairs

Ji Su Park	Jeonju University, Korea
Hiroyuki Takizawa	Tohoku University, Japan
Hui Tian	Griffith University, Australia

Workshop Chairs

Michael Hwa Young Jeong	Kyung Hee University, Korea
Neil Y. Yen	The University of Aizu, Japan

Publicity Chairs

Sushil Kumar Singh	SeoulTech, Korea
Yong Zhang	Shenzhen Inst. of Adv. Tech., China
Byeong-Seok Shin	Inha University, Korea
Kwang-il Hwang	Incheon National University, Korea
Joon-Min Gil	Daegu Catholic University, Korea
Deok-Gyu Lee	Seowon University, Korea
Byoungwook Kim	Gangneung-Wonju National University, Korea

Local Arrangement Chairs

Yan Li	Inha University, Korea
Yeongwook Yang	Hanshin University, Korea

Registration and Finance Chair

Jungho Kang	Baewha Women's University, Korea

Program Committee

Yuebin Bai	Beihang University, China
Raj Bayyar	University of Melbourne, Australia
Guoliang Chen	U. of Sci. & Tech. of China, China
Lin Chen	Sun Yat-sen University, China
Yawen Chen	Otago University, New Zealand
Luo Cao	Wuyi University, China
Huanwei He	Zhongshan College, China
Zhenxiong Hou	Northwestern Polytechnical University, China
Shi-Jin Horng	Nat. Taiwan U. of Sci. and Tech, China
Longkun Guo	Fuzhou University, China
Mingyu Guo	University of Adelaide, Australia
Mirjana Ivanovic	Univ. of Novi Sad, Serbia
Teofilo Gonzalez	Univ. of Calif. Santa Barbara, USA
Ajay Gupta	Univ. of Western Michigan, USA
Huaxi Gu	Xidian University, China
Francis Lau	Univ. of Hong Kong, China
Shuangjuan Li	South China U. of Agriculture, China
Yidong Li	Beijing Jiaotong Univ, China
Yamin Li	Hosei University, Japan
Weifa Liang	Australian National Univ, Australia
Manu Malek	Stevens Institute of Technology, USA
Rui Mao	Shenzhen University, China
Depei Qian	Beihang University, China
Yingpeng Sang	Sun Yat-Sen University, China
Michael Sheng	Macquarie University, Australia
Jigang Wu	Guangdong Univ. of Tech., China
Chengzhong Xu	University of Macau, China
Jingling Xue	Univ. of New South Wales, Australia
Haibo Zhang	Otago University, New Zealand

Yong Zhang	Shenzhen Inst. of Adv. Tech, China
Zonghua Zhang	Huawei France, France
Xiaofan Zhao	Police University of China, China
Cheng Zhong	Guangxi University, China
Albert Zomaya	University of Sydney, Australia
Dongwann Kang	SeoulTech, Korea
Jun-Ho Huh	Korea Maritime and Ocean University, Korea
Alireza Souri	Department of Computer Engineering, Iran
Ka Man	Xi'an Jiaotong-Liverpool University, China
Vimal Shanmuganathan	Ramco Institute of Technology, India
JongBeom Lim	Pyeongtaek University, Korea
Jaehwa Chung	Korea National Open University, Korea
Yeong-Seok Seo	Yeungnam University, Korea

Contents

A Blockchain System for Fake News Detection

Janusz Bobulski(✉)

Department of Computer Science, Czestochowa University of Technology, Czestochowa, Poland
januszb@icis.pcz.pl

Abstract. The media greatly impacts the world's perception and what is happening in it. Nowadays, the Internet is a medium where information flowing from all over the world meets. Every year, more and more information websites are created from which we learn about the reality surrounding us. The problem is that many of these websites do not use primary and reliable sources but only use sources that have already been processed and duplicated. Very often, these portals omit not only important facts for a given event but also reproduce false information. There are also situations where fake news is deliberately created and disseminated. The phenomenon called "fake news" has long been known as disinformation or propaganda. Contemporary scientific analyses focus primarily on its new dimension, i.e. the dissemination of untrue or incorrect information on the Internet, the reasons for the popularity of fake news on the Internet, and the speed and manner of information dissemination. However, the most important problem is preventing the spread of fake news. In the article, the authors propose a method of authorising content using blockchain technology. The presented method is resistant to manipulation attempts, confirmed by the tests.

Keywords: blockchain · fake news · cybersecurity

1 Introduction

Blockchain is a technology that enables the decentralised, secure, and immutable storage, transmission, and verification of information. As a result, it has found applications in various fields, including [1]:

- Cryptocurrencies and finance: Blockchain is widely used in cryptocurrencies such as Bitcoin, Ethereum, and Litecoin. It allows for direct storage and transfer of value between users without the involvement of financial intermediaries. Blockchain also enables the secure verification of financial transactions, identities, and audits.
- Supply chains: With blockchain, products can be traced from producer to consumer, allowing for better control over production processes, fraud prevention, and increased food safety.
- Medical data management: Blockchain enables the secure and unalterable storage and transfer of medical data, which can help protect patient privacy and enhance the effectiveness and speed of healthcare.

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 1–11, 2024.
https://doi.org/10.1007/978-981-99-8211-0_1

- Electronic voting: Blockchain provides secure and reliable electronic voting, which can prevent electoral fraud and ensure transparent and fair elections.
- Identification systems: Blockchain can securely and unalterably verify user identities, helping combat fraud and preventing cyber-attacks.
- Smart contracts: Blockchain enables the creation and execution of smart contracts that automatically fulfil the terms of the contract, without the need for intermediaries.
- Artificial intelligence: Blockchain can store and process large amounts of data required to train machine learning and artificial intelligence algorithms.

These are just a few examples of blockchain applications. This innovative technology has the potential to revolutionise many fields and industries. Blockchain allows for the decentralised, direct, and immutable storage and transmission of information, which makes it a promising tool in the fight against false information, particularly in verifying the credibility of content and information sources. Here is a literature review of the use of blockchain in combating fake news.

In their article [2], the authors systematically reviewed several scientific papers on using blockchain to verify fake news. They found that blockchain could detect and ensure correct information. The findings revealed that the suggested technique is satisfactory and efficient in recognising rumours and preventing their spread.

Due to availability of news and also for the free scope of sharing, most of the time rumours are being extensive in a short period of time. Detecting and preventing rumors and false information remains a significant challenge for social network. The introduction of blockchain technology has paved the way for the development of decentralised apps in order to address this issue. In this technology any information is recorded permanently. The authors in [2] explore a strategy to eliminate bogus news on social media by utilising the benefits of peer-to-peer network ideas. By issuing non-fungible token content rating we can detect and ensure appropriate news. The findings revealed that the suggested technique has a satisfactory performance and efficiency in recognising rumours and preventing their spread.

Prior research has proposed adding a quorum, a group of appraisers trusted by users to verify the authenticity of digital content, to the fake news prevention systems. In the paper [3], the authors propose an entropy-based incentive mechanism to diminish the negative effect of malicious behaviours on a quorum-based fake news prevention system. To maintain the Safety and Liveness of our system, their employed entropy to measure the degree of voting disagreement to determine appropriate rewards and penalties. They use Hyperledger Fabric, Schnorr signatures, and human appraisers to implement a practical prototype of a quorum-based fake news prevention system. The outcomes of the case analyses and experiments show that the presented mechanisms are feasible and provide an analytical basis for developing fake news prevention systems.

In conclusion, scientific literature suggests blockchain technology can help combat fake news by increasing data transparency and immutability. With further research and innovation in the field of blockchain, there is a promising prospect for its development in the fight against fake news.

2 The Structure of the Blockchain

Blockchain, i.e. a chain of blocks, is a technology used to store data securely, immutably and decentralised way. The blockchain structure consists of the following elements:

- Blocks - each block contains a set of transactions grouped and added to the chain at a particular time. Blocks are chained using hashes of cryptographic functions, preventing tampering with the blocks' data.
- Block header - The block header contains information about the block itself, such as the block number, its creation time, the hash of the previous block, and the proof of work used to validate the block's authenticity.
- Proof of Work - This is a mechanism used in the blockchain to protect the network against attacks and confirm the block's authenticity. It consists of solving mathematical problems using high computing power, making it difficult for attackers to falsify data.
- Peer-to-peer network - blockchain works based on a peer-to-peer network in which nodes are connected directly, creating a decentralised network.
- Network Nodes - Network nodes are the computers that create and maintain the blockchain by processing transactions, validating blocks, and sending them to other nodes in the network.
- Addresses - each blockchain account has its unique address used to carry out transactions.
- Transactions are saved in blocks and contain information about transferring values between addresses. Network nodes must verify and accept each transaction before being added to the block.
- Time Company - The blockchain uses a Time Company system to ensure that transactions are added to blocks in the correct order and time.

Thanks to such a structure, blockchain provides security, immutability and decentralised control over stored data.

3 The Problem of the Byzantine Generals

The blockchain network structure should ensure resistance to failures, disruptions in transmission and attempts to attack the content of individual blocks. A properly functioning decentralised system must implement mechanisms ensuring reliability in any situation.

In the case of analysing the situation with the appearance of incorrect messages in the blockchain network, the problem of Byzantine generals is often explored. Its assumptions are as follows: many Byzantine generals are planning an attack on the city. All generals must attack simultaneously or decide to retreat together to be victorious. They can only communicate with each other through messengers. The problem is that there are traitors among the generals who send the wrong messages. The solution to this theoretical problem must consider that the messages are open to all generals, including traitors. This thought experiment is described in detail in [4]. The problem of Byzantine generals' problem directly impacts distributed systems, for example, those based on blockchain.

The nodes in a distributed network, equivalent to the generals in the theoretical problem, can send and receive messages from each other. Those of them that work incorrectly is called Byzantine knots. Any node action that goes against the assumptions can be considered Byzantine action.

One of the conditions necessary for the proper operation of the blockchain network is creating a mechanism that ensures the creation of new blocks correctly. Such a mechanism is correct if it allows the product of blocks according to generally accepted consensus rules. At the same time, it shows high resistance to such problems as nodes with limited connectivity or nodes sending incorrect messages (intentionally or as a result of a failure). It is crucial that after the voting process, the servers make the same joint decision and the related action - adding a new block to the local chain or rejecting it. If even one server decides otherwise, it can lead to anomalies in the future. Supposing any server chooses to include a block that the rest of the network will not, then in future votes, that server will consider each subsequent block incorrect. When the node adds a new block, it checks whether the hash provided as the hash of the previous block in the candidate block is equal to the hash of the most recent block stored in the local copy. If the hash is incorrect, the server votes against that block.

Similarly, the system will behave when it has not previously attached a block that most servers have joined. Also, there will be a conflict when checking the consistency of the previous block's hash. Skipping a block is more likely than adding a block without majority approval. It can happen due to the temporary unavailability of the server. Let's assume, during this time, other servers vote to add a new block and accept it. The inactive server will never know about this vote unless it decides to compare its blockchain instance with one of the other servers later. However, the server does not have to be unavailable not to accept a new block. There may be a situation where a server fails to propagate its address when joining the network sufficiently. As a result, other servers fail to send their opinions about candidate blocks to it. So, the decision is made by a majority vote. It is assumed that consensus is reached if more than 50% of the servers consider the new block correct. In general, most nodes in the network will work fine.

Blockchain-based fake news detection.

Another serious problem is when servers send erroneous messages not because of an error but because of deliberate human interference. These types of situations are more dangerous because the attacker can act in a more coordinated manner and attack several servers simultaneously. One of the assumptions of a decentralised system is that each user or server has an equal impact on the network, and actions that interfere with the system's proper operation are not allowed and must be countered. Situations, where nodes send misleading messages, are closely related to the problem of Byzantine generals.

In the case of a problem with disseminating fake news, misleading messages will be untrue and should not be published. To analyse the problem, we chose a simple network consisting of four servers (A, B, C and D) during the voting process on the correctness of the new block, i.e. the truth of the information. All servers received the block and validated it. Servers B, C, and D determined the block was correct, while server A determined it was invalid and forwarded its vote information to the other servers. Even though server A cast the vote differently from the others in this case, this is not incorrect. Each server has the right to vote following the actual state of its blockchain.

In the analysed situation, the block will be accepted despite one vote against it because 75% of the servers say that the block is correct. In this case, server A will also decide to include a new block based on the votes from other servers. Since its vote was different, its blockchain is broken, and it needs to download the correct version from one of the other trustworthy servers. However, when server A (called the Byzantine or misleading server) knows that the block is correct (his correct vote would be a vote for block inclusion). It sends information to servers B and D that it thinks the block is correct, while it sends a misleading message to block C that the block is incorrect. In this way, it may disrupt the voting process in individual nodes; server B and D, after counting, consider that 100% of servers are in favor of including the block, while server C finds that only 75%.

One change increases the risk of making a wrong decision, in this case, on server C. For networks with less unanimity or more Byzantine servers, it can lead to an error in one or more nodes. In a situation where servers B and D, receive information that 75% of nodes found the block correct, they attach it to the local blockchain instance. Server C, on the other hand, as a result of the operation of the Byzantine server, receives information that the compatibility is 50%. In the case of the analysed example of fake news, a match greater than 50% is needed to reach a consensus, so server C does not accept the block. In this case, the Byzantine server achieves its goal - it manages to corrupt one of the servers, leading it to a situation where it has an outdated string.

The above analysis shows that the operation of Byzantine servers consists in sending misleading messages in such a way as to lead some servers to make an incorrect decision. Such servers may perform attacks randomly or direct them to a specific server. It is possible when the attacker controls multiple servers simultaneously and sends the wrong message only to the selected node.

The authors of this theoretical problem have already proposed the basic algorithm for solving the problem of Byzantine generals. The authors of the problem proposed the OM(m) algorithm, which solves the above problem for a group of not less than 3m + 1 generals, where m is the number of traitors [4].

The algorithm looks like this: for the value of OM(0), the commander sends his vote to every other general, and each general uses the received value to decide. If m is greater than zero, the OM(m) algorithm assumes that each general. After receiving a message, it will additionally send it to the other generals, omitting the original author of the message; it will therefore perform the OM(m-1) algorithm.

The creators also described the operation of this algorithm for the simplest case - four generals, one of whom is a traitor. For one traitor, this is the minimum number of generals, according to the following equation:

$$3 * \mathrm{m} + 1 = 3 * 1 + 1 = 4, \tag{1}$$

where m is the number of traitors.

If the server sending the voice is not a traitor, it sends its valid vote to all servers. Byzantine server B sends server D a false vote in the second voting round. However, server D also gets a vote for adding a block from server C. So it has three votes - two for (from the original server and server C) and one against adding a new block. So, based on the majority's decision, it accepts the new block. The algorithm must also work correctly when the original message's sender is a Byzantine server. The byzantine server sends three messages to other servers; it is not essential what the messages are because, in the second round of sending messages, the servers will exchange received messages. Therefore all nodes that are working correctly will have the same pool of messages (1, 2, 3) and will make the same decision.

The above algorithm has several disadvantages. As the analysis shows, for n nodes, the OM(m) algorithm first calls n-1 separate OM(m-1) algorithms, each of which calls the OM(m-2) algorithm n-2 times, which causes further recursive calls the algorithm until OM(0) is reached.

It follows that for networks with more than 1 Byzantine server, a large number of messages are required to be sent. Therefore, it has a negative impact on network performance, and a mechanism should be implemented that will allow for the differentiation of messages at successive levels of nesting.

3.1 A Simplified Solution with Signed Messages

The problem of Byzantine generals is much easier to solve when we can say with certainty that a given voice is coming from a specific server. As the analysed article shows, the problem under consideration becomes much easier to solve when the following assumptions are added: the signature of a loyal general cannot be forged, and every other general can verify the signature's correctness. As the authors of this algorithm claim and then prove, a solution can be used that works independently of the number of Byzantine generals [4].

The algorithm assumes that each general receives a signed order from the commander and then sends it to the other generals. As emphasised, generals, in contrast to the oral communication solution, can determine with certainty whether a commander is a traitor by comparing the received messages [4].

For the analysed problem of verifying the correctness of messages, a simplified algorithm with signed messages can be implemented based on the above idea. After receiving the voice from the server, each other server additionally marks the received vote and distributes it to other nodes. After receiving a certain number of votes or after the voting time has elapsed, each server checks the received votes. However, to determine what vote a particular server cast, it does not make decisions based on the majority - if it finds even one vote that contradicts the others, it means that the server is a Byzantine server. It is because each vote is cryptographically signed and unquestionably correct. The server thus has evidence that another server has attempted to interfere illegally with the network by sending conflicting messages to different nodes.

Attempted forgery when sending messages about the received vote is even easier to detect. First, the transmitted voice must be cryptographically correct, so you cannot effectively convince the other servers that the given server made a different decision than it was. Each server must provide proof of the original vote they previously received. In addition, there is no possibility of impersonating another server because also, at the second level, the voice must be cryptographically signed [5].

So the algorithm looks like this:

1. Server A sends its signed voice to the other servers.
2. For each server and different from A:
2.1. The server i saves the received vote.
2.2. Server i signs the received vote and distributes it to all other servers except A.
2.3. Server i starts the process of collecting information about what vote other servers received from server A. He gets a vote from every server except A:
2.3.1. The server i validates the voice. If it is incorrectly signed, it rejects it. If it is correct, it adds it to its list.
2.4. After the set time or receiving votes from all servers except A, the server verifies the received votes. If at least one of them is different from the others, it means that server A is a Byzantine server and is trying to make an unauthorised interference with the network. The server i notes this fact in a separate list and does not consider server A's vote.

The above algorithm applies to accepting a vote only from server A. It is performed as many times as there are servers in the network - each time, one of the servers sends its vote on a given block. Based on the stored list of Byzantine servers, each server can determine which votes to take into account and which to ignore during the final counting of votes.

3.2 Selection of the Consensus Algorithm

The correct operation of the application depends on the mechanisms based on which users decide whether a new block can be attached to the blockchain. Such tools are called consensus algorithms. There are many different types of mechanisms.

In public blockchains, it is necessary to introduce restrictions on who and on what basis can create new blocks. It avoids problems such as double spending or filling up the chain with unnecessary blocks with artificial transactions. An appropriate algorithm for reaching a consensus positively impacts the stability of the entire application, as it allows you to establish an actual rate of new data creation. For example, the Bitcoin chain implements a mechanism that determines the difficulty of creating a new block in a given period (difficulty matching period), considering the time stamp in the blocks of a given interval. Thanks to such calibration, regardless of the users' involvement, it always takes an average of 10 min to extract one block [6].

In the case of verifying the authenticity of the message, we suggest using an algorithm for reaching a consensus based on the reputation of a given user/server in the network. So it implements the Proof of Authority algorithm (PoA). In the case of this type of solution, PoA blockchain networks are secured by nodes that are uncompromisingly selected as reliable units [7, 8]. Anti-fake news portals, such as Snops.com, FactCheck.org, PolitiFact, and ABC, will play the role of these nodes. The second group of trusted servers will be recognised by large, credible publishers such as CNN, BBC, Times, Washington Post, etc. [9, 10].

3.3 Network Effectiveness Testing

A messaging application and a Byzantine server implementation were built as part of the research. Such a server sends a random message to each server during voting. It will be possible to examine the behaviour of both the server trying to influence the network and the incorrect action due to a failure. The probability for both vote values is 50%. The result of running the program for four servers, one of which is a Byzantine server, is as follows:

```
Node A | my choice: true | ratio: 75%
Node B | my choice: true | ratio: 75%
Node C | my choice: true | ratio: 100%
Byzantine Node D | my choice: false/false/true | ratio: 100%
Votes for: 4
Votes against: 0
Real ratio: 100 %
Final consensus: true
```

Even though the block was accepted, nodes A and B received a vote from the Byzantine server, resulting in a lower-than-real vote rate (ratio) for adding the block.

In another attempt, the Byzantine server was successful in attacking network integrity. The actual result of the vote is 75% - this is the result that should be in each of the servers, if there were no wrong messages sent. The Byzantine server sent node A a vote for adding a block and nodes B and C against it. Therefore, nodes B and C made a decision contrary to the actual state and rejected the block that, according to the objective state of the network, they should join. In addition, due to the fact that as many as 50% of the servers made a different decision, there was an unintended branching of the blockchain.

```
Node 0 | my choice: false | ratio: 75%
Node 1 | my choice: true | ratio: 50%
Node 2 | my choice: true | ratio: 50%
Byzantine Node 4 | my choice: true/false/false | ratio: 50%
Votes for: 3
Votes against: 1
Real ratio: 75 %
Final consensus: false
```

The test shows that less compatibility of properly functioning nodes increases the probability of error and the performance of Byzantine servers. The research conducted 1000 tests on a network of 100 servers. We conducted a second step or experiment to test the network's performance against the number of Byzantine servers.

The tests were repeated 100 times for each case with a different number of malfunctioning nodes. Since the total number of servers is 100, the number of Byzantine servers also equals their percentage. The obtained results are presented in Table 1 and graph 1. The results show that the network achieved 100% efficiency even when as much as 42% of the nodes were Byzantine servers. In further testing, there were isolated cases where the servers made a decision different from the general conclusion, but the efficiency remained around 99% until Byzantine servers made up 58% of the network; above this value, the efficiency of the network began to drop drastically. With 73% of the nodes defective, more than half of the tests were negative, and with 82%, almost all.

Therefore, a large number of nodes positively affect the correct operation of the network. It should be noted, however, that in the simulations, the Byzantine servers made decisions at random, which was more similar to the behaviour of servers sending incorrect messages as a result of a failure than as a result of intentional interference. One would expect that Byzantine servers could perform targeted attacks, i.e. send multiple messages to specific servers to achieve certain benefits. These types of actions are more difficult to simulate. On the other hand, more such groups of servers could try to exert an unfair influence on the network, and they would pursue different goals, so these tests seem reliable. Regardless of whether we consider Byzantine servers to be random or directed, our tests clearly show that such servers have a very large impact on the network and can easily lead to inconsistencies between individual nodes. The implementation of algorithms that counteract this type of attack is therefore necessary for the proper operation of the network.

Table 1. Correctness of network operation depending on the percentage of Byzantine servers

Percentage of Byzantine servers	Network correctness (%)
20	100
30	100
40	100
42	100
43	99,99
45	99,99
50	99,97
55	99,81
60	98,39
65	91,45
70	69,75
75	31,16
80	4,10
85	0,03
86	0,01
87	0
90	0

Therefore, a large number of nodes positively affect the correct operation of the network. It should be noted, however, that in the simulations, the Byzantine servers made decisions at random, which was more similar to the behaviour of servers sending

incorrect messages as a result of a failure than as a result of intentional interference. One would expect that Byzantine servers could perform targeted attacks, i.e. send multiple messages to specific servers to achieve certain benefits. These types of actions are more difficult to simulate. On the other hand, more such groups of servers could try to exert an unfair influence on the network, and they would pursue different goals, so these tests seem reliable. Regardless of whether we consider Byzantine servers to be random or directed, our tests clearly show that such servers have a very large impact on the network and can easily lead to inconsistencies between individual nodes. The implementation of algorithms that counteract this type of attack is therefore necessary for the proper operation of the network (Fig. 1).

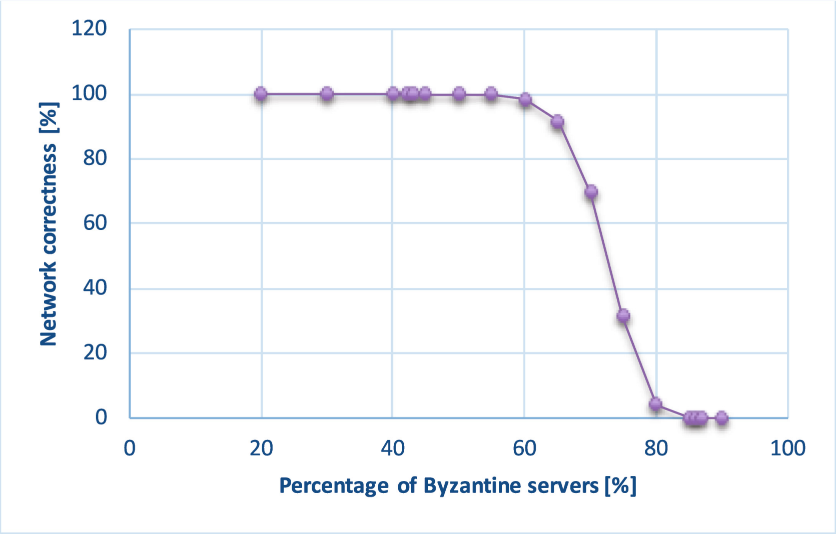


Fig.1. Correctness of network

4 Conclusion

Blockchain technology will find future applications in areas not yet associated with it. This technology can significantly reduce the real scourge of the Internet, fake news. Blockchain technology will allow you to track the way information is created and all attempts to modify it at every stage, thus preventing attempts to create false information. It has not been possible until now. Only blockchain technology allows you to do this by supporting the mechanism of interconnected transaction chains, which are 100% protected against outside interference. This article contains a proposal for the practical application of blockchain to protect against fake news. The study confirms the effectiveness of the proposed method. Its simple structure will undoubtedly be a vote for its use in practice.

References

1. Rutkowski, B.: Blockchain - technological aspects and examples of applications. https://www.lazarski.pl/pl/nauka-i-badania/instytuty/wydzial-ekonomii-i-zarzadzania/centrum-technologii-blockchain/blockchain-aspekty-technologiczne-oraz-przyklady-zastosowan/. Accessed 28 Feb 2023
2. Shahid, I.U., Anjum, M.T., Shohan, M.S., Tasnim, R., Al-Amin, M.: Authentic facts: a blockchain based solution for reducing fake news in social media. In: 4th International Conference on Blockchain Technology and Applications (2021)
3. Chen, C.C., Du, Y., Peter, R., Golab, W.M.: An implementation of fake news prevention by blockchain and entropy-based incentive mechanism. Soc. Netw. Anal. Min. **12** (2022)
4. Lamport, L., Shostak, R., Pease, M.: The Byzantine generals problem. ACM Trans. Program. Lang. Syst. **4**(3), 387–388 (1982)
5. Bashir, I.: Advanced Blockchain Applications, Helion SA (2019)
6. Song, J.: Understand Bitcoin. Cryptocurrency Programming from Scratch, Helion SA (2019)
7. Kozik, R., Choraś, M., Kula, S., Pawlicki, M.: Distributed architecture for fake news detection. In: Herrero, Á., Cambra, C., Urda, D., Sedano, J., Quintián, H., Corchado, E. (eds.) CISIS 2019. AISC, vol. 1267, pp. 208–217. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-57805-3_20
8. Zhou, X., Zafarani, R.: Fake news: a survey of research, detection methods, and opportunities. arXiv preprint arXiv:1812.00315, 2 (2018)
9. Bobulski, J.: Hidden Markov models for two-dimensional data. Adv. Intell. Syst. Comput.Intell. Syst. Comput. **226**, 141–149 (2013)
10. Kubanek M., Bobulski J.: The use of hidden Markov models to verify the identity based on facial asymmetry. Eurasip J. Image Video Process. **2017**(1), Article No. 45 (2017)

Formalization and Verification of the Zab Protocol Using CSP

Wenting Dong[1](✉), Jiaqi Yin[2], Sini Chen[1], and Huibiao Zhu[1]

[1] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University,
Shanghai, China
51255902036@stu.ecnu.edu.cn

[2] Northwestern Polytechnical University, Xi'an, China

Abstract. ZooKeeper Atomic Broadcast (Zab) is a high-performance atomic broadcast protocol, which is a key component of Apache ZooKeeper. By ensuring strong consistency and fault tolerance, the Zab protocol plays a crucial role in building robust and resilient distributed systems. However, the correctness and reliability of the Zab protocol have received limited attention in research. Thus, we employ Communicating Sequential Processes (CSP) to analyze and evaluate of the Zab protocol's properties and behavior. We utilize Process Analysis Toolkit (PAT) to verify six important properties, including Deadlock Freedom, Divergence Freedom, Data Reachability, Consistency, Sequentiality and Atomicity. The verification results demonstrate that the Zab protocol provides assurance of correctness and reliability.

Keywords: Zab protocol · CSP · Modeling · Verification · PAT

1 Introduction

With the rapid advancement of distributed systems and blockchain technology, a large number of novel protocols and algorithms have been proposed, such as Paxos [14], Raft [16], Zab [11], etc. However, numerous widely-utilized protocols and algorithms in distributed systems have yet to undergo further analysis and verification to ensure the security and correctness of the system.

To cover the gap, formal methods provide a rigorous and systematic approach to distributed system development, analysis, and verification [2, 8]. Formal methods are powerful technology and widely applied in numerous domains, such as operating systems [13], blockchain [17], cyber-physical systems [3], artificial intelligence [12] and so on.

Most recently, there are also some works using formal methods to verify protocols and algorithms. Wilcox et al. [19] presented a framework Verdi and verify Raft [16]. Chand et al. [4] used TLAPS to describe formal specification and verification of Multi-Paxos algorithm. Yin et al. [20] employed the TLA+ to specify and verify the limited properties of Zab. Inspired by these works, we choose process algebra CSP [7] to further model and analyze the Zab protocol.

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 12–24, 2024.
https://doi.org/10.1007/978-981-99-8211-0_2

In addition, the research on the correctness and reliability of Apache ZooKeeper and the Zab protocol has received limited attention. Apache ZooKeeper [9, 10] is a centralized and highly reliable distributed coordination service that is widely used in distributed systems. The fundamental protocol employed by ZooKeeper is the ZAB (Zookeeper Atomic Broadcast) protocol [11], which serves as a foundational building block for developing robust, scalable, and reliable distributed applications, and plays a crucial role in the frontier research and innovation in the field of distributed systems. Thus, it is crucial to conduct formal verification of the Zab protocol.

In this paper, we aim to provide a solid foundation for the analysis and verification of the Zab protocol by using process algebra CSP (Communicating Sequential Processes) [7]. We utilize model checker PAT (Process Analysis Toolkit) [18] to verify a broader range of properties, including Deadlock Freedom, Divergence Freedom, Data Reachability, Consistency, Sequentiality and Atomicity. The verification results show that the correctness and reliability of Zab protocol are guaranteed.

The remainder of this paper is organized as follows. Section 2 provides a brief introduction of the Zab protocol and the process algebra CSP. In Sect. 3, we illustrate the detailed modeling of the Zab protocol with three phases. In Sect. 4, we adopt PAT to implement the constructed model and verify six properties. Finally, Sect. 5 provides a summary of the contributions and potential future work.

2 Background

In this section, we give a concise explanation of the Zab protocol. At the same time, we give a brief introduction to the process algebra CSP.

2.1 Workflows of the Zab Protocol

Before introducing the Zab protocol, it is essential to clarify the entities involved in the protocol.

- **Client:** Clients interact with the distributed system and send update requests to the leader.
- **Leader:** Servers in the leading state are responsible for coordinating the replication of data updates across the followers. In case of failures or leader re-election, a new leader is elected among the followers.
- **Follower:** Servers in the following state replicate data updates received from the leader. They maintain a copy of the leader's log and execute the updates in the same order as the leader.
- **Looker:** Servers in the looking state actively participate in the leader election process by requesting votes from other servers, which occurs when there is no leader present in the system.

The Zab protocol operates in three main phases: Discovery phase, Synchronization phase and Broadcast phase. Next, we delineate them respectively.

Discovery Phase. In this phase, the servers in the ZooKeeper system discover each other and determine their roles. Initially, all servers start in the looking state, indicating that there is no leader. In addition, servers communicate and exchange information to elect a leader. They send election requests and respond with election requests from other servers to establish a new primary. Once a server receives votes from the majority, it switches to the leading state, indicating that it is recognized as a leader. The workflows of this phase are illustrated in Fig. 1.

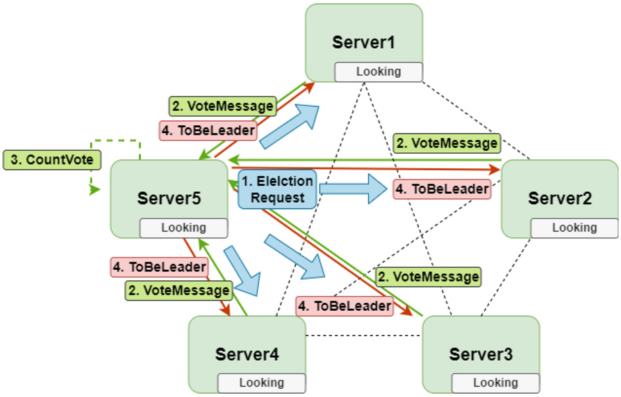


Fig. 1. The Workflows of the Discovery Phase

Synchronization Phase. After the leader is elected, the synchronization phase begins. In this phase, the followers synchronize data with the leader. It sends synchronization requests to the followers, which reply with their last known committed proposal. The leader then compares its own transaction log with the followers' logs and sends the missing proposals to bring them up to date. This ensures that all followers have an identical copy of the leader's log and brings them into a consistent state. The workflows of this phase are shown in Fig. 2.

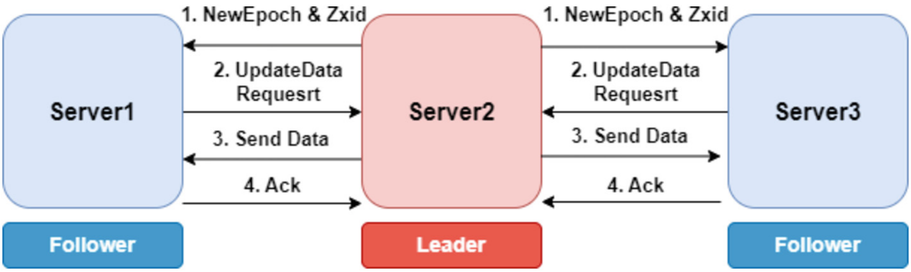


Fig. 2. The Workflows of the Synchronization Phase

Broadcast Phase. Once synchronization is complete, the servers enter into the broadcast phase. During this phase, the leader receives client's requests and proposes new proposals. It orders the proposals, assigns unique identifiers, and broadcasts them to all followers. The followers replicate the proposals. Then, when the leader receives the ACK message from more than half of the followers for the transaction proposal, it will send a commit message to all the followers. This ensures durability and guarantees the consistency of transactions across the entire cluster. Figure 3 shows the workflows of this phase.

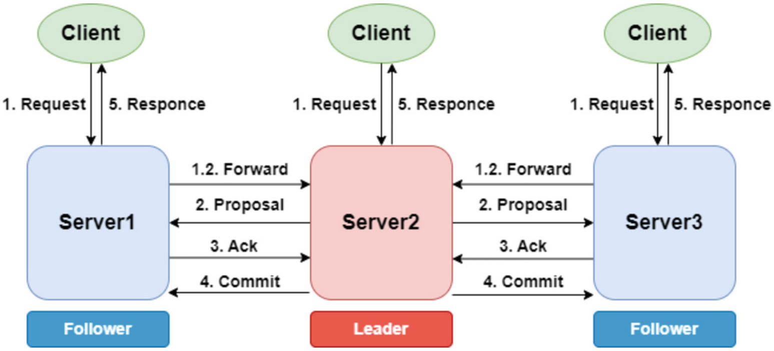


Fig. 3. The Workflows of the Broadcast Phase

2.2 CSP

CSP (Communicating Sequential Processes) [7] provides a rigorous mathematical theory and language. Additionally, CSP has been successfully applied to model and verify various concurrent systems and communication protocols [1, 5, 15]. Part of CSP syntax used in our model are described as follows.

$$P, Q = Skip|Stop|a \rightarrow P|c!x \rightarrow P|c?v \rightarrow P|P \,\Box\, Q$$
$$P||Q|P|||Q|P \lhd b \rhd Q|P;\, Q|P[|X|]Q$$

- *Skip*: The process does nothing, but terminates immediately.
- *Stop*: The process reaches deadlock.
- $a \rightarrow P$: After the execution of event a, process P is executed.
- $c!x \rightarrow P$: The process receives a message through channel c andassigns it to variable x, then behaves like P subsequently.
- $c?v \rightarrow P$: The process sends a value v through channel c, and then starts executing process P.
- $P \Box Q$: It stands for general choice between the processes P and Q.
- $P||Q$: Processes P and Q run in parallel.
- $P|||Q$: Processes P and Q run concurrently without barrier synchronization.
- $P \lhd b \rhd Q$: If the Boolean expression b equals true, then process P is executed, otherwise process Q is executed.
- $P;\, Q$: Processes P and Q execute in sequence.
- $P[|X|]Q$: The parallel composition of P and Q performs the concurrent events on the set X of channels.

3 Modeling

In this section, we present the formal modeling of the Zab protocol with CSP. Firstly, we give the whole structure of our model and then we model each phase of the Zab protocol respectively.

3.1 Sets, Messages and Channels

To facilitate the procedure of modeling and clarify the whole system, we give the definitions of sets, messages and channels used in this model.

Based on the mechanisms and processes involved in the Zab protocol, some sets have been defined. The set **Module** denotes all modules within the Zab protocol, which comprises clients, servers, and proposals. The set **ID** represents the unique identifier for each of the above module. The set **Status** is composed of all status that a server may be in. The set **Data** includes the data transmitted between modules. The set **Req** defines request messages and the set **Ack** contains feedback messages. Furthermore, we show a detailed definition of the relationship between relevant sets and constants in Table 1.

Table 1. The Relationship Between Involved Constants and Predefined Sets

Set	Constants
Module	C (client), S (server), P (proposal)
ID	CID (client id), SID (server id), ZXID (proposal id)
Status	Leading, Following, Looking, Crashing
Data	Data
Req	ReadData (request for data), ProposalMsg (request to store proposal), Election (request for election), LeaderMsg (request to be the leader), ReqProposal (request for proposal), CommitProposal (request to commit proposal)
Ack	True/1 (positive feedback), False/0 (negative feedback), VoteMsg (voting result)

Based on the above sets, we define messages transmitted among components. Depending on the type of messages, we abstract and classify the messages, and the definitions are as follows:

$$MSG_{req} = \{msg_{req}.A.B.content | A \in Module, B \in Module, content \in Req\}$$
$$MSG_{rep} = \{msg_{rep}.A.B.content | A \in Module, B \in Module, content \in Ack\}$$
$$MSG_{data} = \{msg_{data}.A.B.content | A \in Module, B \in Module, content \in \{Data, ID\}\}$$

where, MSG_{req} is composed of the request messages transmitted from module A to module B, MSG_{rep} denotes the response messages and MSG_{data} represents the data messages. A and B are the sender and the receiver respectively, and *content* represents content contained in each message.

Then, we define that *MSG* consists of the above three types of messages.

$$MSG = MSG_{req} \cup MSG_{rep} \cup MSG_{data}$$

Furthermore, we define the channels for communication among various modules. These channels are denoted as *COM_PATH*, shown as below:

- *ComCS*: The channels are between the client and server. In a ZooKeeper system, there are multiple client and server processes interacting with each other. So, the corresponding channels will also be generated, and we use subscript i to distinguish each channel expressed as $ComCS_i$.
- *ComSS*: The channels are between servers. Similarly, there are multiple channels. We use subscript j to distinguish each channel, denoted as $ComSS_j$.

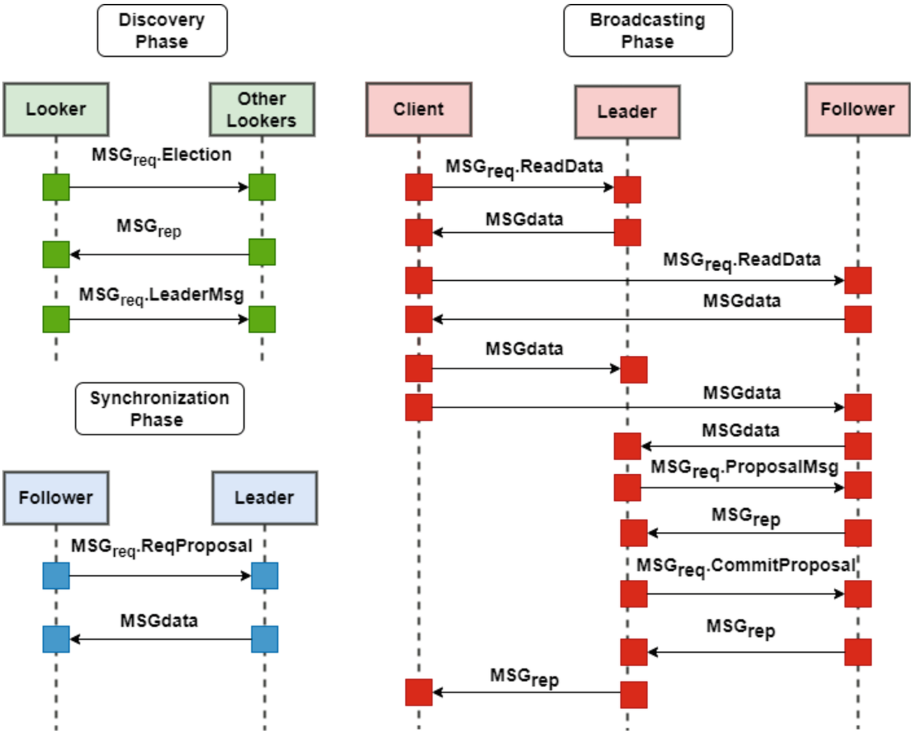


Fig. 4. The Communication Flow of the Zab Protocol

3.2 Overall Modeling

As described in Sect. 2, the behavior and status of servers differ depending on the phase of the Zab protocol. Therefore, the definition of *Server* is as follows:

$$Servers_{sid}() =_{df} \; Looker_{sid}() \lhd Status[sid] == looking \rhd \begin{pmatrix} Leader_{sid}() \lhd Status[sid] == leading \rhd \\ (Follower_{sid}() \lhd Status[sid] == following \rhd \; Faild_{sid}()) \end{pmatrix}$$

where, the array *Status*[*sid*] records the current status of server with IDsid. $Looker_{sid}()$, $Leader_{sid}()$, $Follower_{sid}()$ and $Faild_{sid}()$ are processes.

The ZooKeeper system can be abstracted as a system consisting of clients and servers. The communication flow of the Zab protocol is shown in Fig. 4. We formalize the whole model *System*() as below:

$$System =_{df} ||_{cid \in C; sid \in S; zxid \in P} (Client_{cid} [|COM\_PATH|] Servers_{sid})$$

3.3 Discovery Phase

During this phase, the servers do not process requests from clients. Instead, they communicate with each other using the *ComSS* channels to conduct a leader election. This phase contains three core processes, $Discovery_{sid,lsid}()$, $SendElection_{sid,lsid,T}()$ and $SendLeader_{sid,lsid}()$.

Firstly, the process $Discovery_{sid,lsid}()$ represents what the server needs to do during the discovery phase. If the $leaderCount == 0$, the server needs to initiate election requests, handle election requests from other servers, and process leader election requests. Otherwise, the server needs to synchronize data with the existing leader. The model is shown as follows:

$$\begin{aligned} & Discovery_{sid,lsid}() =_{df} \\ & \begin{pmatrix} Vote(sid, lsid) \rightarrow SendElection_{sid,lsid,T} \\ \Box \begin{pmatrix} \mathrm{ComSS_j}?\, msg_{req}.lsid.sid.Eletion\{Vote(lsid, sid)\} \rightarrow \\ ComSS_j!\, msg_{rep}.sid.lsid.VoteMsg \rightarrow Discovery_{sid,lsid}(\,) \end{pmatrix} \\ \Box\, \mathrm{ComSS_j}?\, msg_{req}.lsid.sid.LeaderMsg \rightarrow Synchronization_{lsid,sid}(\,) \\ \Box\, fail.sid\{Status[sid] = crashing\} \rightarrow Faild_{sid}(\,) \end{pmatrix} \\ & \lhd leaderCount == 0 \rhd \\ & \begin{pmatrix} Synchronization_{leaderID,sid}(\,) \\ fail.sid\{Status[sid] = crashing\} \rightarrow Faild_{sid}(\,) \end{pmatrix} \end{aligned}$$

In the above formula, $Vote(sid, lsid)$ is utilized to cast a vote from server sid to server $lsid$. $leaderCount$ Records the number of leaders in the current system.

Secondly, the process $SendElection_{sid,lsid,T}()$ means that server sid initiate an election request. It sets a time parameter T, so the process will wait for the reply message from the server $lsid$ until the time exceeds T. Within the time limit, if the server receives votes from more than half of the servers, it will be elected as the prospective leader. The model is depicted as follows:

$$SendElection_{sid,lsid,T}(\,) =_{df}$$
$$\left(\begin{array}{c} ComSS_j!\,msg_{req}.\,sid.\,lsid.\,Election \rightarrow SendELection_{sid,lsid,T-1}(\,) \\ \Box ComSS_j?\,msg_{req}.\,lsid.\,sid.\,LeaderMsg \rightarrow Synchronization_{lsid,sid}(\,) \\ \Box\left(\begin{array}{c} ComSS_j?\,msg_{rep}.\,lsid.\,sid.\,VoteMsg\{countVote(sid)\} \rightarrow \\ \left(\begin{array}{c} quasiLeader\{Status[sid] = leading; emptyVote(\,)\} \rightarrow \\ |||_{lsid1 \in S} SendLeader_{sid,lsid1}(\,) \\ \lhd (voteNumber * 2 > S \wedge leaderCount \ == \ 0 \rhd \ Discovery_{sid,lsid}(\,)) \end{array}\right) \end{array}\right) \end{array}\right)$$
$$\lhd T \neq 0 \rhd \ Discovery_{sid,lsid}(\,)$$

In the above formula, $countVote(sid)$ is used to calculate the number of votes received by server sid. $emptyVote()$ is a function to clear voting results.

Thirdly, the process $SendLeader_{sid,lsid}()$ is the last communication in the discovery phase. The prospective leader broadcasts that it intends to be a leader. The model is shown as below:

$$SendLeader_{sid,lsid}() =_{df} ComSS_j!msg_{req}.sid.lsid.LeaderMsg \rightarrow Skip$$

After the discovery phase, the leader is selected among the servers, while the remaining servers become followers. The $Looker_{sid}()$ process is designed to handle the status and process transitions of the servers.

$$Looker_{sid}(\,) =_{df} \ |||_{lsid \in S} \ Discovery_{sid,lsid}(\,);$$
$$\left(\begin{array}{c} \left(\begin{array}{l} toBeLeader\{leaderCount++; leaderID = sid; \\ CurrentEpoch = Epoch[sid]++\} \rightarrow Leader_{sid}(\,) \end{array}\right) \\ \lhd Status[sid] == leading \rhd \\ (Follower_{sid}(\,) \lhd Status[sid] == following \rhd \ Faild_{sid}(\,)) \end{array}\right)$$

Furthermore, we define the process $Faild_{sid}()$ to depict a server that can potentially fail at any time. The model is shown as below:

$$Faild_{sid}() =_{df} \left(\begin{array}{c} Err \rightarrow Faild_{sid}() \\ \Box revive.sid\{Status[sid] = looking\} \rightarrow Looker_{sid}() \end{array}\right)$$

3.4 Synchronization Phase

The synchronization phase primarily involves the followers' synchronizing data with the leader. This phase is composed of two parts: data update and data rollback. The array $CommitPro[sid]$ records the maximum $zxid$ (proposal id) committed by server sid. When $CommitPro[sid] > CommitPro[lsid]$, it indicates that the follower data is stale, so the follower needs to update the leader with the latest data. Otherwise, the follower needs to roll back data. We formalize the model as below:

$$Synchronization_{sid,lsid}(\,) =_{df}$$
$$\begin{pmatrix} ComSS_j!\,msg_{req}.\,lsid.\,sid.\,ReqProposal \rightarrow ComSS_j?\,msg_{data}.\,sid.\,lsid.\,Data \rightarrow \\ \{Proposal[lsid][nextZxid] = 1; CommitPro[lsid] = nextZxid\} \rightarrow \\ Synchronization_{sid,lsid}(\,) \end{pmatrix}$$
$$\lhd\ CommitPro[sid] > CommitPro[lsid]\ \rhd$$
$$\begin{pmatrix} \{rollback(lsid, CommitPro[sid]); CommitPro[lsid] = CommitPro[sid]\} \rightarrow \\ toBeFollower\{Status[lsid] = following;\ Epoch[lsid] = CurrentEpoch;\} \rightarrow \\ Skip \end{pmatrix}$$

Here, $rollback(lsid, zxid)$ is used to roll back data for follower. $Epoch[lsid]$ records which epoch the follower is in. $Proposal[lsid][zxid]$ means that the follower successfully stores the proposal $zxid$.

3.5 Broadcast Phase

After the discovery phase and the synchronization phase, the Zab protocol comes to the last broadcast phase. In this phase, the system is ready to handle external client requests. It contains three core processes $Client_{cid}()$, $Leader_{sid}()$ and $Follower_{sid}()$.

Client Modeling. The client participates in communicating with the followers and the leader to read or write data. We use general choice to split different scenarios of communication and formalize the model of the client as below:

$$\begin{aligned} Client_{cid}(\,) =_{df}\ & ComCS_i!\,msg_{req}.\,cid.\,sid.\,ReadData \rightarrow Client_{cid}(\,) \\ & \Box ComCS_i!\,msg_{data}.\,cid.\,sid.\,ZXID.\,Data \rightarrow Client_{cid}(\,) \\ & \Box ComCS_i?\,msg_{rep}.\,sid.\,cid.\,Data \rightarrow Client_{cid}(\,) \\ & \Box ComCS_i?\,p\_ack \rightarrow Client_{cid}(\,) \end{aligned}$$

Leader Modeling. Leader, responsible for handling data requests from clients and sending proposal to followers, is the important part of ZooKeeper system. $Leader_{sid}'()$ process comprises three main scenarios: handling read data request, broadcasting the proposal, broadcasting a request to commit proposal.

$$Leader'_{sid,fsid}(\,) =_{df}$$
$$\begin{pmatrix} ComCS_i?\,msg_{req}.\,cid.\,sid.\,ReadData \rightarrow \\ ComCS_i!\,msg_{rep}.\,sid.\,cid.\,Data \rightarrow Leader'_{sid,fsid}(\,) \end{pmatrix}$$
$$\Box \begin{pmatrix} ComCS_i?\ msg_{data}.\,cid.\,sid.\,zxid.\,Data\{Proposal[sid][zxid] = 1\} \rightarrow \\ ComSS_i!\,msg_{req}.\,fsid.\,zxid.\,SendProposalMsg.\,Data \rightarrow \\ ComSS_j?\,f\_ack\{countAck(zxid)\} \rightarrow \\ \begin{pmatrix} \begin{pmatrix} \{CommitPro[sid] = zxid\} \rightarrow \\ ComSS_j!\,msg_{req}.\,fsid.\,zxid.\,CommitProposalMsg \rightarrow \\ ComCS_i!\,p\_ack \rightarrow Leader'_{sid,fsid}(\,) \end{pmatrix} \\ \lhd\ ackNumber * 2 > F\ \rhd\ Leader'_{sid,fsid}(\,) \end{pmatrix} \end{pmatrix}$$
$$\Box \begin{pmatrix} fail.\,sid\{Status[sid] = looking;\ leaderCount = 0; leaderID = -1\} \rightarrow \\ Faid_{sid}(\,) \end{pmatrix}$$

Based on the information provided, we understand that the leader needs to broadcast messages to all followers. Therefore, the leader needs to establish a communication process with each follower. Thus, the model is shown as follows:

$$Leader_{sid}() =_{df} |||_{fsid \in S} Leader'_{sid,fsid}()$$

Follower Modeling. A follower maintains a replicated copy of the leader's data and stays in sync with the leader by receiving and applying the leader's proposals. Further, the follower should enter into process $Looker_{sid}()$, when there is no leader in servers. The detailed definition of followers is as below:

$$\begin{array}{l}
Follower_{sid}() =_{df} \\
\left(\begin{array}{c} ComSS_i?msg_{req}.fsid.zxid.SendProposalMsg.Data \rightarrow \\ \{Proposal[sid][zxid] = 1\} \rightarrow ComSS_j!f\_ack \rightarrow Follower_{sid}() \end{array}\right) \\
\square \left(\begin{array}{c} ComSS_j?msg_{req}.fsid.zxid.CommitProposalMsg \rightarrow \\ \left(\begin{array}{c} \{CommitPro[sid] = zxid\} \rightarrow Follower_{sid}() \\ \lhd nextZxid == zxid \rhd \; Follower_{sid}() \end{array}\right) \end{array}\right) \\
\square fail.sid\{Status[sid] = crashing\} \rightarrow Faild_{sid}() \\
\lhd leaderCount > 0 \rhd \; \{Status[sid] = looking\} \rightarrow Looker_{sid}()
\end{array}$$

4 Verification

In this section, we conduct verification through the model checker tool PAT to verify the properties of the constructed formal model, including *Deadlock Freedom, Divergence Freedom, Data Reachability, Consistency, Sequentiality, Atomicity*, and then analyze the results.

4.1 Properties

Property 1: Deadlock Freedom. Deadlock means the system is blocked and no further operation can be done. PAT provides us with a primitive:

$$\#assertSystem()deadlockfree;$$

Property 2: Divergence Freedom. Divergence refers to the state in which the system enters an infinite loop. PAT also provides a primitive assertion:

$$\#assertSystem()divergencefree;$$

Property 3: Data Reachability. The Zab protocol should ensure that clients can successfully obtain the requested information. Then we have:

$$\begin{array}{c} \#define\ DataReachability(data\_reachability == true); \\ \#assert\ system\ (reaches)\ DataReachability \end{array}$$

Property 4: Consistency. According to the CAP (Consistency, Availability, Partition tolerance) theorem [6], consistency refers to ensuring that all nodes in a distributed system will see the same data at the same time. We define the array *Proposal*[*sid*][*zxid*] to represent proposals stored locally by each server. Thus, *Proposal*[*sid*][*zxid*] of each server needs to be consistent, then we have:

```
#define Consistency(Proposal[0][0] == Proposal[1][0] &&
                    Proposal[2][0] == Proposal[1][0] &&
                    Proposal[0][1] == Proposal[1][1]&&
                    Proposal[2][1] == Proposal[1][1]);
#assert System() |= Consistency;
```

Property 5: Sequentiality. The Zab protocol should ensure that all write operations are broadcast and executed in the order as they are received within the system. It means that under no circumstances where *Proposal*[*sid*][1] is received before *Proposal*[*sid*][0]. The assertions are as follows:

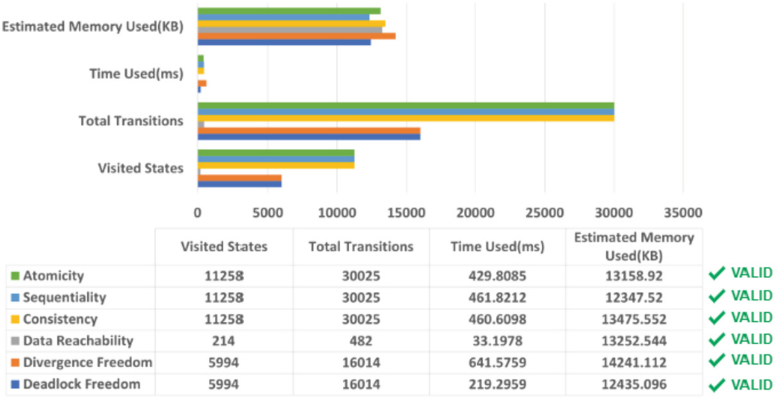
```
#define Sequentiality(! ((Proposal[0][0] == 0 && Proposal[0][1] == 1) ||
                        (Proposal[1][0] == 0 && Proposal[1][1] == 1) ||
                        (Proposal[2][0] == 0 && Proposal[2][1] == 1)));
#assert System() |= Sequentiality;
```

Property 6: Atomicity. Atomicity means each write operation is either fully broadcast and executed across all nodes or not executed at all. Thus, we adopt *Pro*[*zxid*] to record whether proposal with number *zxid* is sent. To guarantee the atomicity of message broadcasts, *Proposal*[*sid*][*zxid*] should have the same value as *Pro*[*zxid*]. The asserts are defined as follows:

```
#define Atomicity(Proposal[0][0] == Pro[0] && Proposal[0][1] == Pro[1]&&
                  Proposal[1][0] == Pro[0] && Proposal[1][1] == Pro[1]&&
                  Proposal[2][0] == Pro[0] && Proposal[2][1] == Pro[1]);
#assert System() |= Atomicity;
```

4.2 Verification and Results

Based on the above definitions and assertions, we implement the formal model in PAT. Our experiments are conducted on a computer with Win10 OS, 12th Gen Intel(R) Core (TM) i7-12700H 2.30 GHz CPU and 16 GB RAM. At the end, we get the results of verification shown in Fig. 5. It shows the verification statistics provided by PAT, including visited states, total transitions, time used and estimated memory used. Further, we can see that the six properties are all valid, which means the Zab protocol can enable reliable and consistent communication in the distributed systems, making it suitable for applications that require strong consistency and fault tolerance.



	Visited States	Total Transitions	Time Used(ms)	Estimated Memory Used(KB)	
🟩 Atomicity	11258	30025	429.8085	13158.92	✓ VALID
🟦 Sequentiality	11258	30025	461.8212	12347.52	✓ VALID
🟨 Consistency	11258	30025	460.6098	13475.552	✓ VALID
⬜ Data Reachability	214	482	33.1978	13252.544	✓ VALID
🟧 Divergence Freedom	5994	16014	641.5759	14241.112	✓ VALID
🟦 Deadlock Freedom	5994	16014	219.2959	12435.096	✓ VALID

Fig. 5. Verification Results

5 Conclusion and Future Work

In this paper, we have applied the process algebra CSP to describe and model the Zab protocol. Further, we abstracted six fundamental properties from Zab the protocol and utilized the verification tool PAT to verify the constructed model. Ultimately, the verification results demonstrate the ability of the Zab protocol to perform reliable and consistent communication in the distributed systems.

In the future, we will continue to refine our work on the formalization and validation of ZooKeeper by adding more detailed workflows and exploring the security aspects of the system.

Acknowledgments. This work was partially supported by the “Digital Silk Road” Shanghai International Joint Lab of Trustworthy Intelligent Software (No. 22510750100), Shanghai Trusted Industry Internet Software Collaborative Innovation Center, the Open Project of Shanghai Key Laboratory of Trustworthy Computing, and the Fundamental Research Funds for the Central Universities.

References

1. Antonino, P., Gibson-Robinson, T., Roscoe, A.: Efficient verification of concurrent systems using synchronisation analysis and SAT/SMT solving. ACM Trans. Softw. Eng. Methodol. **28**(3), 1–43 (2019)
2. Bowen, J.P., Hinchey, M.G.: Formal methods. In: Tucker, A.B. (ed.) Computer Science Handbook, 2nd edn, Section XI, Software Engineering, chap. 106, pp. 106–1–106–25. CRC Press, Boca Raton (2004)
3. Bu, L., et al.: Toward online hybrid systems model checking of cyber-physical systems’ time-bounded short-run behavior. ACM SIGBED Rev. **8**(2), 7–10 (2011)
4. Chand, S., Liu, Y.A., Stoller, S.D.: Formal verification of multi-paxos for distributed consensus. In: Fitzgerald, J., Heitmeyer, C., Gnesi, S., Philippou, A. (eds.) FM 2016. LNCS, vol. 9995, pp. 119–136. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48989-6_8

5. Chen, N., Zhu, H., Yin, J., Fei, Y., Xiao, L., Zhu, M.: Modeling and verifying NDN-based IoV using CSP. J. Softw. Evol. Process **34**(10), e2371 (2022)
6. Gilbert, S., Lynch, N.A.: Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. SIGACT News **33**, 51–59 (2002)
7. Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall, Englewood Cliffs (1985)
8. Holloway, C.: Why engineers should consider formal methods. In: 16th DASC. AIAA/IEEE Digital Avionics Systems Conference. Reflections to the Future. Proceedings, vol. 1, pp. 16–22 (1997)
9. Hunt, P., Konar, M., Junqueira, F.P., Reed, B.: Zookeeper: wait-free coordination for internet-scale systems. In: USENIX Annual Technical Conference, vol. 8 (2010)
10. Junqueira, F., Reed, B.: ZooKeeper: Distributed Process Coordination. O'Reilly Media, Inc., Sebastopol (2013)
11. Junqueira, F.P., Reed, B.C., Serafini, M.: Zab: high-performance broadcast for primary-backup systems. In: IEEE/IFIP 41st International Conference on Dependable Systems and Networks, pp. 245–256 (2011)
12. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: an efficient SMT solver for verifying deep neural networks. In: Majumdar, R., Kunčak, V. (eds.) CAV 2017. LNCS, vol. 10426, pp. 97–117. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63387-9_5
13. Klein, G., et al.: sel4: Formal verification of an OS kernel. In: Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles, pp. 207–220 (2009)
14. Lamport, L., Malkhi, D., Zhou, L.: Vertical paxos and primary-backup replication. In: Proceedings of the 28th ACM Symposium on Principles of Distributed Computing, pp. 312–313 (2009)
15. Lowe, G., Roscoe, B.: Using CSP to detect errors in the TMN protocol. IEEE Trans. Softw. Eng.Softw. Eng. **23**(10), 659–669 (1997)
16. Ongaro, D., Ousterhout, J.K.: In search of an understandable consensus algorithm. In: USENIX Annual Technical Conference, Philadelphia, PA, USA, pp. 305–319. USENIX Association (2014)
17. Rosu, G.: Formal design, implementation and verification of blockchain languages (invited talk). In: 3rd International Conference on Formal Structures for Computation and Deduction. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2018)
18. Sun, J., Liu, Y., Dong, J.S.: Model checking CSP revisited: introducing a process analysis toolkit. In: Margaria, T., Steffen, B. (eds.) ISoLA. CCIS, vol. 17, pp. 307–322. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88479-8_22
19. Wilcox, J.R., et al.: Verdi: a framework for implementing and formally verifying distributed systems. In: Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation, pp. 357–368 (2015)
20. Yin, J., Zhu, H., Fei, Y.: Specification and verification of the Zab protocol with TLA+. J. Comput. Sci. Technol.Comput. Sci. Technol. **35**(6), 1312–1323 (2020)

Using MPIs Non-Blocking Allreduce for Health Checks in Dynamic Simulations

Jana Gericke(✉), Harald Klimach, Neda Ebrahimi Pour, and Sabine Roller

German Aerospace Center (DLR), Institute of Software Methods for Product Virtualization,
Zwickauer Str. 46, Dresden, Germany
{jana.gericke,harald.klimach,neda.ebrahimipour,
sabine.roller}@dlr.de
https://www.dlr.de/sp/en/

Abstract. Large-scale simulations often require frequent checks on global conditions that are not directly needed for the computation itself. While such health checks are not an integral part of the numerical algorithm, they serve an important role in controlling and coordinating the simulation. In distributed parallel computations their required communication may negatively impact the actual parallel computation and result in unnecessary synchronization points. We show that by using a non-blocking reduction these synchronization requirements can be loosened and the impact on the actual computation minimized. Further, it enables us to shift the communication into the background and progress it during the *MPI* calls that are done during the computation anyway. We demonstrate that a sufficient amount of *MPI* calls in between is required to allow for progress to happen. The presented approach delays the decision to be made in response to those health checks. But as it is not vital for the correct computation itself such a delay is usually tolerable and could offer a more robust scaling to large process counts.

Keywords: MPI · Non-Blocking Collectives · Dynamic Simulations

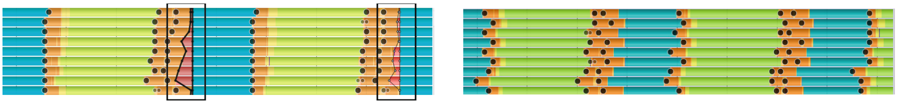
1 Introduction and Methods

In transient simulations that may last several days of computation, there often arises the need to check for some conditions globally. For example, to determine the end of the computation based on different criteria like negative density on some process or the detection of a steady-state. Checking for such global conditions requires communication in distributed parallel computations and hence, may impact the scalability of the solver, though that communication is not strictly necessary for the calculations in the simulation itself. This kind of checks are traditionally achieved by blocking collective reductions across all processes, leading to strict synchronization points. Frequent synchronizations result in a larger dependency on run time variations across processes. Therefore, we describe the utilization of a non-blocking reduction operation (`MPI_Iallreduce`), that was introduced [1] in *MPI 3* [2], for health checks in our open-source simulation framework *APES* [3]. *MPI*s non-blocking collectives have been successfully utilized to improve the scalability of computations [4–9] by overlapping communication and computation and thus, hiding the cost of the collective.

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 25–31, 2024.
https://doi.org/10.1007/978-981-99-8211-0_3

For codes, that were successful in avoiding collectives for the numerical scheme, the need for health checks brings collectives back in. To overlap these health check collectives with (the fully non-collective/fully point-to-point) computation introduces the need for an improved concept. As the health checks are often necessary, but not urgent, they are not required instantaneously (what would be the advantage of blocking communication). Thus, non-blocking communication is the method of choice. To demonstrate the approach, we employ our Lattice Boltzmann solver *Musubi* [10]. It uses very few operations per element but requires the communication of relatively large data from its neighbors, resulting in a large communication-to-computation ratio. Further, it makes use of the *TreElM* [11] library where this health check mechanism is implemented.

For this mechanism, we introduce the option to delay the evaluation of health checks by using the non-blocking *Allreduce* of *MPI*. Up to now, the communication of the status information was achieved with the blocking `MPI_Allreduce` operation. Setting several status flags as logicals on each process, allows us to use a *logical or* as reduction operation in the *Allreduce* operation and provides the information if any process has encountered a specific status. This blocking communication, however, introduces a synchronization point as shown in Fig. 1a.


(a) Trace of blocking routine.

(b) Trace of non-blocking routine.

Fig. 1. Comparison of traces for blocking and non-blocking (delayed) health check routine. Computations are green and blue, communication brown, the *Allreduce* red.

With the non-blocking `MPI_Iallreduce` introduced in *MPI 3* it is now possible to overcome that synchronization by performing the communication in the background (see Fig. 1b) and delay the evaluation of the reduction operation to some later point in time. Such a delayed evaluation allows for a larger variation in computing times across processes without causing idle times for the synchronization. Delaying the check can be achieved within a single subroutine that combines both, the initiation of the collective reduction operation and the waiting on the resulting global status. This is represented in Listing 1.1.

Listing 1.1: Non-blocking (delayed) health check routine in Fortran.

```fortran
type status_type
  integer :: check_request = MPI_REQUEST_NULL
  logical :: bits(stat_nFlags) = .false.
  logical :: oldbits(stat_nFlags) = .false.
end type

subroutine status_communicate_delayed(me, comm)
  type(status_type), intent(inout) :: me
  integer, intent (in) :: comm
  integer :: iError, sync_status(MPI_STATUS_SIZE)
  logical :: local_bits(stat_nFlags)

  local_bits = me%bits

  if (me%check_request /= MPI_REQUEST_NULL) then
    ! Wait on previous Iallreduce to update status.
    call MPI_Wait(me%check_request, sync_status, &
      &           iError                         )
    me%bits = me%oldbits
  end if

  me%oldbits = local_bits

  ! Initiate reduction for current iteration.
  call MPI_Iallreduce(MPI_IN_PLACE, me%oldbits,  &
  &     stat_nFlags, MPI_LOGICAL, MPI_LOR, comm, &
  &     me%check_request, iError                 )

end subroutine status_communicate_delayed
```

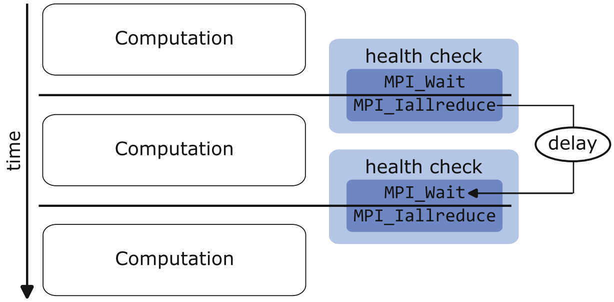


Fig. 2. Timeline for the non-blocking (delayed) health check routine.

The implementation in *TreElM* can be found in the public code repository [11]. This routine acts as a drop-in replacement for the check with a blocking MPI_Allreduce. The application modifies and reads the status information provided by the logicals in me%bits. Communication of the local status (set by the calling program between calls) uses a copy of bits, stored in me%oldbits. The MPI_LOR reduction operation determines if a flag has been set by any process and the result is made known to all processes in place. The request handle for this non-blocking collective operation is stored in me%check_request. Upon the next call of the routine, the synchronized global status is copied back into the `me%bits` field after waiting on the communication to complete. The timeline for the usage of this non-blocking (delayed) health check routine is shown in Fig. 2. Between the calls to the status communication one or more time iterations of the simulation are performed in our applications, involving their own *MPI* communications that allow the reduction operation to progress.

2 Results

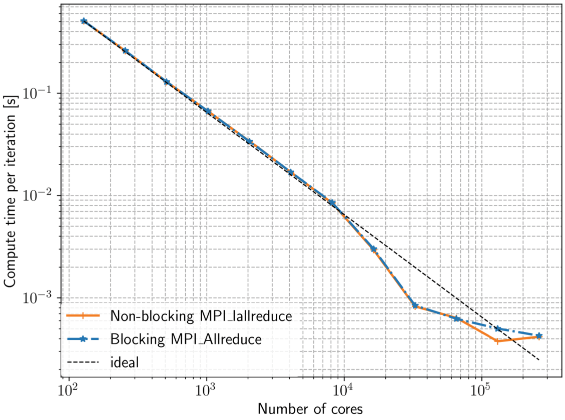


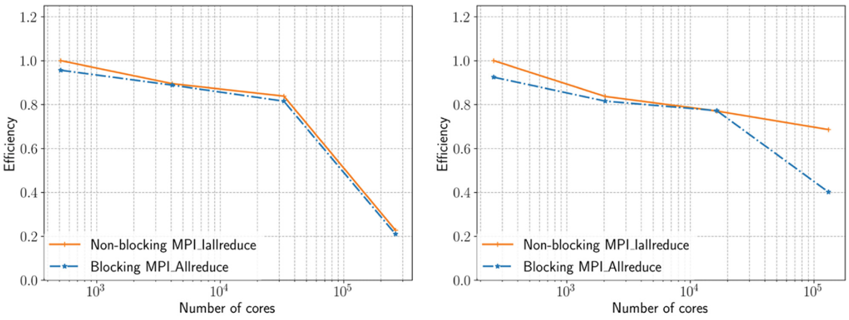
Fig. 3. Compute time per iteration of *Musubi* on *Hawk* using 1 to 2,048 nodes. Strong scaling measurement for 134,217,728 elements with non-blocking (orange) and blocking (blue) reduction. A dashed black line indicates ideal scaling.

To investigate the effect of this approach for health checks in the *TreElM* library, we run *Musubi* on the *Hawk* supercomputing system installed at the High-Performance Computing Center Stuttgart (*HLRS*). This supercomputer is based on the *HPE Apollo* system and utilizes an *InfiniBand HDR200* network. Each node has two *AMD EPYC 7742* processors with 64 cores each, operating at 2.25 GHz [12].

In our runs we use up to 2,048 compute nodes resulting in 262,144 cores. The *GCC* compiler 9 together with the *HPE MPI* library is used to compile the solver. In all runs one *MPI* process per physical core is used for the computation. A simple, homogeneous setup with balanced partitions is applied in the investigation. Each run is performed for around 6 min of run time. Figure 3 shows the compute time per iteration including the health check exchange in *Musubi* on *Hawk*.

A near ideal scaling is achieved for up to 10,000 processes. In this region, the choice of health check routine has no effect. Once the problem per process gets sufficiently small (8,192 elements per process), some caching effects can be observed and a super-linear speed-up is achieved. When the actual computing operations per process become small (1,024 elements per process), a benefit of roughly 40% by the non-blocking health check can be observed. This is also illustrated in more detail in the efficiency plots of a weak scaling at the peak in the speed-up in Fig. 4a and for the largest deviation in Fig. 4b.

For 4,096 elements per process, the benefit is in the range of single-digit percentage, while it is much higher for even smaller partitions. For 1,024 elements per process, it enables even more speed-up towards the extreme end.

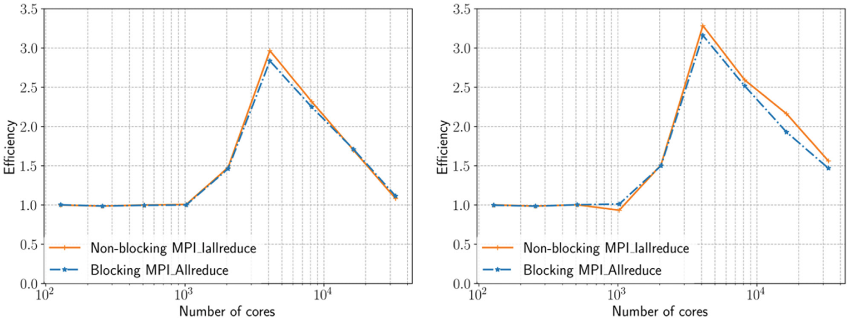


(a) 4,096 elements per process.

(b) 1,024 elements per process.

Fig. 4. Efficiency of *Musubi* on *Hawk* using the smallest possible number of nodes up to 2,048. Weak scaling measurement for non-blocking (orange) and blocking health check (blue) for the peak performance of the strong scaling (left) as well as for the largest deviation (right) in Fig. 3.

The effect of the interval length between health checks is shown in Fig. 5. With a slightly larger interval between checks the benefit increases, especially for smaller partitions where an improvement of the overall computational efficiency of up to 10% can be observed. A larger check interval yields more intermediate *MPI* calls and thus, more time for the collective to complete. This requirement of a sufficient amount of *MPI* calls in between to allow for progress to happen, is also observed in a small dedicated kernel that emulates the setup. Further increases of the interval length do not improve the run times.



(a) Check interval = 1.

(b) Check interval = 5.

Fig. 5. Efficiency of *Musubi* on *Hawk* using 1 to 256 nodes for different check intervals: 1 (left) and 5 (right). Strong scaling measurement for 16,777,216 elements with non-blocking (orange) and blocking health check (blue).

A first glimpse at a second test case with about 500 million elements and load imbalances – boundary conditions, grid refinement etc. – reveals that there is a huge benefit for this kind of application. Besides, the time spend for each non-blocking health check approaches zero.

3 Conclusion and Future Work

We investigated a possible application for non-blocking collective operations: health checks in distributed parallel simulations. Delaying the check loosens the synchronization requirements and hides the cost of the *Allreduce*, though for the considered application the impact of this simple change remains small. We observe that a sufficient amount of intermediate *MPI* calls is needed to progress. The impact in more complex simulation setups requires further investigations and consideration of other solvers.

Acknowledgement. The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer HAWK at Höchstleistungsrechenzentrum Stuttgart (www.hlrs.de).

References

1. Hoefler, T., Kambadur, P., Graham, R.L., Shipman, G., Lumsdaine, A.: A case for standard non-blocking collective operations. In: Cappello, F., Herault, T., Dongarra, J. (eds.) EuroPVM. LNCS, vol. 4757, pp. 125–134. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75416-9_22
2. Message Passing Interface Forum: MPI: A Message-Passing Interface Standard Version 3.0. https://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf. Accessed 31 May 2023
3. Roller, S., Bernsdorf, J., Klimach, H. et al.: An adaptable simulation framework based on a linearized octree. In: Resch, M., Wang, X., Bez, W., Focht, E., Kobayashi, H., Roller, S. (eds.) High Performance Computing on Vector Systems 2011, pp. 93–105. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22244-3_7
4. Hoefler, T., Gottschling, P., Lumsdaine, A., et al.: Optimizing a conjugate gradient solver with non-blocking collective. Parallel Comput. **33**(9), 624–633 (2007). https://doi.org/10.1016/j.parco.2007.06.006
5. Kandalla, K., Yang, U., Keasler, J., et al.: Designing non-blocking allreduce with collective offload on infiniband clusters: a case study with conjugate gradient solvers. In: IEEE 26th International Parallel and Distributed Processing Symposium, pp. 1156–1167 (2012)
6. Widener, P., Ferreira, K., Levy, S., et al.: Exploring the effect of noise on the performance benefit of nonblocking Allreduce. In: Proceedings of the 21st European MPI Users' Group Meeting, pp. 77–82. New York, USA (2016). https://doi.org/10.1145/2642769.2642786
7. Eller, P.R., Gropp, W.: Scalable non-blocking preconditioned conjugate gradient methods. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 204–215. (2016). https://doi.org/10.1109/SC.2016.17
8. Eller, P.R., Hoefler, T., Gropp, W.: Using performance models to understand scalable Krylov solver performance at scale for structured grid problems. In: Proceedings of the ACM International Conference on Supercomputing, pp. 138–149. New York, USA (2019). https://doi.org/10.1145/3330345.3330358

9. Kwon, O.K., Lee, J., Lee, J., et al.: MPI parallel implementation for pseudo-spectral simulations for turbulent channel flow. Int. J. Comput. Fluid Dyn. **34**(7–8), 569–582 (2020). https://doi.org/10.1080/10618562.2020.1828579
10. Hasert, M., Masilamani, K., Zimny, S., et al.: Complex fluid simulations with the parallel tree-based Lattice Boltzmann solver Musubi. J. Comput. Sci.Comput. Sci. **5**(5), 784–794 (2014)
11. TreElM repository. https://osdn.net/projects/apes/scm/hg/treelm. Accessed 31 May 2023
12. HLRS: HPE Apollo (Hawk). https://www.hlrs.de/solutions/systems/hpe-apollo-hawk. Accessed 31 May 2023

Parallelizable Loop Detection using Pre-trained Transformer Models for Code Understanding

Soratouch Pornmaneerattanatri[1](✉), Keichi Takahashi[2], Yutaro Kashiwa[1], Kohei Ichikawa[1], and Hajimu Iida[1]

[1] Nara Institute of Science and Technology, Nara 630-0192, Japan
pornmaneerattanatri.so.pn8@is.naist.jp
[2] Tohoku University, Sendai 980-8578, Japan

Abstract. Parallel programming is essential to utilize multi-core processors but remains challenging because it requires extensive knowledge of both software and hardware. Various automatic parallelization tools based on static analysis have been developed to ease the development of parallel programs. However, hand-parallelized codes still outperform auto-parallelized codes. Meanwhile, transformer-based large language models have made ground-breaking progress in coder understanding and generation tasks. In this paper, we fine-tune a transformer-based code understanding model, CodeT5, to create a model for automatically identifying parallelizable for-loops. The trained model helps developers to identify independent for-loops that can be potentially parallelized using tools such as OpenMP to improve the program performance. Our model is trained over 90,908 for-loops collected from 9 million C/C++ source files of public GitHub repositories, and achieves a 0.895 F1 score in identifying parallelizable for-loops in public GitHub projects and a 0.713 F1 score in the NAS Parallel Benchmark suite.

Keywords: Parallel computing · OpenMP · Automatic parallelization · Deep learning · Large Language Model

1 Introduction

A wide variety of programming languages, libraries [16], and frameworks [17] have been proposed to support the development of parallel applications. OpenMP [6] is one of the prominent language extensions for C/C++ and Fortran for developing parallel applications. OpenMP provides a set of simple compiler directives that allow developers to explicitly annotate sections of code to be parallelized on a shared-memory computer.

To overcome these challenges, automatic parallelization tools [2, 7, 10, 13] have been developed. These tools analyze the source codes and automatically generate parallelized

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 32–42, 2024.
https://doi.org/10.1007/978-981-99-8211-0_4

versions of the source codes. However, while these tools are steadily improving, it is often reported that they produce lower-performance executables than human-written parallelized codes [2].

This is because current automatic parallelization tools rely on static analysis that identifies if certain code sections (*e.g.*, loop iterations) can be safely executed in parallel. Since static analyzers only have access to information available at compile time, automatic parallelization has fundamental limitations compared to hand parallelization. First, automatic parallelization is unable to parallelize a loop if it requires information given at runtime or application-specific knowledge. Second, even if a loop can be executed in parallel, parallelizing it might not be profitable due to the overhead of parallelization. Furthermore, the runtime for static analysis grows rapidly as the size and complexity of the code increase.

On the other hand, recent advances in deep learning-based natural language processing techniques (NLP) [18] have made it possible to extract knowledge from source codes by trailing NLP models using the massive amounts of source codes available on public code hosting systems such as GitHub. Many studies [9, 19] have successfully demonstrated the applicability and usefulness of NLP techniques for various tasks involving computer languages. Therefore, it might be feasible to learn various parallelization patterns from publicly available source codes and develop tools to support parallel programming using deep learning-based NLP techniques.

In this research, we propose a model that classifies whether a for-loop is parallelizable or not by a fine-tuned CodeT5 model. CodeT5 is a deep learning-based NLP model that is pre-trained on various programming languages including C, Java, and Python. This classification model helps developers to easily transform serial programs into parallel programs.

2 Related Work

2.1 Automatic Parallelization

Recent compilers are capable of automatic parallelization using static analysis. GNU C Compiler (GCC) and Intel C Compiler (ICC) can generate parallel versions of the code. In the case of ICC, dataflow analysis is used [1]. Although codes parallelized by these compilers can achieve speedups, parallelization of complex loops still requires manual intervention and guidance.

Some source-to-source compilers also feature automatic parallelization based on various static analysis techniques. Cetus [3, 7], Rose [10, 14], DawnCC [13] and Clava [2, 5] mainly use dependence and range analysis. Dependence analysis determines whether a for-loop is parallelizable or not using privatization, reduction, or alias analysis. Also, Range analysis determines parallelizable for-loops by calculating the lower and upper bounds of integer variables interacting at each point in the application.

These source-to-source compilers differ in the following aspects. Cetus only supports C language and uses privatization, reduction, and alias analyses for dependence analysis. Rose's dependence analysis determines parallelizable loops by solving a set of linear integer equations of loop induction variables to access arrays in the loops using

Gaussian elimination algorithm. DawnCC can transform C/C++ source code into parallelized code that uses OpenMP, OpenCL or CUDA. It mainly uses symbolic range analysis supported by classic dependence analysis to determine parallelizable loops. Clava uses induction variables, privatization, and scalar and array reductions for dependence analysis. Scalar and array reductions are applied only when the first two techniques fail to determine parallelizable loops. Nevertheless, the source codes generated by these source-to-source compilers still cannot surpass hand-written source codes in terms of execution performance [2].

2.2 Natural Language Processing Deep Learning Model

The advancement of deep learning has opened new possibilities in various fields that use human-generated data on the Internet. Text is the most abundant data source, but it is not instantly available. First, the context of a sentence must be labeled. Moreover, to achieve higher performance from the language models, a large amount of labeled sentences is needed.

A breakthrough in the NLP field occurred when the Transformer model [18] was proposed and achieved a new record in machine translation tasks. Prior to the Transformer model, Recurrent Neural Networks (RNN) were commonly used for deep learning-based NLP, but the Transformer model introduced the self-attention mechanism. Various studies attempted to improve the Transformer model with different approaches. One approach that showed significant improvement was Bidirectional Encoder Representations from Transformer (BERT) [8]. BERT enhanced the Transformer model by using Masked Language Model (MLM) and Next Sentence Prediction (NSP). The main advantage of BERT is that it allows users to fine-tune the output layer of the pre-trained model for a specific downstream task.

Another approach, Text-To-Text Transfer Transformer (T5) [15], is based on the original Transformer model, which has an encoder-decoder architecture. Unlike BERT, which is an encoder-only architecture, T5 converts NLP tasks into a text-text format, where both the input and output are text. T5 also modifies some layers of the original Transformer model to suit their approaches. The T5 model can handle various downstream tasks, such as language translation, summarization, and question answering.

The field of software analysis has been inspired by deep learning-based NLP.

Code-understanding BERT (CuBERT) [9] is a BERT architecture model pre-trained with Python source codes. Since computer languages have different structures from natural languages, this study creates a Python tokenizer to adapt Python tokens to train with the BERT model. They demonstrate that CuBERT outperforms previous studies in five downstream tasks. Another study that has a similar approach, CodeT5 [19] is based on the T5 architecture as the name suggests. The difference is the target downstream tasks. CodeT5 tries to tackle Natural Language (NL) to Programming Language (PL) tasks, such as generating source code from code comments, PL to NL tasks, such as generating code comments from source code, and PL to PL tasks, such as language translation.

3 Methodology

3.1 Overview

This section describes the processes to develop the parallelizable for-loop classification model. We first gather C/C++ source files that include OpenMP directives from public GitHub repositories. We then extract for-loops from the collected source files and label them according to the presence or absence of OpenMP directives. Finally, using the set of labeled for-loops, we fine-tune a pre-trained CodeT5 model to classify whether a given for-loop can be parallelized or not.

3.2 Data Collection

To the best of our knowledge, there are no open datasets for building models that can predict parallel for-loops (*i.e.,* a collection of labeled for-loops). We thus collect a large number of source files containing OpenMP directives. We first retrieve source files of GitHub repositories using Google BigQuery. BigQuery is a data warehouse service that allows storing massive amounts of data and making queries over them using an SQL-like language. Google publishes a number of public datasets[1] on BigQuery, which include source files of public GitHub repositories. From the public dataset, we extract C and C++ source files distributed under open-source licenses (*e.g.*, MIT, GPL, and Apache license). As a result, we collected 5,047,074 C files and 4,194,787 C++ files.

Next, we find source files that contain OpenMP directives. We first remove all comments from the source codes to reduce the parsing time. We then filter out files that contain the string #pragma omp, resulting in 7,810 files. We only focus on these files because they indicate that their authors are familiar with OpenMP and use OpenMP to parallelize their code. If we include files that do not use OpenMP, there may be many for-loops that could or should be parallelized, but are not annotated with OpenMP directives. These cases would generate false positives in the model evaluation and affect its accuracy.

3.3 Data Labeling

We extract for-loops from the collected files and categorize them into two classes: parallel and serial for-loops. To locate the for-loops in the collected files, we begin with generating the Abstract Syntax Tree (AST) of the source code using Clang[2]. We then traverse the AST to find the nodes representing for-loops. We record the start line, end line, and the content (*i.e.,* source code) of each for-loop as an instance.

Finally, we assign each for-loop with a label, either *parallel* or *serial*, based on its parent node in the AST. If the parent node is an OpenMP directive (*e.g.*, OMPForDirective, OMPParallelForDirective and OMPDistribute-

ParallelForDirective), we label the for-loop as a parallel. Otherwise, we label it as serial.

[1] https://cloud.google.com/bigquery/public-data.

[2] https://clang.llvm.org/.

Note that we exclude for-loops from the dataset if the loop or its content does not follow the language syntax or has invalid OpenMP directives. This is to prevent our model from learning invalid syntax.

3.4 Fine-Tuning the CodeT5 Model

To predict whether a for-loop is parallelizable, we fine-tune a pre-trained transfer model using the dataset we built. We employ the CodeT5 [19] model provided by Salesforce, which is pre-trained on multiple programming languages and shows a state-of-the-art performance on code-related tasks. Out of the various sizes of the CodeT5 model, we use the smallest model, CodeT5-small[3], to reduce the computational cost for fine-tuning. This may compromise the performance compared to larger models, but it allows us to show the minimum performance, which is consistent with other studies that use CodeT5 [12].

While the model has already been trained with 1,000,000 C source files, it is not trained with C++ source files [19]. However, since C++ is a superset of C and has a similar syntax, we decided not to further pre-train the CodeT5 model on C++ source files. We employ the gradient accumulation technique for training the model with a larger batch size due to memory constraints and early stopping to avoid overfitting.

4 Evaluation Setup

4.1 Hardware and Software

To fine-tune the model, we used a server equipped with two Intel Xeon Gold 6230R CPUs, 256 GB of memory, and two NVIDIA A100 40 GB PCIe cards. We used Clang 16 to parse the source codes and extract ASTs. To fine-tune the CodeT5 model, we used Transformers[4] 4.26 along with PyTorch 1.13 and CUDA 11.6.

4.2 Fine-Tuning

The fine-tuning dataset we built contains 56,817 parallel for-loops and 95,188 serial for-loops. However, training with such an imbalanced dataset can degrade the performance of the model. We thus apply under-sampling to balance the number of parallel and serial for-loops in the training dataset (*i.e.,* 45,454 parallel for-loops and 45,454 serial for-loops).

We split the dataset into training and testing data using an 80–20 ratio while keeping the ratio, maintaining the balance of parallel and serial for-loops. We fine-tune the CodeT5 model with different batch sizes (128, 256, 512, 1024) and a learning rate of $2 \times 10-4$. After fine-tuning, we found the best number of epochs for each batch size was: 27 epochs for 128 batch size, 28 for 256, 34 for 512, and 27 for 1024.

[3] https://huggingface.co/Salesforce/codet5-small.

[4] https://github.com/huggingface/transformers.

4.3 Evaluation Benchmarks

We evaluate the performance of our models using the following two datasets:

GitHub Projects: This is the testing set of the dataset we built. Since some duplicate for-loops appear in the dataset, we removed the duplicate for-loops to avoid biases in the evaluation result. The final testing dataset consisted of 3,276 parallelizable for-loops and 9,074 serial for-loops. We used this dataset to evaluate the performance of our model in predicting whether a given for-loop was parallelized with OpenMP directives by the developers. The input to the model was the code of a for-loop without any labels or OpenMP directives (*i.e.,* no data-leak occurs).

NAS Parallel Benchmarks: We employ the well-known NAS Parallel Benchmarks (NPB) [4, 11] because it has both serial and parallel versions written in C++ and OpenMP. NPB consists of eight benchmarks, classified into kernels and pseudo-applications. The kernels consist of five benchmarks: Embarrassingly Parallel (EP), Multi-Grid (MG), Conjugate Gradient (CG), discrete 3D fast Fourier Transform (FT), and Integer Sort (IS). The pseudo-applications consist of three benchmarks: Block Tri-diagonal solver (BT), Scalar Penta-diagonal solver (SP), and Lower-Upper gauss-seidel solver (LU).

4.4 Baseline Models

We compare our model with Clava [2], a source-to-source compiler that has a library named *Autopar* for automatic parallelization using static analysis. Clava achieves state-of-the-art performance in this task. To replicate the results of Clava, we created a script for selecting the for-loop from NPB Benchmarks. The script queries all the loops, identifies parallelizable for-loops, generates an OpenMP directive for each parallelizable for-loop, leaves the outermost loop that has been parallelized untouched, and comments out the inner parallelized loop.

4.5 Performance Metrics

We evaluate the performance of binary classification, where each loop is classified as either a parallelizable loop or a serial loop. We employ four metrics to measure the performance of the predictions: accuracy, F1 score, precision, and recall.

Accuracy. Represents how accurately the model can classify parallel or serial for-loop, which can be formulated as follows:

$$Accuracy = \frac{\#\ of\ \ correct\ predictions}{\#\ of\ \ predictions} \tag{1}$$

F1 Score. Is the harmonic mean of precision and recall. Precision measures how accurately the parallel for-loops are predicted, while recall measures how completely the parallel for-loops are detected. As precision and recall are well-known to have a trade-off relationship, the F1 score is frequently used to balance them. Precision, recall, and F1-score are defined as follows:

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2}$$

$$Precision = \frac{\#\ of\ \ correct\ predicted\ parallel\ for - loops}{\#\ of\ \ predicted\ parallel\ for - loops} \tag{3}$$

$$Recall = \frac{\#\ of\ \ correct\ predictied\ parallel\ for - loops}{\#\ of\ \ parallel\ for - loops} \tag{4}$$

Note that Clava analyzes the source code and suggests the appropriate OpenMP directives for parallelization (*e.g.,* "#pragma omp parallel for", "#pragma omp for" and "#pragma omp for private(i)"). However, our evaluation only verifies whether the loop is parallelizable or not. Therefore, we consider the output to be correct even if Clava produces incorrect OpenMP directives, as long as the loop can be parallelized.

5 Evaluation Result

5.1 GitHub Projects

Table 1 shows the performance of the GitHub project evaluation, using our proposed models trained with different batch sizes. Note that the baseline requires compilable source code but most of the files in this experiment could not be compiled. We, therefore, do not show the performance of the baseline models.

All the models show high performance around 0.9 in terms of all the performance measures. In particular, the model trained with a batch size of 512 achieved the highest accuracy (0.944), F1 score (0.895), and recall (0.902). While the difference between the highest and the lowest scores is smaller than 2%, the models with larger batch sizes (*i.e.,* 512 and 1024) show slightly better performance in terms of accuracy and F1 score.

Table 1. GitHub Project Evaluation Results.

Model	Accuracy	F1	Precision	Recall
128	0.941	0.889	**0.893**	0.885
256	0.942	0.891	**0.893**	0.889
512	**0.944**	**0.895**	0.889	**0.902**
1024	0.942	0.892	0.888	0.895

5.2 NAS Parallel Benchmarks

Table 2 shows the performance of the NPB benchmark evaluation using our proposed models and the baseline method (*i.e.,* AutoPar-Clava). Similar to the GitHub project evaluation, the proposed models show high accuracy scores (around 0.9) for all batch sizes. On the other hand, in terms of F1, precision, and recall, there are larger differences between the highest and lowest scores than that in the GitHub project evaluation.

Table 2. NPB Benchmark Evaluation Results.

Model	Accuracy	F1	Precision	Recall
128	0.912	0.646	0.707	0.594
256	0.894	0.679	0.575	**0.830**
512	**0.816**	0.663	**0.722**	0.613
1024	0.911	**0.713**	0.630	0.820
Clava	0.905	0.669	0.628	0.716

Specifically, the difference between the highest and lowest F1 scores is approximately 10%.

In addition, whereas the highest F1 score is shown by the 1024 batch size model, the highest precision and recall scores are shown by the 512 and 256 batch size models, respectively. However, the 512 and 256 batch size models show relatively lower recall and precision scores, respectively, which result in their low F1 values. Similar to the GitHub project evaluation, using the models built with 512 or 1024 batch sizes tend to show stable results.

Compared with the baseline model, AutoPar-Clava, most of the proposed models show higher performance on all measures. Notably, even the lowest performance of the proposed method shows a similar score to that achieved by the baseline model (*i.e.,* 1.2% and 3.4% difference in terms of accuracy and F1, respectively).

6 Discussion

6.1 Why Can Our Model Predict Correctly?

In the NPB benchmarks evaluation, we compared our models with a state-of-the-art static analysis tool, AutoPar-Clava. We observed 27 for-loops that the proposed method correctly predicted as parallelizable, but Clava did not. We attribute this difference to two possible reasons.

The first reason is because of the limitations of static analyzers. As discussed in the introduction, static analyzers may fail to recognize parallelizable loops if runtime information or application-specific knowledge is required. Figure 1 shows an example of this case. The data dependency analysis of Clava concludes that the loop cannot be parallelized because key_buff_ptr_global uses key_buff2 as an index and is assigned to k variable. However, this loop is actually parallelizable by privatizing k and i, the index of key_buff2, as shown in the OpenMP directive in Fig. 1.

The second reason involves the computational complexity of static analysis. Figure 2 shows an example where Clava encounters an out-of-memory exception. The snippet has three nested loops and the loop body is approximately 300 lines of code. Clava tried to analyze all the dependencies in the loops to determine whether the loops can be parallelized, but it consumes too much memory and ends up throwing an out-of-memory exception. On the other hand, the memory consumption of our model is constant with respect to input size and complexity.

```
#pragma omp parallel for private(i,j,k,k1) schedule(dynamic)
for (j = 0; j < NUM_BUCKETS; j++) {
    k1 = (j > 0)? bucket_ptrs[j-1] : 0;
    for (i = k1; i < bucket_ptrs[j]; i++) {
        k = --key_buff_ptr_global[key_buff2[i]];
        key_array[k] = key_buff2[i];
}}
```

Fig. 1. An example of parallelizable code that the proposed method correctly predicted, but Clava did not due to dependency analysis failure (line 511 in full_verify function from IS benchmark OpenMP version).

```
for (k = 1; k <= grid_points[2] - 2; k++) {
    for (j = 1; j <= grid_points[1] - 2; j++) {
        for (i = 0; i <= isize; i++) {
            tmp1 = rho_i[k][j][i];
            tmp2 = tmp1 * tmp1;
            ... 287 more lines
```

Fig. 2. An example of parallelizable code that the proposed method correctly predicted, but Clava did not due to out-of-memory (line 2323 in x_solve function from BT benchmark).

6.2 Why Does Our Model Mispredict?

We observed 16 for-loops that Clava can correctly predict parallelizable code, but the proposed method cannot. Figure 3 shows the case where Clava predicted correctly, but our model with a batch size of 256 did not. This may be attributed to the complexity of the code, which involves three nested loops and computations with multiple four-dimensional arrays. If the model is not trained on enough data that contains such complex code, the proposed model would predict it as non-parallelizable code, which resulted in false negatives.

The cause of this issue is similar to the one encountered by the static analysis tool in the previous section, namely that the input code is too complex for the model to handle. A possible solution for the static analysis tool is to manually implement new rules that can handle such complex cases. However, for the proposed method, this problem can be relatively easily prevented by increasing the amount of data used for fine-tuning. Therefore, our deep-learning method has an advantage over static analysis tools in terms of performance and less effort for practitioners.

```
for (j = 1; j <= grid_points[1] - 2; j++) {
    for (i = 1; i <= grid_points[0] - 2; i++) {
        for (m = 0; m < 5; m++) {
            rhs[k][j][i][m]=rhs[k][j][i][m]-dssp*(u[k-2][j][i][m]-4.*u[k-1][j][i][
                m]+5.*u[k][j][i][m]);
}}}
```

Fig. 3. An example of parallelizable code that Clava correctly predicted, but the proposed method did not due to complexity (line 1053 in compute_rhs function from BT benchmark).

7 Conclusion and Future Work

This study proposed a deep learning-based NLP model for automatic parallelization of source code. We fine-tuned CodeT5, a state-of-the-art transformer model, with the source code from GitHub repositories that contain OpenMP directives, so that it can identify loops that can be parallelized.

Throughout our evaluation using source files from GitHub projects and the NAS Parallel Benchmarks, we observed that our proposed models outperform the baseline method, AutoPar-Clava. Specifically, the proposed method achieved an F1 score of 0.895 in the GitHub projects evaluation and an F1 score of 0.713 in the NAS Parallel Benchmarks evaluation.

As future work, we plan to train deep-learning models to suggest OpenMP directives after identifying the parallelizable loops using the method proposed in this study. Also, we plan to evaluate the performance improvement by our automatic parallelization tool.

Acknowledgments. This work was partially supported by JSPS KAKENHI Grant Numbers JP21H03416 and JP23K16890.

References

1. Automatic Parallelization—intel.com. https://www.intel.com/content/www/us/en/docs/fortran-compiler/developer-guide-reference/2023-1/automatic-parallelization.html. Accessed 07 June 2023
2. Arabnejad, H., Bispo, J., Barbosa, J.G., Cardoso, J.M.P.: Autopar-clava: an automatic parallelization source-to-source tool for C code applications. In: Proceedings of the 9th Workshop on Parallel Programming and RunTime Management Techniques for Manycore Architectures and 7th Workshop on Design Tools and Architectures for Multicore Embedded Computing Platforms, pp. 13–19 (2018)
3. Bae, H., et al.: The cetus source-to-source compiler infrastructure: overview and evaluation. Int. J. Parallel Program. **41**(6), 753–767 (2013)
4. Bailey, D.H., et al.: The NAS parallel benchmarks. Int. J. High Perform. Comput. Appl. **5**(3), 63–73 (1991)
5. Bispo, J., Cardoso, J.M.P.: Clava: C/C++ source-to-source compilation using LARA. SoftwareX **12**, 100565 (2020)
6. Dagum, L., Menon, R.: Openmp: an industry-standard API for shared-memory programming. IEEE Comput. Sci. Eng. **5**(1), 46–55 (1998)
7. Dave, C., Bae, H., Min, S., Lee, S., Eigenmann, R., Midkiff, P.: Cetus: a source-to-source compiler infrastructure for multicores. Computer **42**(11), 36–42 (2009)
8. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186 (2019)
9. Kanade, A., Maniatis, P., Balakrishnan, G., Shi, K.: Learning and evaluating contextual embedding of source code. In: Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 5110–5121 (2020)
10. Liao, C., Quinlan, D.J., Willcock, J., Panas, T.: Extending automatic parallelization to optimize high-level abstractions for multicore. In: Proceedings of the 5th International Workshop on OpenMP: Evolving OpenMP in an Age of Extreme Parallelism, vol. 5568, pp. 28–41 (2009)

11. Löff, J., et al.: The NAS parallel benchmarks for evaluating C++ parallel programming frameworks on shared-memory architectures. Future Gener. Comput. Syst. **125**, 743–757 (2021)
12. Mastropaolo, A., et al.: Studying the usage of text-to-text transfer transformer to support code-related tasks. In: Proceedings of the 43rd IEEE/ACM International Conference on Software Engineering, pp. 336–347 (2021)
13. Mendonca, G.S.D., Guimarães, B.C.F., Alves, P., Pereira, M.M., Araujo, G., Pereira, F.M.Q.: DawnCC: automatic annotation for data parallelism and offloading. ACM Trans. Archit. Code Optim. **14**(2), 13:1–13:25 (2017)
14. Quinlan, D., Liao, C.: The ROSE source-to-source compiler infrastructure. In: Cetus Users and Compiler Infrastructure Workshop, in Conjunction with PACT, vol. 2011, p. 1 (2011)
15. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 140:1–140:67 (2020)
16. Szuppe, J.: Boost.compute: a parallel computing library for C++ based on OpenCL. In: Proceedings of the 4th International Workshop on OpenCL, pp. 15:1–15:39 (2016)
17. Tierney, L., Rossini, A.J., Li, N.: Snow: a parallel computing framework for the R system. Int. J. Parallel Program. **37**(1), 78–90 (2009)
18. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, pp. 5998–6008 (2017)
19. Wang, Y., Wang, W., Joty, S.R., Hoi, S.C.H.: Codet5: identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 8696–8708 (2021)

SAHF-LightPoseResNet: Spatially-Aware Attention-Based Hierarchical Features Enabled Lightweight PoseResNet for 2D Human Pose Estimation

Ali Zakir[1](✉), Sartaj Ahmed Salman[1], and Hiroki Takahashi[1,2]

[1] Graduate School of Informatics and Engineering, Department of Informatics, The University of Electro-Communications, Tokyo, Japan
{a2240012,s2140019}@edu.cc.uec.ac.jp, rocky@inf.uec.ac.jp

[2] Artificial Intelligence Exploration Research Center/Meta-Networking Research Center, The University of Electro-Communications, Tokyo, Japan

Abstract. In recent years, 2D human pose estimation (HPE) has become increasingly important in complex computer vision tasks, including understanding human behavior and interaction. Despite challenges like occlusion, unfavorable lighting, and motion blur, deep learning techniques have revolutionized 2D HPE by allowing automatic feature learning from data and improving generalization. We proposed a new model called Spatially-aware Attention-based Hierarchical Features Enabled Lightweight PoseResNet (SAHF-LightPoseResNet) for 2D HPE. This model extends the simple baseline network by using Spatially-aware Attention-based Hierarchical Features to enhance accuracy while minimizing parameters. The proposed model efficiently captures finer details by incorporating ResNet18, Global Context Blocks, and a novel SAHF module. Our SAHF-LightPoseResNet approach demonstrates superior performance compared to existing state-of-the-art methods, achieving PCKh@0.5 a of 90.8 and a Mean@0.1 metric of 41.1, highlighting its enhanced accuracy and efficiency. This model has important practical applications in robotics, gaming, and human-computer interaction, where accurate and efficient 2D HPE is essential.

Keywords: 2D human pose estimation · SAHF-LightPoseResNet · Global Context Blocks

1 Introduction

The utilization and advancement of Computer Vision (CV) technology in diverse real-world environments, including but not limited to smartphones, digital cameras, and Closed-Circuit Television (CCTV), have led to a persistent flow of extensive data in the form of images and videos. Information regarding human activities found within these data is incredibly significant. Human Pose Estimation (HPE) involves identifying and categorizing the various joints in the human body. Essentially, HPE captures each joint's coordinates, including arms, head, and torso, which are commonly referred to

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 43–54, 2024.
https://doi.org/10.1007/978-981-99-8211-0_5

as key points that define a person's posture. Over the past few decades, the automatic understanding of HPE has been a major focus of research in CV. 2D HPE offers a fundamental base for several complex CV assignments, including predicting 3D HPE, identifying human actions and motion prediction, parsing human body components, and retargeting human movements. 2D HPE offers extensive support for a wide range of applications, human behaviour understanding, identification of crowd disturbances and riots, detection of violent incidents, recognition of unusual behavior, enhancement of human-computer interaction, and enabling autonomous car driving [4].

The 2D HPE is considered challenging because it is impacted by several significant factors, such as the occlusion of keypoints, unfavorable lighting and background conditions, motion blur, and the complexity of implementing the model in real-world scenarios due to its extensive parameters [20]. Researchers employed conventional techniques like probabilistic graphical models in the early stages to tackle these challenges. However, these methods heavily relied on manually crafted features, restricting the model's generalization ability and limited performance. The progress of 2D HPE has been significantly boosted by introducing deep learning methods, which overcome the generalization limitation of hand-crafted features by enabling automatic feature learning from the data. The remarkable performance of Convolutional Neural Networks (CNNs) in 2D HPE paved the way for developing many deep learning techniques that rely on their success [3].

The main objective of this paper is to achieve high prediction accuracy while minimizing the number of parameters utilized rather than solely focusing on improving the prediction accuracy of existing approaches. The simple baseline network [17] accomplished the best outcomes compared to other top-down approaches. Its effectiveness and simplicity make it an appropriate starting point for creating more sophisticated methods for 2D HPE. In order to accomplish this, we have proposed a unique approach named SAHF-LightPoseResNet, which builds upon the basic framework network by incorporating Spatially-aware Attention-based Hierarchical Features. To reduce the complexity of the model, we have opted to use ResNet18 instead of the more intricate models like ResNet50, 101, or 152, which contain a more significant number of parameters. In our implementation of ResNet18, we have discarded the average pooling segment and the last fully connected segment and exclusively incorporated convolutional layers. Additionally, we have added two deconvolution layers to improve the model's visual processing capabilities and overcome quantization distortion resulting from a large output stride size. We have incorporated Global Context Blocks (GCBs) [2] into the proposed model to equip down-sampler and up-sampler modules with powerful global context features. Our newly developed module, SAHF, merges the feature representations extracted from multiple down-sampler layers, and then enhances these representations by utilizing spatial attention. Finally, SAHF allocates the enriched feature representations to their respective layers in the up-sampler, avoiding conventional skip connections [12, 13]. This approach produced hierarchical representations with spatial awareness and can more effectively capture finer details.

The threefold contribution of the SAHF-LightPoseResNet model can be summarized as follows:

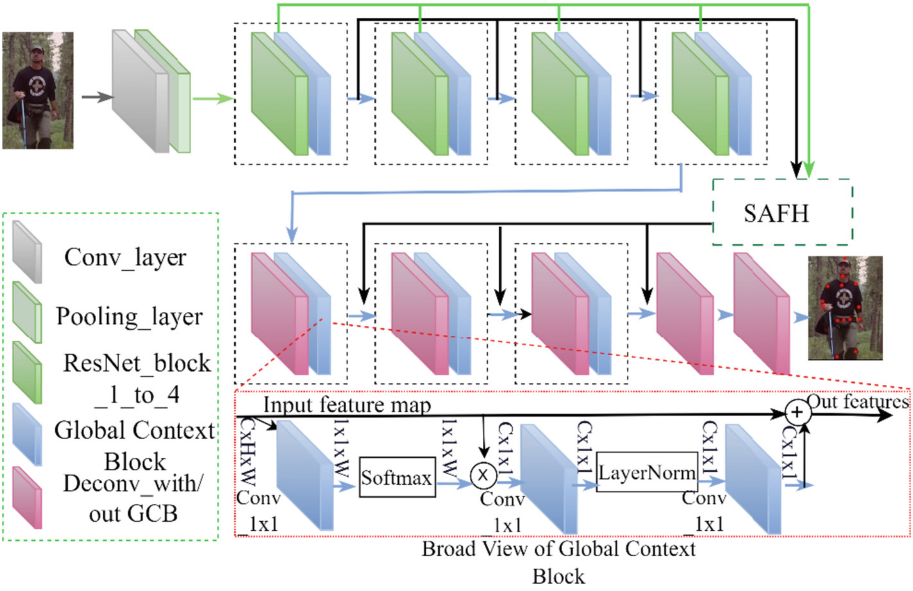


Fig. 1. The proposed SAHF-LightPoseResNet framework.

- We developed a novel model called SAHF-LightPoseResNet, where ResNet18 was used instead of more complex models to reduce the model's complexity. Two deconvolution layers are utilized to improve the model's visual processing capabilities and overcome quantization distortion resulting from a large output stride size. GCBs are incorporated into the model's down-sampler and up-sampler modules to augment it with potent global context features.
- Our proposed SAHF module combines features extracted from different down-sampler layers, which are then enhanced using a spatial attention mechanism and distributed to the corresponding up-sampler layers. As a result, hierarchical representations with spatial awareness are generated, which can capture finer details more effectively.
- Experiments were carried out on the MPII dataset to verify the efficiency of the proposed approach. Evaluation of the quantitative and qualitative outcomes indicated that our model achieved better accuracy and lower computational cost than existing 2D human pose estimation techniques.

This article follows a structured approach with several sections. Section 2 presents an overview of prior research conducted in the same field. Section 3 elaborates on the comprehensive methodology of our Proposed SAHF-LightPoseNet. Section 4 covers pertinent information regarding the experimental setup and implementation details. An analysis of both qualitative and quantitative results is exhibited in Sect. 5. The final section, Sect. 6, draws the conclusion and lays out plans for future exploration.

2 Related Works

Deep learning approaches are utilized in designing network architectures for 2D HPE to extract robust features that span from low to high levels. These approaches are typically categorized into two frameworks: the top-down and bottom-up frameworks. The method of the top-down paradigm involves a sequential process where the initial step is to identify the human bounding boxes in an image, followed by executing the single HPE for every identified box. This type of approach is not a suitable method for managing large crowds as the computational time for the second step increases in association with the number of individuals present [1, 6]. A. Toshev et al. [15] has made a pioneering contribution to the field of HPE by introducing CNN for the first time.

They leveraged the CNN's robust fitting capability to regress the coordinates of human joints and implemented a cascading structure to refine the outcomes continuously. Though, the model tends to overfit because the weights of the fully connected layer depend on the distribution of the training dataset. Convolutional Pose Machine (CPM) [16] and stacked hourglass networks [12] solved this issue by predicting heatmaps of 2D joint locations. Two main object detection techniques exist in 2D HPE: the RCNN [7] series and the SSD series [10]. The RCNN series employs a complicated network structure that achieves high accuracy. Introduced the Mask-RCNN approach, which builds upon the faster-RCNN architecture [7] by incorporating keypoint prediction. As a result, this method achieves excellent results in HPE, demonstrating strong competitiveness in this domain. Conversely, the SSD series offers an average compromise between precision and Y. Chen et al. [5] presents the concept of a cascaded pyramid network (CPN) that uses Global-Net to identify simple keypoints and Refine-Net to handle more challenging keypoints. To be more precise, Refine-Net includes multiple standard convolutional layers that merge feature representations from all levels of GlobalNet.

The process of bottom-up methods start with detecting keypoints for every human instance present in an image. Subsequently, the keypoints of the same individual are joined to form skeletons of multiple instances. This grouping optimization problem is crucial in determining the outcome of the bottom-up approach. Some representative methods utilize this approach, and they are [3, 14]. Open-Pose, as described in [3], utilized two branches - one of which employed a CNN to predict all keypoints based on heatmaps, and the other used a CNN to acquire part affinity fields. The part affinity fields represent 2D direction vectors, and they serve as a confidence metric to determine if the keypoints are associated with the same person. Ultimately, both branches are merged to generate the concluding prediction. The approach known as associative embedding [11], derived from Hourglass [12], is end-to-end trainable. The source detected and accumulated keypoints in one step without requiring two separate processes. Implementing bottom-up approaches can be challenging due to the difficulty of combining information from multiple scales and grouping features together. Even with the introduction of effective grouping procedures, these methods still struggle to contest top-down strategies for pose estimation. In recent times, the majority of cutting-edge outcomes have been achieved through top-down methodologies. Our research traced the top-down approach and developed a successful 2D HPE model. This addresses the issue of top-down approaches by modifying a baseline network with Spatially-aware Attention-based Hierarchical Features. We utilized a simpler ResNet18 model and removed specific layers to reduce complexity. We then added deconvolution layers and Global Context Blocks to improve visual processing and global context features. The proposed SAHF module combines and enhances feature representations from various layers, enabling better capture of finer details through hierarchical representations with spatial awareness.

3 Our Proposed SAHF-LightPoseNet

To formally define the task of estimating human pose, we can state it as follows: when given an RGB image or video frame I as input, the goal is to estimate pose P of human(s) present in the data. The pose P can be represented as a set of K's keypoint positions, where a two-dimensional coordinate represents each keypoint (x_k, y_k), and K can vary depending on the dataset. Therefore, we aim to estimate the pose $P = \{P_i\}_{i=1}^{n}$ for all n individuals in the input data. Our research builds upon the simple baseline network for 2D HPE that was previously developed. Our proposed SAHF-LightPoseResNet is shown in Fig. 2. Further information and comprehensive explanations regarding the components of SAHF-LightPoseResNet are introduced in the following subsections.

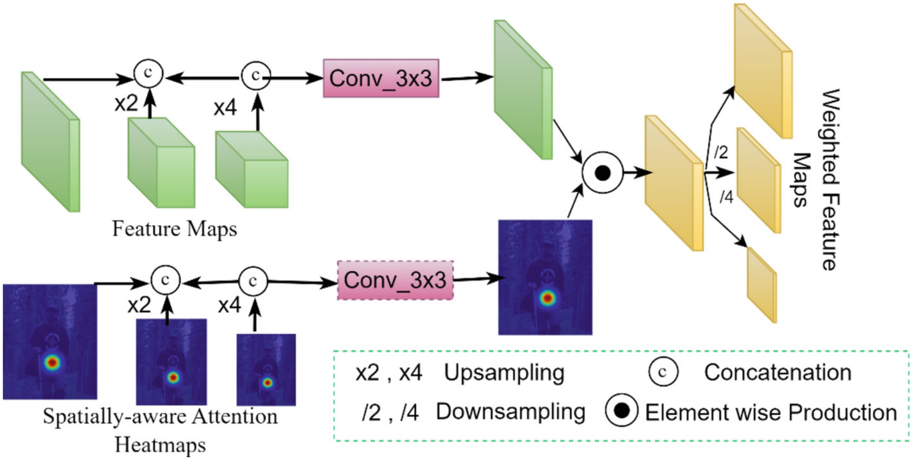


Fig. 2. Proposed SAHF Module.

3.1 Enhancing Backbone Model with Modified ResNet and Deconvolution Module

The structure of the autoencoder network is commonly utilized for dense labeling tasks. To achieve this, we employed an autoencoder network structure that slowly decreases the resolution of embeddings to capture extended-range details, which subsequently increases feature maps while recovering spatial resolution. Hourglass and simple baseline networks create smaller output feature maps than their input feature maps, which are then resized using a simple transformation technique that can cause quantization errors.

When data processing is biased, prediction errors can occur due to horizontal flipping and how the model processes the output resolution [9].

We incorporated two deconvolution modules into our approach to tackling the above-mentioned challenges. These modules were designed to generate a complete output feature map and were integrated within the architecture of the simple baseline network. We opted to use ResNet 18 and 34, which have fewer parameters compared to more complex ResNet models like 50, 101, or 152. We modified ResNet [8] by removing the average pooling segment and fully connected part and replacing them with four ResNet blocks after a convolutional and pooling layer. The first set of layers in the network, which includes a convolutional layer and a pooling layer, reduces the size of the feature maps by half. As the input passes through each block of the network, additional convolutional layers are used to decrease the feature maps by two strides while simultaneously increasing the number of filters by a factor of two. We added five deconvolutional modules with batch normalization and Hardswish activation, each doubling the feature resolution map until the output matches the input. The proposed architecture is illustrated in Fig. 1. The 4th and 5th deconvolutional layers have channel sizes of 128 and 64, respectively.

3.2 Amplifying Model Performance with Global Context Blocks

In computer vision, a global context block is a module designed to capture the overall spatial information of an input feature map, aiming to improve object recognition in an image. In convolutional layers, the association among pixels is only considered within a local neighborhood, and baseline network. We opted to use ResNet 18 and 34, which have fewer parameters compared to more complex ResNet capturing long-range dependencies requires multiple convolution layers. To address this limitation, researchers proposed a non-local operation [18], which employed a self-attention mechanism from [19] to model long-range dependencies. Using a global network creates an attention map tailored to each query position, enabling the collection of contextual features that can then be integrated into the features of the corresponding position. GCNet is presented as a highly well-organized and operative method for global context modeling [2]. This method employs a query-agnostic attention map to generate a contextual representation that can be globally shared and then incorporates it into the features of each query location in the network.

Our proposed method uses global context blocks [2] to enhance the spatial information of input feature maps. Specifically, as illustrated by blue blocks in Fig. 1 global context blocks are incorporated into each ResNet block as well as the first three blocks of the deconvolution modules. We generate a spatially-aware attention heatmap using a 1 × 1 convolution and SoftMax to produce attention weights, which are then used in attention pooling to extract a global context feature. Channel-wise dependencies are obtained using the bottleneck transform technique. Afterward, the resulting global context features are combined with the features of each position in the network, as shown in the following equation.

$$f_g = \sum_{i=1}^{h} \sum_{j=1}^{w} w_{ij} f_{ij} \tag{1}$$

where in Eq. 1, f_g represent the global context feature, $h\ and\ w$ are the height and width of the input feature map, w_{ij} is the attention weights at position (i, j) and $f\,(i, j)$ is the feature vector at the position (i, j).

3.3 SAHF Module

The Spatially-aware Attention-based Hierarchical Features (SAHF) module overcomes the limitations of earlier frameworks, such as the simple baseline framework, which did not integrate skip connections [12, 13]. These connections have proven effective in U-Net and hourglass networks for retaining spatial information at each feature map, allowing for an efficient transfer of spatial information across the network, leading to improved localization.

Our proposed SAHF module, depicted in Fig. 2, is an alternative to traditional skip connections used in previous works [12, 13]. The SAHF module combines hierarchical features from different layers, using spatial attention to enhance features. It receives feature maps from the first three Global Context Blocks, ResNet blocks, and Spatially-aware attention feature maps. These feature maps are multiplied elementwise to generate enhanced features, which are then allocated to the deconvolution modules, excluding the last one. The Spatially-aware attention technique focuses on locations related to pose estimation and helps generate helpful detail while suppressing background information. The enhanced features from the SAHF module improve the capabilities of related deconvolution models, leading to an overall improvement in network performance as shown in table 1 and visualize the performance of SAHF in Fig. 3 (a) and (b).

3.4 Heatmap Joint Prediction

Our model predicts joint positions at the pixel level by converting them into heatmaps within a bounding box, using a 2D Gaussian function to generate ground truth. The resulting heatmap represents the probability of a joint being located at each pixel.

$$H_{k(x,y)} = exp(\frac{-[(x - y_k)^2 + (y - y_k)^2]}{2\sigma^2} \tag{2}$$

In Eq. 2 H_k represent heatmap for kth joint where $k \in \{1, 2, \ldots, K\}$, and (x, y) show the position of the specified pixel in the heatmap. The kth joints coordinated are denoted by (x_k, y_k). The value for spatial variance σ is set to 12 in this experiment.

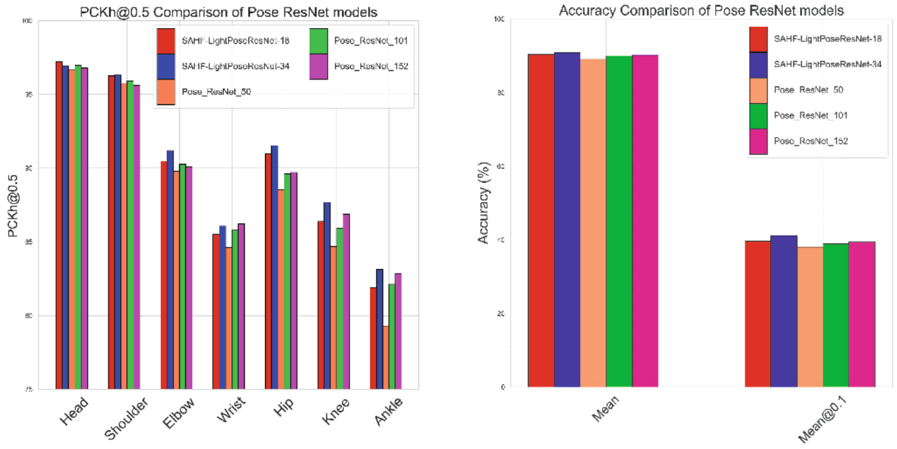
4 Experimental Setup

4.1 Dataset

Our experimentation to evaluate the effectiveness of our proposed model was carried out using the extensively recognized MPII (Max Planck Institute for Informatics) Dataset [21]. This comprehensive dataset comprises more than 25,000 annotated images, capturing over 40,000 individuals, each mapped with 16 distinct key-points. This substantial data set has been strategically split into two subsets, one for training and the other for testing. A total of 28,000 images were employed to build and refine our model in the training phase. Subsequently, a separate set of 11,000 images was exclusively leveraged to test the model's performance, providing an objective measure of model's robustness and accuracy.

4.2 Implementation Details

We utilized data augmentation techniques to improve the model's ability to handle scale variance and spatial rotation, including random horizontal flip, rotation within -40 to + 40 degrees, and scaling between 0.7 to 1.3 in our approach. Our designed model was implemented using PyTorch. The training process included a learning rate of 1e-05, a batch size of 16, a number of workers set to 6, and 150 epochs.



(a) **(b)**

Fig. 3. (a) Illustration of PCKh@0.5 Results: Proposed Model and Simple Baseline Models (b) Graphical Analysis of Mean and Mean@0.1: Proposed Models and Simple Baseline

In our research, we implemented three key components to ensure precise model training and enhanced model performance: MSE loss function, AdamW optimizer, and Hard Swish activation function.

The Mean Square Error (MSE) loss function, which has been effectively utilized in previous works such as [6, 17], was chosen to evaluate the model's error. The formula for MSE is presented below.

$$L = \frac{\sum_{k=1}^{K} \|H_k - \widehat{H}_k\|}{K} \tag{3}$$

In Eq. 3, $\widehat{H}_k$ represent estimated heatmap for the k$^{\text{th}}$ joint, where as H_K is the heatmap for the k$^{\text{th}}$ joint $k \in \{1, 2, \ldots, K\}$.

The model optimization process was further enhanced using a variant of the Adam optimizer, AdamW. Distinct from the original Adam optimizer, AdamW separates weight decay from the learning rate, enabling independent optimization and significant reduction in overfitting.

Finally, our research employed the Hard Swish activation function. This superior function offers significant advantages over the commonly used ReLU function, including superior accuracy, efficiency, smoother gradient, and the ability to address the 'dying neurons' issue often seen with ReLU. Utilizing Hard Swish, we witnessed an overall performance improvement in our neural network model and achieved superior experimental results, suggesting its potential benefits across various deep-learning applications.

4.3 Evaluation Metrics

We used PCK (Percentage of Correct Keypoints) and Mean@0.1, widely used evaluation metrics in HPE tasks. PCKh, a variation of PCK, compares the predicted and actual keypoints using the head bone link length as a reference. The prediction is considered correct if the distance between the predicted and actual keypoints is less than 50% of the head bone link length (PCKh@0.5). Mean@0.1, on the other hand, measures the average distance between predicted and actual keypoints, normalized by the head bone link length, and is scale-invariant.

5 Experimental Results and Discussion

Our model was trained on different input sizes, namely 256 × 256, 288 × 384, and 384 × 384. The only exception was the simple baseline model, which did not use the 288 × 384 input size. In Fig. 4, you can observe the inference results of the LightPoseResNet-18 model on the MPII dataset. To compare the performance of our proposed SAHF-LightPoseResNet model with that of the basic baseline model, we present their outcomes in Table 1. We also provide a visualization of each joint with PCKh@0.5 for our proposed models and the simple baseline models that used the 256 × 256 input size, as shown in Fig. 3(a). Finally, Fig. 3(b) displays both models' overall Mean and Mean@0.1 predictions using the 256 × 256 input size. Initially, we conducted training on SAHF-LightPoseResNet-18 using input sizes of 256 × 256. As a result, we are able to obtain PCKh@0.5 values of 89.425 and 90.297, along with mean@0.5 values of 34.483 and 39.670. These values were found to be higher than the PCKh@0.5 and Mean@0.1 values of all the basic baseline models. We conducted an experiment on the LightPoseResNet-18 model using input sizes 288x384 and 384 × 384. Our results showed that the LightPoseResNet-18 model outperformed the simple baseline. Notably, despite achieving better results, the LightPoseResNet-18 model used only 21 million parameters during the training process, which is fewer than all the simple baseline models.

Some experiments were conducted on LightPoseResNet-34 using the input sizes mentioned earlier. The outcomes revealed that the model yielded better results in terms of PCKh@0.5 and Mean@0.1, despite having fewer parameters than the simple baseline model. These findings are presented in Table 1 and visualized in Fig. 3(a) and Fig. 3(b).

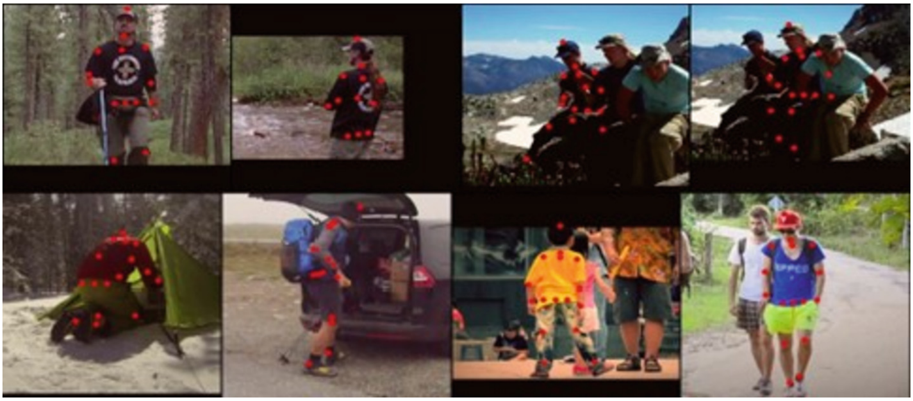


Fig. 4. Qualitative Results on MPII pose estimation result, containing viewpoint change, and occlusion and self-occlusion.

Therefore, our study demonstrates the effectiveness of the LightPoseResNet-18 and 34 models in terms of both computation and performance.

Table 1. Performance comparisons of our SAHF-LightPoseResNet with simple baseline results on MPII dataset

Model	No.par	input	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean	Mean@0.1
Pose_Resnet_50	34.0M	256x256	96.351	95.329	88.989	83.176	88.420	83.960	79.594	88.532	33.911
		384x384	96.658	95.754	89.790	84.614	88.523	84.666	79.287	89.066	38.046
Pose_Resnet_101	53.0M	256x256	96.862	95.873	89.518	84.376	88.437	84.486	80.703	89.131	34.020
		384x384	96.965	95.907	90.268	85.780	89.597	85.935	82.098	90.003	38.860
Pose_Resnet_152	68.6M	256x256	97.033	95.941	90.046	84.976	89.164	85.311	81.271	89.620	35.025
		384x384	96.794	95.618	90.080	**86.225**	89.700	86.862	82.853	90.200	39.433
SAHF-LightPoseResNet _18	21.0M	256x256	96.965	95.688	89.398	84.051	90.254	85.029	80.728	89.425	34.483
		288x384	97.169	95.788	90.131	84.462	90.341	85.331	81.696	89.766	36.435
		384x384	97.203	96.264	90.472	85.489	90.981	86.379	81.890	90.297	39.670
SAHF-LightPoseResNet _34	30.0M	256x256	97.237	95.805	90.012	84.891	90.064	85.976	81.507	89.846	36.417
		288x384	**97.271**	96.247	90.608	85.642	91.016	86.984	82.712	90.536	38.158
		384x384	96.930	**96.298**	**91.188**	86.072	**91.535**	**87.668**	**83.137**	**90.877**	**41.137**

6 Conclusion and Future Work

In this research work, we proposed SAHF-LightPoseResNet for 2D HPE. The SAHF-LightPoseResNet model is a novel approach utilizing ResNet18 to reduce complexity while achieving effective visual processing capabilities. The model's down-sampler and upsampler modules are enhanced with GCBs to provide potent global context features. The SAHF module combines and distributes features with spatial attention to produce hierarchical representations with spatial awareness that capture finer details effectively.

SAHF-LightPoseResNet performs better than basic baseline models on the MPII dataset due to improved features, better activation function, and advanced model optimizer, as simulation results indicate. In the future, our model can be utilized for 3D human pose estimation with object recognition and hand pose estimation due to its general applicability.

References

1. Bertasius, G., Feichtenhofer, C., Tran, D., Shi, J., Torresani, L.: Learning temporal pose estimation from sparsely-labeled videos. In: Advances in Neural Information Processing Systems 32 (2019)
2. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: non-local networks meet squeeze excitation networks and beyond. In: Proceedings of the IEEE/CVF international conference on computer vision workshops, p. 0 (2019)
3. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2D pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7291–7299 (2017)
4. Chen, H., Feng, R., Wu, S., Xu, H., Zhou, F., Liu, Z.: 2D human pose estimation: a survey. Multimedia Systems, pp. 1–24 (2022)
5. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7103–7112 (2018)
6. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: Rmpe: regional multi-person pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2334–2343 (2017)
7. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
9. Huang, J., Zhu, Z., Guo, F., Huang, G.: The devil is in the details: delving into unbiased data processing for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5700–5709 (2020)
10. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
11. Newell, A., Huang, Z., Deng, J.: Associative embedding: End-to-end learning for joint detection and grouping. In: Advances in Neural Information Processing Systems 30 (2017)
12. Alejandro Newell, Kaiyu Yang, Jia Deng,: Stacked hourglass networks for human pose estimation. In: Bastian Leibe, Jiri Matas, Nicu Sebe, Max Welling, (ed.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29
13. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
14. Salman, S.A., Zakir, A., Takahashi, H.: Cascaded deep graphical convolutional neural network for 2D hand pose estimation. In: International Workshop on Advanced Imaging Technology (IWAIT) 2023. vol. 12592, pp. 227–232. SPIE (2023)

15. Toshev, A., Szegedy, C.: Deeppose: human pose estimation via deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1653–1660 (2014)
16. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4724–4732 (2016)
17. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European conference on computer vision (ECCV), pp. 466–481 (2018)
18. Wang, X., Ross, G., Abhinav, G., He, K.: non local neural networks. In: Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition, pp. 7794–7803. (2018)
19. Vaswani, A., et al.:Attention is all you need. In: Advances in Neural Information Processing Systems 30 (2017)
20. Zheng, C., et al.: Deep learning-based human pose estimation: a survey. arXiv preprint arXiv:2012.13392 (2020)
21. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: In: 2D human pose estimation: new benchmark and state of the art analysis. In: Proceedings of the IEEE Conference on computer Vision and Pattern Recognition, pp. 3686–3693 (2014)

List-Based Workflow Scheduling Utilizing Deep Reinforcement Learning

Wei-Cheng Tseng and Kuo-Chan Huang(✉)

Department of Computer Science, National Taichung University of Education, No.227,
Minsheng Rd., West Dist., Taichung City 403012, Taiwan (R.O.C.)
kchuang@mail.ntcu.edu.tw

Abstract. Workflow scheduling is a well-known NP-complete research problem with wide applications and increasing importance. Traditionally, heuristic and guided random search methods, e.g. genetic algorithm, are the two major categories of scheduling approaches developed to tackle this challenging problem. With the rise of deep reinforcement learning (DRL), this paper tries to apply it to solve the workflow scheduling problem in two different ways. The first way utilizes DRL as an iterative optimization method to find the best schedule for a specific workflow. In the second way, DRL is used to train a neural network which could be adopted to schedule new workflows not in the training set. Our DRL-based workflow scheduling method is based on the policy gradient (PG) reinforcement learning algorithm and utilizes a convolutional neural network (CNN). Experimental results show that our DRL-based method can produce more efficient workflow execution schedules, compared to the state of the art of heuristic-based scheduling algorithms. The superior performance of the DRL-based method indicates a promising direction for future research work on workflow scheduling.

Keywords: Deep Reinforcement Learning · Workflow Scheduling · Policy Gradient · Convolutional Neural Network

1 Introduction

Workflow is a common task-parallel model for parallel computing [5] with various applications, ranging from instruction scheduling within parallel compilers in earlier days to modern high performance computing on cloud platforms. A workflow is usually represented by a Directed Acyclic Graph (DAG), G = (V,E), where V is the set of nodes representing computational tasks in the workflow, and E is the set of edges describing precedence relations among tasks. Figure 1 shows an example of workflow DAG structure. Each node, i.e. a computational task, is annotated with a value indicating the required execution time. The number next to an edge represents the required data transfer time between the two connected tasks.

The workflow scheduling problem on a parallel computing platform is to determine the start time and processor allocation for each task in order to optimize a specific goal, e.g. the execution makespan between the start time of the first task to the finish time of the

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 55–61, 2024.
https://doi.org/10.1007/978-981-99-8211-0_6

last task. Workflow scheduling is a well-known NP-complete problem [5]. Therefore, instead of finding optimal solutions, various practical algorithms were developed to produce good-enough schedules in reasonable time. Most previous workflow scheduling methods can be roughly categorized into two types [1]: heuristic-based and guided random search approaches, e.g. genetic algorithm.

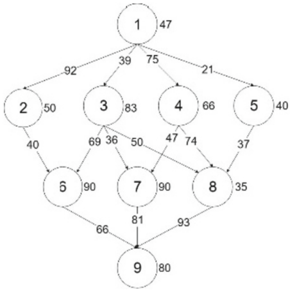


Fig. 1. Workflow structure represented by a DAG

Recently, deep reinforcement learning (DRL) has shown great achievements and potential in many application fields, e.g. game playing [6]. In this paper, we try to apply DRL to the workflow scheduling problem. Reinforcement learning (RL) [7] is a subfield of machine learning where an agent learns to make the best decisions by trial and error, through interaction with the environment, in order to maximize a specific kind of cumulative reward. DRL combines reinforcement learning and deep learning, allowing agents to make decisions from unstructured input data and large complex state space. Our DRL-based scheduling method is based on the policy gradient (PG) [7] reinforcement learning algorithm and utilizes a convolutional neural network (CNN) [8].

The proposed DRL-based workflow scheduling method could be used in two different ways. The first way utilizes DRL as an iterative optimization method to find the best schedule for a single workflow from scratch. In the second way, DRL is adopted to train a neural network based on a set of workflows first. The neural network model could then be used to schedule new workflows not in the training set. Experimental results show that our DRL-based method can produce more efficient workflow execution schedules in terms of makepsan, compared to the most famous heuristic-based workflow scheduling algorithm HEFT [1] and the state of the art of heuristic-based scheduling algorithms IPPTS [2]. The superior performance of the proposed DRL-based method indicates a promising direction for future research work on workflow scheduling.

2 Related Work

Heuristic-based workflow scheduling methods rely on empirical rules or heuristics to quickly find good schedules, while guided random search methods, e.g. genetic algorithm, usually involve extensive trial and error to search for the best schedules, and thus generally require more scheduling time. Heuristic-based methods can be further divided into three categories: list-based, clustering-based, and duplication-based [1]. Among them, list-based methods are the most commonly used ones because of their simplicity and wide applicability.

List-based workflow scheduling methods consist of two major parts: task prioritization and processor allocation. Different list-based methods differ in the mechanisms adopted in these two parts. HEFT [1] is the most famous list-based workflow scheduling algorithm, which prioritizes tasks according to their bottom ranks [1] and allocates processors based on the earliest-finish-time principle. Many later algorithms made improvement based on it, such as a lookahead variant of HEFT [10] and PEFT [9]. IPPTS [2] is a most recent state-of-the-art work on list-based workflow scheduling, which was shown to outperform many previous methods, including HEFT [1], its lookahead variant [10], and PEFT [9]. At the task prioritization stage, IPPTS adopts a Predict Cost Matrix (PCM) to calculate each task's rank, and takes into consideration the task's out-degree in the workflow structure. In the processor allocation phase, IPPTS uses a principle called looking ahead earliest finish time, which is a bi-directional downward and upward approach to allocating a task and its heaviest successor onto the most optimistic processor.

List-based workflow scheduling blends well with a Markov decision process, i.e. a series of decisions on selecting a task to schedule and a processor to allocate. Therefore, in this paper we try to apply DRL to list-based workflow scheduling and compare its performance to previous typical algorithms, including HEFT [1] and IPPTS [2].

3 Deep Reinforcement Learning for Workflow Scheduling

Reinforcement Learning (RL) [7] is a machine learning approach that learns how to make optimal decisions by continuously interacting with an environment as shown in Fig. 2. Its core principle is to establish a state-action-reward model. The state represents the current environment, the action refers to the operation taken in that state, and the reward is the feedback provided by the environment. The goal of reinforcement learning is to maximize the cumulative sum of rewards.

To apply RL to workflow scheduling, we turn the optimization problem of workflow scheduling, i.e. minimization of execution makespan, into the decision making problem of a Markov decision process following the list-based scheduling methodology, where at each step the best selection of both the task to schedule and the processor to allocate should be made. Specifically, we try to solve the workflow scheduling problem based on deep reinforcement learning (DRL), which incorporates deep learning into RL, i.e. representing the learned policy as a neural network. As shown in Fig. 3, the DRL-based workflow scheduling is an iterative process, which continuously generates new training data by scheduling a set of workflows based on current version of a neural network model, and then improves the neural network model based on the collected training data. The iterative process will continues until the scheduling quality could not be improved further or the training time limit is reached.

In our DRL-based workflow scheduling method, the RL part is governed by the Policy Gradient (PG) algorithm [7], whose basic principle is to adjust the policy based on feedback. Specifically, when receiving positive rewards, the probability of corresponding actions are increased. On the other hand, the probability of corresponding actions will be decreased when receiving negative rewards. In PG, it is necessary to define a policy function, which generates an action based on the current state. Typically, a neural network is used to represent the policy function. In our method, a CNN model is adopted, which accepts current state, including workflow structure and current partial schedule, as input. In this paper, we assume that each workflow contains 60 tasks at most, and is to be scheduled onto five processors for execution. Therefore, the neural network input, i.e. current state, is represented by three matrices of dimension 60 × 60, 60 × 5, and 60 × 2, respectively. The first matrix contains the inter-task communication costs of a workflow, while the second matrix represents the computation costs of each task on different processors in a heterogeneous computing environment. The third matrix records current partial schedule during the scheduling process.

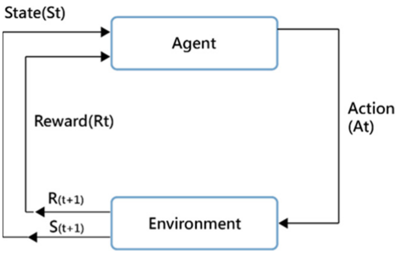


Fig. 2. Reinforcement learning process

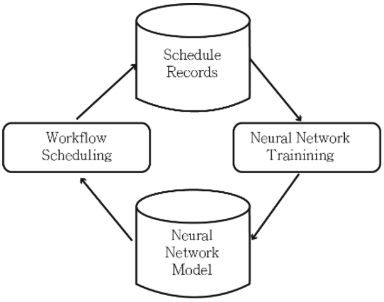


Fig. 3. DRL for workflow scheduling

The proposed DRL-based workflow scheduling method could be used in two different manners. In the first manner, the DRL-based method is used to continuously optimize a single workflow's schedule, similar to what the traditional guided random search methods [2] did. On the other hand, in the second manner, the DRL-based method is used to train a neural network model first based on a training set of workflows. Then, the neural network model is used as a policy function, i.e. returning how to select tasks and allocate processors, to schedule other workflows not in the training set in a list-based way. Both of these two manners will be evaluated in the following section.

4 Performance Evaluation

This section presents the performance evaluation of the proposed DRL-based workflow scheduling method conducted on the commonly used WorkflowSim [3] platform. The workflows used in the performance evaluation were randomly generated from WorkflowGenerator [4]. Each workflow might contain up to 60 tasks to be scheduled onto five processors. Since our DRL-based method follows the list-based scheduling methodology [1, 2], it was compared to the most famous list-based workflow scheduling method HEFT [1] and the state-of-the-art work IPPTS [2] in terms of execution makespan in the following performance evaluation. In the training process of our DRL-based method, the reward of each produced workflow schedule is assigned to the makespan improvement made by our method compared to IPPTS [2].

Tables 1 and 2 show the experimental results where the proposed DRL-based method was used to optimize a single workflow's makespan iteratively in speed-heterogeneous and speed-homogeneous environments, respectively. Heterogeneous environments were common for traditional distributed computing, while homogeneous environments could be created easily on modern cloud computing platforms. The proposed DRL-based method could produce significantly shorter workflow schedules than HEFT [1] and IPPTS [2] in both heterogeneous and homogeneous environments.

For Tables 3 and 4, a neural network model was trained with 13 randomly generated workflows using the proposed DRL-based method first. Then, the neural network model was used to schedule five new workflows in the list-based manner efficiently. Table 3 shows that the proposed DRL-based method achieves the shortest makespan among the three scheduling methods in four of the five workflows in a heterogeneous environment, indicating promising generalization capability. However, for a homogeneous environment, Table 4 shows that current experimental result is not quite good, where only one of the five workflows could achieve the best performance under the proposed DRL-based method. More training data might be helpful to improve the performance further since current training set contains only 13 workflows which might not be enough for achieving good performance.

Table 1. Optimizing a single workflow's makespan in a heterogeneous environment

Workflows	DRL makespan	IPPTS makespan	HEFT makespan
1	**8700.72**	13974.85	11436.44
2	**8877.99**	11747.95	17260.26

Table 2. Optimizing a single workflow's makespan in a homogeneous environment

Workflows	DRL makespan	IPPTS makespan	HEFT makespan
1	**12892.25**	15747.45	16385.63
2	**12624.57**	14123.23	16252.56

Table 3. DRL-based workflow scheduling in a heterogeneous environment

Workflows	DRL makespan	IPPTS makespan	HEFT makespan
1	**10800.56**	12301.70	10983.5
2	**10971.97**	14887.44	22393.05
3	11564.96	**11324.79**	14287.08
4	**11750.57**	12054.5	12565.47
5	**11204.16**	12149.25	11827.92

Table 4. DRL-based workflow scheduling in a homogeneous environment

Workflows	DRL makespan	IPPTS makespan	HEFT makespan
1	15099.79	**14434.20**	14619.49
2	**15133.12**	15816.7	16450.7
3	18869.33	**15415**	19474.79
4	18382.52	**18028.24**	18691.32
5	19128.88	**13774.70**	15790.1

5 Conclusions and Future Work

This paper presents a new workflow scheduling method based on deep reinforcement learning. The proposed method could be applied in two manners: (1) iteratively optimizing a single workflow's makespan from scratch, (2) training a neural network model to efficiently schedule new workflows never seen before following the list-based scheduling methodology without relying on human heuristics. Experimental results show significant performance improvement in both manners, compared to the most famous and state-of-the-art list-based workflow scheduling methods, i.e. HEFT [1] and IPPTS [2].

The preliminary results presented in this paper indicates a promising future research direction for workflow scheduling based on deep reinforcement learning. In addition, more research efforts are needed to investigate some issues and improve the scheduling quality further. For example, it will be interesting to find out why a heterogeneous environment could benefit more from the proposed DRL-based method as shown in Tables 3 and 4. Moreover, all of the adopted neural network model, reinforcement learning algorithm, state representation, and reward assignment method, involved in the proposed DRL-based method, deserve more research work to improve their designs for further performance improvement in workflow scheduling.

References

1. Topcuoglu, H., Hariri and Min-You Wu, S.: Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Trans.Parall. Distrib. Syst.**13**(3), 260–274 (2002)
2. Djigal, H., Feng, J., Lu, J., Ge, J.: IPPTS: an efficient algorithm for scientific workflow scheduling in heterogeneous computing systems. IEEE Trans. Parallel Distrib. Syst. **32**(5), 1057–1071 (2021)
3. Chen, W., Deelman, E.: WorkflowSim: a toolkit for simulating scientific workflows in distributed environments. In: IEEE 8th International Conference on E-Science, pp. 1–8 (2012)
4. Juve, G., et al.: Workflow Generator. https://github.com/pegasus-isi/WorkflowGenerator
5. Sinnen, O.: Task Scheduling for Parallel Systems, John Wiley (2007)
6. Silver, D., Hubert, T., Schrittwieser, J.: Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv:1712.01815v1 [cs.AI] (2017)
7. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, MIT Press, (2018)
8. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning, (MIT Press, 2016)

9. Arabnejad, H., Barbosa J.G.: List scheduling algorithm for heterogeneous systems by an optimistic cost table. IEEE Trans. Parall. Distrib. Syst. **25**(3), 682–694 (2013)
10. Bittencourt, L.F., Sakellariou, R., Madeira, E.R.M.: DAG scheduling using a lookahead variant of the heterogeneous earliest finish time algorithm. In: 18th Euromicro Conference on Parallel, Distributed and Network-based Processing, pp. 27–34 (2010)

Federated Learning for Skin Cancer Classification

Zhe-Kai Xu(✉) and Yen-Wen Lin

Department of Computer Science, National Taichung University of Education, Taichung, Taiwan
applewish9@gmail.com

Abstract. Cancer is one of the deadliest diseases globally. Early detection is crucial for effective treatment. This paper proposes a new method that adopts federated learning for skin cancer classification. Experimental results demonstrate the potential of the proposed method for enhancing the accuracy of skin cancer classification. The proposed method can achieve 98% accuracy on the HAM10000 [1] dataset.

Keywords: Federated Learning · Skin Cancer Classification · Data Augmentation

1 Introduction

1.1 Background

The traditional method [2] for detecting skin cancer involves visual examination of skin lesions by dermatologists or experts, with further testing if necessary. However, this process is time-consuming and expensive. With the latest advancements in machine learning and computer vision, researchers have started exploring methods to automatically detect skin cancer using deep learning [3] techniques.

1.2 Motivations

By using deep learning techniques, the accuracy of skin cancer classification can be improved. However, the widespread adoption and application of artificial intelligence technology in recent years have posed significant challenges to personal privacy. Artificial intelligence systems require a large amount of data for learning and making predictions, which often includes sensitive information such as personal identification and health status. If related data is not protected, it may be illegally accessed or leaked, resulting in the misuse and infringement of personal information.

Federated learning [4] is a machine learning approach that allows multiple devices to train cooperatively, each device trains a model on local data, shares the parameters of the model, and periodically sends model updates to a central server for aggregation, without sharing their private data (see Fig. 1). The process of federated learning can be simply divided into three steps: First, clients upload parameters. Second, the

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 62–67, 2024.
https://doi.org/10.1007/978-981-99-8211-0_7

server aggregates parameters. Third, the server returns parameters. The approach enables privacy-preserving machine learning, making it ideal for medical image analysis. Table 1 is a brief comparison of federated learning and traditional machine learning.

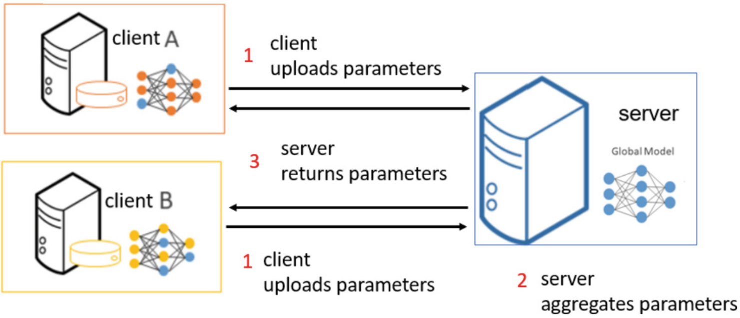

client A
1 client uploads parameters
server
Global Model
3 server returns parameters
client B
1 client uploads parameters
2 server aggregates parameters

Fig. 1. The process of federated learning.

Table 1. Comparing Federated Learning and Machine Learning

Compare Items	Federated Learning	Machine Learning
Items uploaded to the server	parameter	data
Privacy	yes	no

1.3 Goal

This study aims to use federated learning for skin cancer classification and optimize the performance.

2 Research Method

2.1 Problem Description

In this paper, the performance of using federated learning for training skin cancer classification model is studied. Besides, to improve the accuracy of the trained model, the methods of solving related problems, including overfitting and data imbalance, are figured out.

2.2 Dataset

The HAM10000 dataset [1] is a publicly available dataset widely used for classification and diagnosis of skin diseases. The dataset collects images from 10,015 patients, including different types of skin lesions [1] (i.e. seven classes in Table 2).

For testing the performance of federated learning, this dataset is divided into two clients, client1 and client2. They have almost the same numbers of the images of each class. The numbers of each class are listed in Table 2.

Furthermore, as shown in Table 2, the number of images of each class is not equal. In this study, for investigating the effects of data balance, the number of images of client1 and client2 are expanded to 4000 with data augmentation.

Table 2. Used Dataset [1]

skin cancer classification	Class	ISIC2018	Client1	Client2	Augmented Client1/Client2
actinic keratosis	1	327	163	164	4000
basal cell carcinoma	2	514	257	257	4000
dermatofibroma	3	115	57	58	4000
melanoma	4	1113	556	557	4000
nevus	5	6705	3352	3353	4000
pigmented benign keratosis	6	1099	549	550	4000
vascular lesion	7	139	71	71	4000

2.3 Model

A Convolutional Neural Network (CNN) [5] is a deep learning neural network architecture primarily used for image recognition, classification, and other related tasks. For image processing tasks, CNN is a suitable model. It can effectively capture the local features in the image, and has a good perception of image details such as edges, textures, and shapes.

CNN typically consists of multiple modules, including convolutional layers, pooling layers, and fully connected layers [5]. The proposed design in this study is illustrated in Fig. 2.

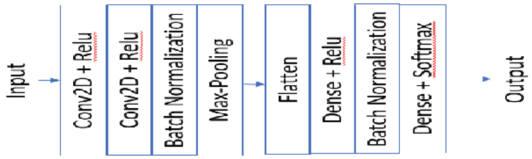


Fig. 2. Basic convolutional neural network

3 Experiments and Results Discussion

3.1 Experimental Design

To study the performance of federated learning, data augmentation [6], and data balancing [7]. Related experiments are carried out as follows. The settings of the following experiments are listed in Table 3.

Table 3. Parameter Settings

Name	Value
Learning rate	0.001
Batch size	32
Optimizer	Adam
Loss function	SparseCategoricalCrossentropy

3.2 The Effects of Federated Learning

Related experiments are implemented here to check the effects of adopting federated learning in skin cancer classification. As shown in Fig. 3, without compromising the privacy, the accuracy of federated learning is almost equal to that of machine learning. It implies that federated learning is viable for skin cancer classification.

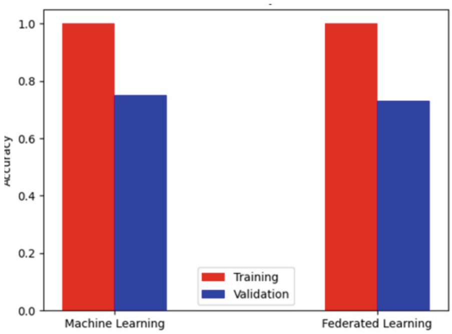


Fig. 3. Effect of federated learning

3.3 The Effects of Data Augmentation

As displayed in Fig. 3, in either machine learning or federated learning, the accuracy of the validation set is obviously lower than that of the training set; that possibly results from over-fitting. In the proposed system, data augmentation is used for reducing the over-fitting. In Fig. 4, the difference between the accuracy of the training set and the validation set is reduced by employing data augmentation.

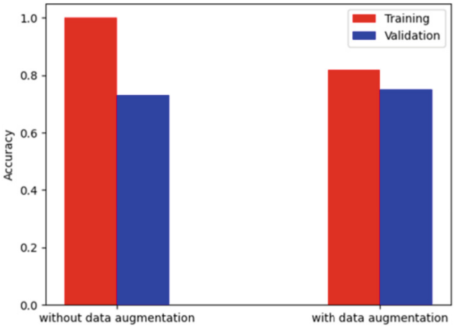


Fig. 4. Effect of data augmentation

3.4 The Effects of Data Balancing

As offered in Table 2, the number of images of various classes is not equal; that might yield bad effects on the accuracy of the trained models. Therefore, data balancing is adopted in the proposed system to remedy the class imbalance problem. As shown in Fig. 5, the accuracy of the model trained with balanced data is higher than that of the one trained with imbalanced data. It suggests that data balancing can improve the accuracy of model.

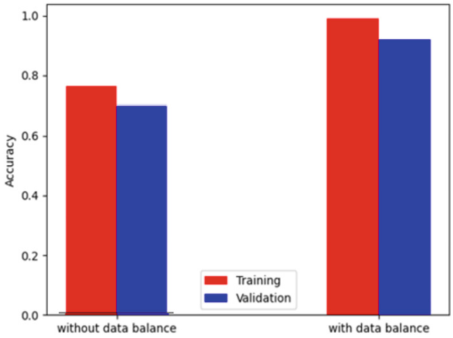


Fig. 5. Effect of data balancing

4 Conclusion

In this work, the performance of using federated learning in skin cancer classification is investigated. Besides, data augmentation and data balancing are adopted for further promoting the accuracy of the trained models. Experiment results show that the accuracy of the proposed methods is up to 98%.

References

1. ISIC Dataset Page. https://challenge.isic-archive.com/data/#2018
2. Simoes, M.F., Sousa, J.S., Pais, A.C.: Skin cancer and new treatment perspectives: A review. Cancer Lett. **357**(1), 8–42 (2015)
3. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521** (7553), 436–444 (2015)
4. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: Challenges, methods, and future directions. IEEE Signal Process. Mag. **37**(3), 50–60 (2020)
5. O'Shea, K., Nash, R.: An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458 (2015)
6. Van Dyk, D.A., Meng, X.L.: The art of data augmentation. J. Comput. Graph. Stat. **10**(1), 1–50 (2001)
7. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor. Newslett. **6**(1), 20–29 (2004)

A Task Offloading and Content Caching Strategy for the Internet of Vehicles in Cloud-Edge Environment

Yaping Wang, Junye Qiao, Zekun Hu, and Pengwei Wang(✉)

School of Computer Science and Technology, Donghua University, Shanghai 201620, China
wangpengwei@dhu.edu.cn

Abstract. A wide range of emerging in-vehicle applications can make the travel experience better for users. As the amount of vehicles on the road increases, so does the number of computational tasks that need to be processed, however, different vehicle users may request the same content, resulting in wasted resources. Therefore, IoV requires better compute offloading and content caching strategies to improve performance with respect to time latency and energy consumption. This paper proposes a joint task offloading and content caching optimization method based on forecasting traffic stream, called TOCC. First, temporal and spatial correlations are extracted from the preprocessed dataset using FOST and integrated to predict the traffic stream to obtain the number of tasks in the region at the next moment. To obtain a suitable joint optimization strategy for task offloading and content caching, the multi-objective problem of minimizing delay and energy consumption is decomposed into multiple single-objective problems using an improved MOEA/D via the Tchebycheff weight aggregation method, and a set of Pareto-optimal solutions is obtained. Finally, experimental results show the effectiveness of TOCC's task offloading and task caching strategies and that TOCC outperforms than other methods with respect to time delay and energy consumption.

Keywords: Task Offloading · Content Caching · IoV · Traffic Stream Prediction · Cloud-Edge Environment

1 Introduction

With the rapid advancement of socioeconomic factors and 5G technology, urban areas are witnessing a significant surge in the number of vehicles. This exponential growth has transformed the transportation landscape, primarily owing to the emergence of Internet of Vehicles (IoV) technology, enriching users with an enhanced driving experience. However, in the context of high-speed vehicle operations, seamless access to a vast array of internet-based content becomes paramount [1]. For time-critical tasks, failing to meet completion deadlines can lead to severe consequences. One of the challenges faced is that vehicles often possess limited computing and storage capacities, potentially leading to delays or incomplete task processing when handled locally. To cater to the low

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 68–80, 2024.
https://doi.org/10.1007/978-981-99-8211-0_8

latency demands of IoV, Mobile Edge Computing (MEC) has emerged as a promising solution. By offloading tasks to resource-intensive edge servers, the driving experience can be significantly enhanced.

The efficiency of task offloading and content caching can be significantly enhanced through the utilization of traffic stream forecasting, which offers valuable insights into the task volume in different areas. In the context of Internet of Vehicles (IoV), the escalating computational demands of vehicle terminals necessitate effective computational offloading strategies. To address this challenge, this paper highlights the development of task-level offloading methods [2]. We present a novel optimization approach named TOCC, which integrates traffic stream prediction with task offloading and content caching.

Specifically, this work makes the following main contributions:

- Utilizing Microsoft's FOST [3] tool, the real dataset BikeNYC undergoes preprocessing to adapt the data structure to the model's operational requirements. The model extracts temporal and spatial correlations from the preprocessed dataset and integrates them to predict future traffic stream.
- We use the predicted traffic stream and an enhanced multi-objective evolutionary algorithm (MOEA/D) [4] to decompose the multitask offloading and content caching problem into individual optimization problems. This decomposition helps us to obtain a set of Pareto optimal solutions, which is achieved through the Tchebycheff weight aggregation approach. As a result, our approach successfully reduces both execution time and energy consumption.
- Finally, the performance of the TOCC algorithm is assessed through an evaluation using a comprehensive simulation dataset.

2 Related Work

In the high-speed mobile environment of IoV, efficient task execution plays a crucial role in minimizing time delays. However, the limited computing and storage capacity of automotive systems often necessitate offloading resource-intensive tasks. These tasks are strategically shifted to edge or cloud servers, enabling efficient execution and ensuring prompt fulfillment of user requests. Automotive applications frequently involve repeated content requests, making caching popular content on edge servers highly effective in reducing latency and energy consumption for subsequent accesses in IoV. Nevertheless, improper task offloading and inaccurate content caching can lead to increased energy consumption and latency, mainly due to network congestion and server queuing. To address these challenges, the joint optimization of task offloads and content caching is essential. By harmonizing these processes, we can effectively minimize time latency and energy consumption, enhancing the overall performance and user experience in IoV.

In recent years, significant research efforts have been dedicated to devising effective task offloading schemes in cloud-edge environments. Zhao et al. [5] proposed an energy-efficient offloading scheme that utilizes a three-layer architecture: cloud, fog, and user devices. Their approach compares the energy consumption of offloading tasks to the cloud and fog, aiming to minimize energy usage. While their algorithm demonstrates low energy consumption for single tasks, further research is needed to investigate its effectiveness for multiple tasks. Chen et al. [6] presented a UT-UDN system model, showcasing a 20% reduction in time delay and a 30% decrease in energy costs based on simulation results. Additionally, several studies have also employed heuristic algorithms to tackle task offloading problems. Xu et al. [7], for instance, utilize enumeration algorithm and branch-and-bound algorithm to address these challenges.

The in-vehicle edge environment synergizes computing resources from both the vehicle and the edge, creating a powerful platform for delivering computing services. Yang et al. [8] propose a location-based offloading scheme that strikingly balances task completion latency with communication and computational resources, leading to a notable reduction in system costs. To address challenges related to task latency and constraints in RSU (Roadside Unit) server resources, Zhang et al. [9] introduce a contract-based computing resource allocation scheme within a cloud environment. This innovative approach aims to maximize the benefits of Mobile Edge Computing (MEC) service providers, enhancing vehicle utility and resource utilization while adhering to latency limitations. Dai et al. [10] present a novel approach by decoupling the joint load balancing and offloading problem into two sub-problems, formulated as a mixed-integer nonlinear programming problem. The primary objective is to maximize the system utility under latency constraints. In the realm of 5G networks, Wan et al. [11] introduce an edge computing framework for offloading utilizing multi-objective optimization and evolutionary algorithms. This framework efficiently explores the synergy between offloading and resource allocation within MEC and cloud computing, resulting in optimized task duration and server costs. In the pursuit of comprehensive optimization, Zhao et al. [12] propose a collaborative approach that minimizes task duration and server costs through joint optimization of offloading and resource allocation in the MEC and cloud computing domains.

These studies primarily delve into the task offloading problem within edge computing, cloud computing, and cloud-edge integration environments. However, when considering the specific context of IoV, these approaches often overlook the influence of future traffic streams, resulting in a loss of offloading accuracy. Addressing this gap, Fang et al. [13] proposed the ST-ResNet network for traffic prediction, complemented by the NSGA-III algorithm for multi-objective optimization. Their method showcased superior performance in terms of latency and energy consumption, outperforming other existing methods.

In light of these findings, our research aims to tackle the joint optimization problem of task offloading and edge content caching. We achieve this by leveraging an improved multi-objective evolutionary algorithm based on traffic prediction. The overarching goal is to minimize transmission and computation latency while concurrently reducing energy consumption.

3 System Model and Problem Formulation

3.1 System Framework

Figure 1 illustrates a robust three-tier cloud-edge-vehicular network framework designed to cater to diverse tasks across different regions. The framework comprises three layers: the vehicle terminal layer, the Mobile Edge Computing (MEC) layer, and the cloud computing layer. In the cloud computing layer, cloud servers cover the entire area, providing extensive coverage. The MEC layer consists of edge servers along the roadside, covering an area. The vehicle terminal layer comprises vehicles traversing the road, communicating with both the MEC layer and the cloud computing layer via wireless channels. During operation, the vehicle terminal layer undertakes one or more computing tasks with varying probabilities, taking advantage of time gaps to execute the tasks efficiently. Three possible destinations for task offloading are available: local processing (i.e., handling tasks on the computing device within the vehicle), processing by the edge server, or processing by the cloud server. The edge server is equipped with the capability to cache popular content, further enhancing task offloading efficiency.

A region's traffic stream is the count of vehicles passing through it in a given time slot, such as one minute. Let Tr denote the set of vehicle trajectories at the t^{th} time slot. The vehicle traffic stream in region i during the t^{th} time slot can be determined by $TS_i(t) \triangleq \sum_{tr \in Tr} \mathbb{I}(tr)$, where $tr =< s_1, s_2, \ldots, s_{|tr|} >$ is an ordered set representing the discrete trajectory of a user multimedia request over time; s_i represents the geographical position of the user's multimedia request at certain times; and $\mathbb{I}(tr)$ is a binary variable that equals 1 if s_k in region i exists in tr. Otherwise, that equals 0. The problem of predicting the stream of vehicles can be defined as follows: given the historical traffic stream $\{TS_i(t)|1 \le i \le I,\ 1 \le t \le T\}$, the goal is to predict the future traffic stream $\{TS_i(T+1)|1 \le i \le I\}$.

In a partitioned region, we assume $N \ge Q$ where N is the number of vehicles and Q is the number of tasks. This assumption accounts for prevalent tasks that are repeatedly requested and executed. Each vehicle can perform only one task at a time, and multiple vehicles may request the same task based on their preferences. To simplify notation, we define $q_{i,n}$ as the task q generated by the n^{th} vehicle in region i.

To characterize different computational tasks, a triple is employed as the computational task model: $q_{i,n} = \left(c_q^{i,n}, d_q^{i,n}, DDL_q^{i,n}\right)$. Task $q_{i,n}$ can be partially offloaded to either the MEC server or the cloud computing server for processing. The parameter $c_q^{i,n}$ is the amount of CPU cycles required to accomplish task $q_{i,n}$. The parameter $d_q^{i,n}$ represents the input data size required for processing task $q_{i,n}$, while $DDL_q^{i,n}$ signifies the maximum deadline for completing the task. It is assumed that the value of $c_q^{i,n}$ remains constant regardless of whether task $q_{i,n}$ is processed locally, offloaded to the MEC server, or executed on the cloud computing server. Furthermore, MEC servers within the region are assumed to have limited computation capacity, denoted as c_m, and a cache size of s_m.

3.2 Execution Time and Energy Consumption Model

For the task offloading problem, we divide the computational task into multiple parts and define the offloading decision variable $\alpha_n^i = (\alpha_{i,n}^l, \alpha_{i,n}^m, \alpha_{i,n}^c)$, where $\alpha_{i,n}^l, \alpha_{i,n}^m, \alpha_{i,n}^c \in [0, 1]$, denotes the percentage of task $q_{i,n}$ offloaded to the vehicle local, MEC server and cloud server, respectively. The constraint $\alpha_{i,n}^l + \alpha_{i,n}^m + \alpha_{i,n}^c = 1$ ensures that the entire task is accounted for. For example, if $\alpha_{i,n}^l = 1$, the task is executed l exclusively within

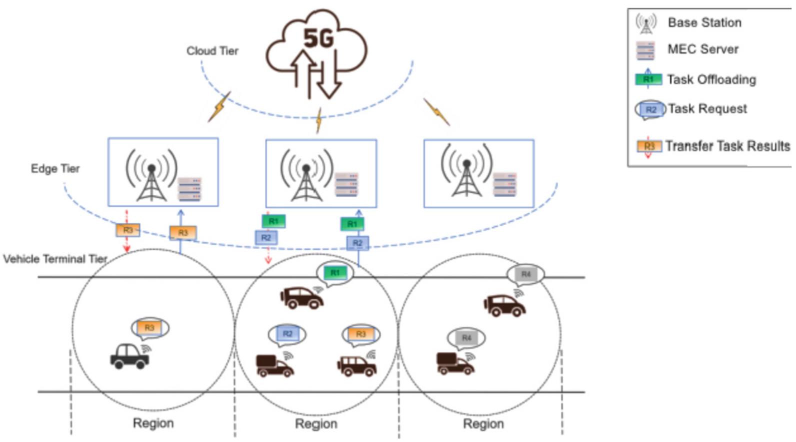


Fig. 1. Three-tier cloud-edge-vehicle network framework with different tasks.

the vehicle, where $\alpha_{i,n}^{l} = 0$ indicates complete offloading to the MEC or cloud server. The overall offloading decision policy is denoted as $A = [\alpha_1^1, \alpha_2^1, \cdots, \alpha_n^i, \cdots, \alpha_N^I]$.

Execution Time and Energy Consumption of Local Task Computation. If the task $q_{i,n}$ is selected for local processing, then $TL_q^{i,n}$ is defined as the local execution time. Due to the difference in its own computational power brought by vehicle heterogeneity, the local execution time delay of task $q_{i,n}$ is

$$TL_q^{i,n}(t) = \alpha_{i,n}^{l} \cdot c_q^{i,n} / fl_n^i \tag{1}$$

Energy consumption is calculated as

$$EL_q^{i,n}(t) = \alpha_{i,n}^{l} \cdot c_q^{i,n} \cdot \left(fl_n^i\right)^2 \cdot \varsigma \tag{2}$$

where fl_n^i is the computational power of the n^{th} vehicle in region i. $\left(fl_n^i\right)^2 \cdot \varsigma$ is the energy consumption per CPU cycle.

Execution Time and Energy Consumption of Edge Task Computing. When task $q_{i,n}$ is offloaded to the MEC server, the process involves the following steps: the n^{th} vehicle uploads the task's input data to the MEC server via the BS/RSU. The MEC server allocates computational resources for task processing and returns the result to the vehicle. The time of tasks offloaded to MEC server m can be as described below:

$$TM_q^{i,n}(t) = \alpha_{i,n}^{m} \cdot \left(d_q^{i,n} / v_n^i(t) + c_q^{i,n} / fm_i + o_q^{i,n} / v_m\right) \tag{3}$$

This means that the energy consumed is

$$EM_q^{i,n}(t) = \alpha_{i,n}^{m} \cdot \left(c_q^{i,n} \cdot (fm_i)^2 \cdot \varsigma + o_q^{i,n} / v_m \cdot \delta^m\right) \tag{4}$$

where fm_i is computing capability of edge servers in region i, $v_n^i(t)$ is the data offload rate from the n^{th} vehicle to the mobile edge server in region i at time slot t, v_m is backhaul transmission rate of edge server m, δ^m is offload capability for edge server m.

Execution Time and Energy Consumption of Cloud Server Computing. The cloud server c is situated at a greater distance from the task source compared to the edge server, resulting in potential latency. Hence, we opt to incorporate the cloud computing model for task processing only under specific conditions. These conditions include cases where the task demands extensive computational resources that surpass the processing capacity of the edge server or in scenarios where the edge server is already operating at its maximum capacity due to concurrent multitasking. Hence, the time $TC_q^{i,n}$ of the task offloaded to cloud server c is defined as

$$TC_q^{i,n}(t) = \alpha_{i,n}^c \cdot \left(d_q^{i,n} / v_{n_c}^i(t) + c_q^{i,n} / fc_c + o_q^{i,n} / v_c \right) \tag{5}$$

And the energy consumption is:

$$EC_q^{i,n}(t) = \alpha_{i,n}^c \cdot \left(c_q^{i,n} \cdot (fc_c)^2 \cdot \varsigma + o_q^{i,n} / v_c \cdot \delta^c \right) \tag{6}$$

Based on the analysis, the execution time for the task on the n^{th} vehicle in region i can be calculated according to the equation below:

$$TT_q^{i,n}(t) = max\left(TL_q^{i,n}, TM_q^{i,n}, TC_q^{i,n} \right) \tag{7}$$

And the energy consumption is:

$$ET_q^{i,n}(t) = EL_q^{i,n} + EM_q^{i,n} + EC_q^{i,n} \tag{8}$$

3.3 Edge Data Caching Model

Task caching refers to the storage of completed tasks and their associated data on the edge cloud. This paper formulates the content caching problem using a binary cache decision variable $s_m^{i,n} \in \{0, 1\}$. Therefore, the task caching policy is expressed as: $S = \left\{ s_1^{1,1}, s_1^{1,2}, \cdots, s_m^{i,n} \right\}$.

Considering the joint processing of task offloading and content caching to vehicle local, edge and cloud servers, the total execution latency of task $q_{i,n}$ generated by the n^{th} vehicle in region i is

$$T_q^{i,n}(t) = s_m^{i,n} o_q^{i,n} / v_m + \left(1 - s_m^{i,n} \right) TT_q^{i,n}(t) \tag{9}$$

And the total consumption of energy is:

$$E_q^{i,n}(t) = s_m^{i,n} EM_q^{i,n}(t) + \left(1 - s_m^{i,n} \right) ET_q^{i,n}(t) \tag{10}$$

Therefore, the average execution time of vehicles in the region i is

$$\overline{T}^i(t) = 1 / TS_i(t) \sum\nolimits_{i=1}^{TS_i(t)} T_q^{i,n}(t) \tag{11}$$

The energy consumption of vehicles in the region i is

$$E^i(t) = \sum\nolimits_{i=1}^{TS_i(t)} E_q^{i,n}(t) \tag{12}$$

The combined size of the data contained in the regional edge servers is:

$$S_i = s_m^{i,n} \sum\nolimits_{i=1}^{TS_i(t)} O_q^{i,n} + \left(1 - s_m^{i,n}\right) \sum_{i=1}^{TS_i(t)} \alpha_{i,n}^m d_q^{i,n} \tag{13}$$

3.4 Problem Definition

In this paper, the aim is to minimize the average time taken to execute and total energy consumption within each region. This problem is formulated with the consideration of maximum latency and computing power constraints, and can be described as follows:

$$min\overline{T}^i(t), minE^i(t)\,\forall i = 1, 2, ..., I \tag{14}$$

$$\begin{aligned}
&St:\\
&C1: \alpha_{i,n}^l,\ \alpha_{i,n}^m,\ \alpha_{i,n}^c \in [0,\,1]\alpha_{i,n}^l + \alpha_{i,n}^m + \alpha_{i,n}^c = 1\\
&C2: s_m^{i,n} \in \{0,\,1\}\\
&C3: T_q^{i,n}(t) \le DDL_q\\
&C4: S_i \le s_m\\
&C5: \sum\nolimits_{n=1}^{N} \alpha_{i,n}^m fm_i \le c_m
\end{aligned}$$

4 Joint Optimization for Offloading and Content Caching with Traffic Stream Prediction

In this section, FOST is first used to solve the problem of predicting traffic stream. The BikeNYC dataset is preprocessed to provide a dataset format adapted to the model, after which FOST extracts temporal and spatial correlations on the dataset and integrates them to predict traffic stream, thus improving the accuracy of predicted traffic stream. Then, MOEA/D is used to search for the optimal solution for the joint optimization of task offloading and content caching to determine whether to cache the vehicle-generated tasks to the edge or offload them to each platform.

4.1 FOST Enabled Traffic Stream Prediction

Data Preprocessing Requirements. Before using the data model for prediction, it is necessary to provide a data set format adapted to the model, including *train* and *graph*, where *train* involves time and target values, and *graph* involves the spatial relationship weights between nodes. To reflect the spatial relationship weights of multiple regions, we define w_{ij} represents the degree of influence of the region i on the region j in the space. If $w_{ij} = 1$, region i and region j are adjacent. Otherwise, $w_{ij} = 0$.

Spatial-Temporal Correlation Extraction. Based on the pre-processed dataset, we use MLP and RNN to extract temporal correlations in the *train* dataset and use GNN to extract spatial correlations of the nodes in the *graph* dataset. In vehicular traffic prediction, we have to consider not only the temporal variation but also the spatial interactions. Because the traffic results of one region will be influenced by other regions, especially the neighboring regions, the spatial correlation cannot be ignored. Therefore, it is necessary to extract the temporal and spatial correlations in traffic stream data. First, the time series data of the recent time gaps, including the total traffic stream in and out of this region for each time slot, need to be input in the underlying deep temporal neural network module. After that, the temporal module of the model will first learn the features in the historical data and represent them as a set of vectors in the hidden space. Next, it is necessary to further implement spatial information aggregation by superimposing information on the timing patterns of adjacent spaces.

4.2 MOEA/D-Based on Joint Optimization

The IoV environment always consists of multiple regions, so the joint optimization problem of task offloading and content caching with the objective of minimizing the average execution time and total energy consumption in each region under the constraints of maximum latency and computational power is a multi-objective optimization problem. The multi-objective optimization problem is converted to a multi-group single-objective optimization problem using the Tchebycheff approach. A collaborative approach combined with a population evolution strategy is used to optimize these subproblems simultaneously using the neighborhood relationship between the subproblems.

Tchebycheff Weight Aggregation Approach. Among the various decomposition methods of MOEA/D, we choose to use Tchebycheff approach because it can handle relatively more complex problems with high computational efficiency. Then, the decomposition cost form of the sub-problem optimization problem of computing the content caching or offloading of tasks in a region is minimize $\mathrm{g^{tche}(x|w,\ z^*)} = \max_{1\le i\le m}\{\lambda_i|f_i(x) - w_i^*\}$, where, $z^* = \{z_1^*, z_2^*\}(i \in \{1, 2, ..., P\})$ is the optimal value of $\overline{T}^i(t)$ and$E^i(t)$. w_i represents the weight of the i^{th} objective function for each$x$, and P represents the population size defined by MOEA/D. In this paper, letf_1 be the energy consumption of the area, and f_2 be the average time delay of the area. Decompose the multi-objective optimization problem into m sub problems according to the Chebyshev ray uniform expansion, assign the objective weight of each sub problem, and obtain the sub problem weight matrix $w = [w_1, w_2, ..., w_i, ..., w_m]$, where $w_i = \left(w_i^1, w_i^2\right)$ represents the weight vector of the i^{th} sub problem. Let $w_i^{1'} = i \times \frac{1}{m+1}$, $w_i^{2'} = 1-, w_i^{1'}$; $w_i^1 = (1/w_i^{1'})/\left(\frac{1}{w_i^{1'}} + \frac{1}{w_i^{2'}}\right)$, $w_i^2 = 1 - w_i^1$

Chromosome Coding and Genetic Operators. In this paper, the chromosome encoding for the population evolution strategy uses RI, where each bit on the chromosome represents the true value of the decision variable. For selection, ETour was used. For recombination, we use SBX to simulate single point crossover based on binary strings.

For mutation, we generated a new population chromosome matrix by polynomially varying each decision variable in the real integer-encoded population chromosome matrix according to the mutation rate.

4.3 Description of the Algorithm

```
ALGORITHM TOCC: Task Offloading and Content Caching Strategy based on MOEA/D
Require: historical traffic stream $\{TS_i(t)|1 \le i \le I、1 \le t \le T\}$，task set $Q$, maximum
number of iterations $G$, population size $P$, reference point $z^*$
Ensure: Offloading policy $A$, content caching policy $S$, time-delay optimal solution objective
$\bar{T}^i(t)$ , energy consumption optimal solution objective $E^i(t)$
Traffic Stream Forecasting
 1:   Pre-processing data formats, training train and graph data, adapting models
 2:   MLP RNN to extract temporal correlation, GNN to extract spatial correlation for
      modeling training, and the model set Fusion
 3:   Output $\{TS_i(t+1)|1 \le i \le I\}$
Joint Optimization for Task Offloading and Edge Content Caching Optimization
 4:   Initialize $P_0$
 5:   Calculate $\bar{T}^i(t)$(11), Calculate $E^i(t)$(12)
 6:   Initialize $z^* = \{z_1^*, z_2^*\}^T$, $V$
 7:   Record the $T$ vector in $B(i)$ that is closest to any vector
 8:   $for\ g\ =\ 1\ to\ G$ do
 9:     $for\ i\ =\ 1\ to\ N$ do
10:       Randomly choose $f$ and $h$ at $B(i)$
11:       The genetic operators $x^f$ and $x^h$ evolve to produce the solution set $y$
12:       if $z_1^* < \bar{T}^i(t)$ then
13:         $z_1^* = \bar{T}^i(t)$
14:       end if
15:       if $z_2^* < E^i(t)$ then
16:         $z_2^* = E^i(t)$
17:       end if
18:       for Each $j \in B(i)(j = 1,2)$ do
19:         Update the neighborhood solution set $B(i)$
20:     end for
21:   end for
22:   return $A$, $S$
```

The algorithm provides an overview of the optimization process. Steps 1–3 involve the pre-processing traffic stream data and initializing trainable parameters for model training. Steps 4–22 focus on joint optimization of task offloading and content caching using MOEA/D. First, G is denoted as the maximum number of iterations, and T is denoted as the number of neighbours of each weight vector w_i. To initialize the population P_0, p individuals are randomly generated within the decision space. Each individual in population P_0 computes the values of $\overline{T}^i(t)$ and $E^i(t)$ for the objective function defined in Eq. (15). Initialize the reference point $z^* = \left\{z_1^*, z_2^*\right\}^T$ and the set of uniformly distributed individual weight vectors $V = \{v_1, v_2, ..., v_T\}$. Next, the Euclidean distance between every pair of weight vectors is calculated to identify the T nearest weight vectors for each weight vector, denoted as $B(i) = \{i_1, i_2, ..., i_T\}$. After that, the evolutionary process is carried out with updates. Subsequently, the minimum delay and minimum energy consumption are updated. If the termination condition is met, the computation is concluded, and the results are generated. Finally, the corresponding set of offloading policies A and caching policies S are outputted.

5 Experimental Evaluation

This section begins with a description of the experimental setup, after which the performance of TOCC is evaluated.

5.1 Experimental Setup

The performance of the TOCC was evaluated and compared to three baselines. These baselines are briefly described below.

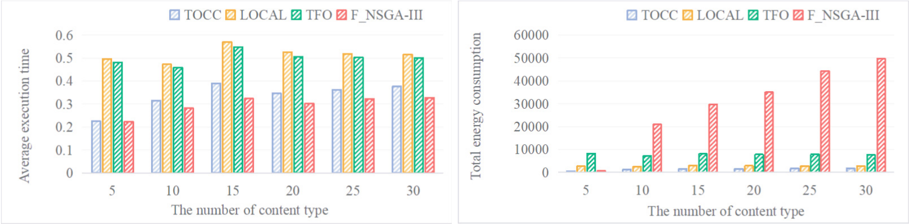
- LOCAL: No utilization of edge content cache and MEC, i.e., tasks are processed locally on the vehicle upon generation.
- TFO: No edge content cache is utilized. The offload destination for the task is determined by selecting the destination with the lowest execution time.
- F_NSGA-III: An evolutionary algorithm based on FOST traffic stream prediction for solving multi-objective problems.

In our experiments, we selected and evaluated 128 city center areas with a maximum traffic stream of 30 in three dimensions. We evaluate the effectiveness of the method in various environmental situations, including the number of content types ranging from 5 to 30, the size of the input data varying from 1 to 2 times, and the number of CPU cycles varying from 1 to 6 times. The implementation of the methods was done using Python3 on Ubuntu 18.04 LTS.

5.2 Comparative Experiments

Comparison with Changing of Content Types. As the content type increases, the average execution time of TOCC is better than that of LOCAL and TFO, as shown in Fig. 2(a). The cache of TOCC is sensitive to the content type and is greatly affected by it, which causes the average execution time to continuously increase. In contrast, TFO and LOCAL are less affected by content type changes and the average execution time fluctuates within a certain range of higher levels. The average execution time of TOCC is

46% better than LOCAL and 44% better than TFO. This highlights the effectiveness of MEC and edge content caching in reducing IoV time latency. TOCC is slightly worse than F_NSGA-III, 12% lower. Figure 2(b) shows that with the increase of content type, TOCC has the lowest total energy consumption compared with LOCAL, TFO and F_NSGA-III. In terms of total energy consumption, TOCC outperforms LOCAL by 60%, TFO by 85%, and F_NSGA-III by 79%.

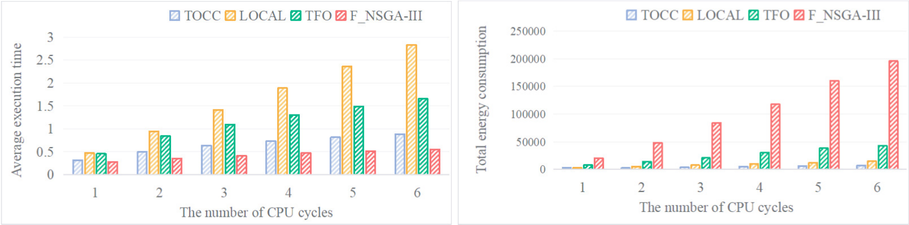


(a) Average execution time

(b) Total energy consumption

Fig. 2. Comparison of average execution time and total energy consumption with changing of content types

Comparison with Changing of CPU Cycles. Figure 3(a) clearly illustrates the significant advantage of TOCC over LOCAL and TFO when the number of CPU cycles per task increases by a factor, and TOCC slightly loses to F_NSGA-III. In terms of average execution time, TOCC achieves 54% improvement over LOCAL, 53% improvement over TFO, and 46% reduction over F_NSGA-III. In Fig. 3(b), as the number of CPU cycles per task increases, TOCC achieves the lowest energy consumption compared to LOCAL, TFO, and F_NSGA-III. In terms of total energy consumption, TOCC is 78% higher than LOCAL, 93% higher than TFO, and 94% higher than F_NSGA-III. This is because the strategy in TOCC can better balance the average execution time and energy consumption, and does not make one party dominant.

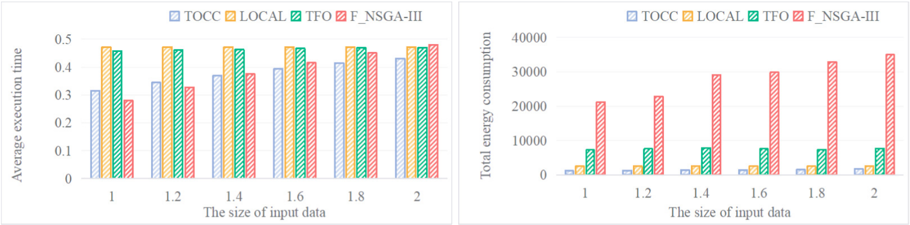


(a) Average execution time

(b) Total energy consumption

Fig. 3. Comparison of average execution time and total energy consumption with changing of CPU cycles

Comparison with Changing of Input Data Size. Figure 4(a) clearly depicts that TOCC outperforms LOCAL, TFO and F_NSGA-III in achieving the lowest average execution time when the input data size of a single task is increased by 20%, showing a flat upward trend. This is because the input data affects the amount of tasks in the edge cache and the offloading of computational tasks. TOCC improved by 29% over LOCAL, 28% over TFO and 1.2% over F_NSGA-III. In Fig. 4(b), the total energy consumption of TOCC is also the lowest as the number of tasks increases. In terms of total energy consumption, TOCC exceeds LOCAL by 78%, is 93% higher than TFO, and 94% higher than F_NSGA-III.



(a) Average execution time

(b) Total energy consumption

Fig. 4. Comparison of average execution time and total energy consumption with changing of input data size

6 Conclusion

To address the issue of resource wastage resulting from redundant content requests in Telematics, we propose an innovative approach that combines task offloading and content caching optimization. Leveraging the FOST deep learning model, we extract spatial and temporal correlations to forecast traffic patterns. Building upon this, we divide the multi-objective problem of optimizing latency and energy consumption into several single-objective objectives, employing the Tchebycheff weight aggregation method through the MOEA/D algorithm. The effectiveness of our approach, named TOCC, is substantiated through comprehensive experimental demonstrations. Moving forward, our future work will encompass addressing dynamic changes in factors such as resource allocation problems in computational offloading. Furthermore, we aim to extend our approach to more realistic application scenarios, enabling its applicability to diverse and evolving vehicular environments.

Acknowledgement. Pengwei Wang is the corresponding author. This work was partially supported by the National Natural Science Foundation of China (NSFC) under Grant 61602109, DHU Distinguished Young Professor Program under Grant LZB2019003, Shanghai Science and Technology Innovation Action Plan under Grant 22511100700, Fundamental Research Funds for the Central Universities.

References

1. Xia, Z., Wu, J., Wu, L., et al.: A Comprehensive survey of the key technologies and challenges surrounding vehicular ad hoc networks. ACM Trans. Intell. Syst. Technol. **12**(4), 1–30 (2021)
2. Liu, B., Xu, X., Qi, L., et al.: Task scheduling with precedence and placement constraints for resource utilization improvement in multi-user MEC environment. J. Syst. Architect. **114**(6), 101970 (2021)
3. FOST. https://www.msra.cn/zh-cn/news/features/fost, Accessed Dec 07
4. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
5. Zhao, X., Zhao, L., Liang, K., et al.: An energy consumption oriented offloading algorithm for fog computing. In: Quality. Reliability, Security and Robustness in Heterogeneous Networks: 12th International Conference, pp. 293–301. Springer International Publishing, Korea (2017)
6. Chen, M., Hao, Y.: Task offloading for mobile edge computing in software defined ultra-dense network. IEEE J. Sel. Areas Commun. **36**(3), 587–597 (2018)
7. Xu, J., Hao, Z., Sun, X.: Optimal offloading decision strategies and their influence analysis of mobile edge computing. Sensors **19**(14), 3231 (2019)
8. Yang, C., Liu, Y., Chen, X., et al.: Efficient mobility-aware task offloading for vehicular edge computing networks. IEEE Access **7**, 26652–26664 (2019)
9. Zhang, K., Mao, Y., Leng, S., et al.: Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks. In: 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM), pp. 288–294. IEEE (2016)
10. Dai, Y., Xu, D., Maharjan, S., et al.: Joint load balancing and offloading in vehicular edge computing and networks. IEEE Internet Things J. **6**(3), 4377–4387 (2018)
11. Wan, S., Li, X., Xue, Y., et al.: Efficient computation offloading for Internet of Vehicles in edge computing-assisted 5G networks. J. Supercomput. **76**, 2518–2547 (2020)
12. Zhao, J., Li, Q., Gong, Y., et al.: Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks. IEEE Trans. Veh. Technol. **68**(8), 7944–7956 (2019)
13. Fang, Z., Xu, X., Dai, F., et al.: Computation offloading and content caching with traffic flow prediction for internet of vehicles in edge computing. In: 2020 IEEE International Conference on Web Services (ICWS), pp. 380–388. IEEE (2020)

Privacy-Preserving Retrieval Scheme Over Encrypted Medical Records with Relevance Ranking

Wanting Lei[1], Xiehua Li[1](✉), Yingzhu Wang[1], and Xiaoyu Mei[1,2]

[1] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, Hunan, China
beverly@hnu.edu.cn

[2] New Lynn School, 1 Hutchinson Avenue, New Lynn, Auckland 0600, New Zealand

Abstract. Electronic medical records (EMRs) contain a large amount of highly private and sensitive information of patients and medical institutions. For privacy concerns, EMRs are usually encrypted before outsourcing them to the cloud storage platform. However, it is difficult to retrieve the encrypted EMRs accurately and efficiently. The existing encrypted data retrieval schemes can hardly achieve the goals of fuzzy multi-keyword search, relevance ranking and high retrieval accuracy. Thus, this paper proposes a Privacy-preserving Retrieval scheme over Encrypted Medical Records (PREMR) that can satisfy those goals. We utilize the Possibility-Levenshtein based Spelling Corrector (PLSC) to support fuzzy multiple input keywords. A homomorphic-based encryption algorithm is proposed for relevance score encryption and calculation so that the encrypted medical records can be ranked without leaking private information. We theoretically prove that our scheme can achieve data confidential and privacy preserving. With the experiments' evaluation, we analyze the costs and efficiency of our scheme. Finally, the comparison of PREMR with other related schemes shows that our scheme is more efficient and secure.

Keywords: Encrypted medical records · Fuzzy multi-keyword search · Homomorphic encryption · Relevance ranking · Searchable encryption

1 Introduction

Electronic medical records (EMR) are widely used in healthcare systems as the healthcare industry is moving toward digitization. Many hospitals and medical institutes use EMRs for online diagnosis, health screening, and new drug development. Since a large amount of EMRs and images are generated everyday, most hospitals and institutes use the public cloud storage platform to store these data. However, medical records contain highly sensitive personal information that should not be outsourced without protection. A simple way to protect EMRs information is to encrypt them before outsourcing, but this reduces the accuracy and efficiency of EMR retrieval. Aim to solve this problem, the first searchable encryption scheme was proposed by Song *et al.* [1], in which some

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 81–92, 2024.
https://doi.org/10.1007/978-981-99-8211-0_9

basic search approaches over encrypted data were discussed. Boneh *et al.* [2] proposed the first public key encryption scheme. After that searchable encryption becomes an important technology for encrypted data retrieval with privacy preserving [3–5].

Another issue that affects the retrieval accuracy and efficiency is the correctness of the input keywords. Errors in the input keywords would cause inaccurate search results and even retrieval failure. In this paper, we utilize our former proposed Probability-Levenshtein based Spelling Correction (PLSC) algorithm [6, 7] in recommended keywords ranking and medical keywords correction. So that PLSC can support fuzzy multiple keywords input and provide a more accurate search query. Then, we propose a correlation encryption and calculation algorithm based on homomorphic encryption, so that the cloud server can securely complete the calculation of the sum of keywords the relevance scores in the EMR. In addition, proxy is introduced in our scheme to support multiple EMR owners and multi-keyword relevance score ranking. Finally, we compared our PREMR scheme with the newly published searchable encryption scheme for performance evaluation. Our contributions can be summarized as follows.

- In order to test the accuracy of PLSC, we build a library that contains 2000 medical records with more than 3000 medical words. Based on this library, we compare our work with Norvig's spelling corrector and edit distance.
- We design a relevance score encryption and ranking algorithm based on homomorphic encryption to support secure keyword-based query and retrieval. The algorithm adopts Paillier-based encryption to sum up encrypted multi-keyword relevance scores.
- We built up an encrypted EMR retrieval system that can support data outsourcing and dynamic updates for multiple EMR owners. We also implement the performance comparison among PREMR and several similar searchable encryption schemes.

The rest of the paper is organized as follows. Section 2 is the related work on searchable encryption. Section 3 introduces the template of EMR and PLSC correction evaluation. Section 4 presents the constructions and definitions of our scheme. The detailed description of our PREMR scheme is represented in Sect. 5. Theoretical security analysis is given in Sect. 6. We give the scheme implementation results and comparison in Sect. 7. Section 8 is the conclusion of the whole paper.

2 Related Work

Most researches on searchable encryption are aiming at improving accuracy and security of data retrieval. For improving accuracy, Sun *et al.* [9] proposed a multi-keyword search scheme using a vector space model and a cosine measure with TF (word frequency) × IDF (inverse text frequency index) to provide order-preserving document retrieval. Kabir *et al.* [10] improved Sun's scheme by writing the plaintext TF values in the index tree orderly. However, the plaintext TF values may leak information about keywords and documents. To improve security of encrypted document retrieval, Liu *et al.* [11] proposed a verifiable searchable encryption scheme that can verify the correctness of retrieval results over dynamic data collection. Du *et al.* [12] proposed a searchable symmetric encryption scheme that combines access control and boolean queries. Liu *et al.* [13] adopted attribute hierarchy with the comparison-based encryption to achieve dynamic access control over encrypted personal health records. Those searchable encryption schemes are usually considered as a way to guarantee data privacy and search efficiency. Also, there are many researches on searchable encryption schemes with multiple keyword support [14, 15] and are applied in many other areas [16]. However, these schemes have some limitations in retrieval efficiency, accuracy or privacy. In cloud computing applications, especially in medical cooperation projects, the searchable encryption should be able to support precise and efficient retrieval on outsourced medical records for further diagnosis.

Another research topic on searchable encryption is fuzzy search for multiple keywords. Li *et al.* [13] proposed a scheme that used kNN and Euclidean distance to select *k* nearest database records, but the search accuracy is not desirable. Traditional spelling correction algorithms, such as the Levenstein distance, do not achieve high correction accuracy if the spelling error is more than two letters. Zhong [8] proposed a fuzzy search scheme that used k-gram to construct a fuzzy keyword set and Jaccard coefficient to calculate the similarity of keywords. Gnanasekaran [18] converted keyword into a vector, and used LSH (Local Sensitive Hash) to support fuzzy keyword search. Aritomo [19] used simhash to realize the keyword fuzzy search, and the VP-tree to improve search accuracy. K. Wang [20] used LSH to build index, and used Bloom filter to realize fuzzy search over multiple keywords. However, those schemes did not consider the misalignment of letters in the keywords, which may lead to less accurate search results.

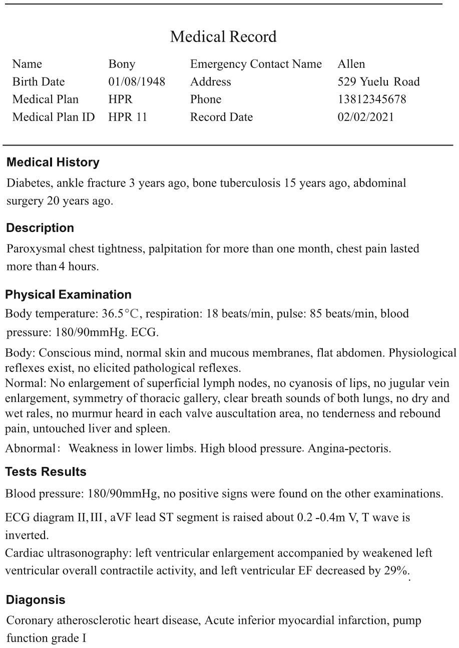
3 Spelling Correction on Electronic Medical Records

3.1 Electronic Medical Records Templates

In order to support fuzzy search, we adopt our previous PLSC (Probability-Levenshtein based Spelling Correction) algorithm [6] to correct the ambiguous input search words. We build a library with 2000 EMRs that contain 3000 common medical terms. The medical terms are selected from [17]. The format example of EMR is shown in Fig. 1. This is a typical EMR, which contains private information such as the patient's name, address and phone number, and also sensitive information such as the patient's condition, diagnosis and prescription.

3.2 Spelling Correction Evaluation of EMRs

We evaluate the PLSC algorithm using our EMRs library. The experiment tests the correction accuracy of PLSC, Norvig's spelling corrector, and edit distance. Table 1 gives the correction probability of three spelling correctors, where spelling errors in each keyword are random. The test result shows that PLSC is able to give more accurate candidate correction especially when there are more than two random errors in the input keywords.



Medical Record

Name	Bony	Emergency Contact Name	Allen
Birth Date	01/08/1948	Address	529 Yuelu Road
Medical Plan	HPR	Phone	13812345678
Medical Plan ID	HPR 11	Record Date	02/02/2021

Medical History

Diabetes, ankle fracture 3 years ago, bone tuberculosis 15 years ago, abdominal surgery 20 years ago.

Description

Paroxysmal chest tightness, palpitation for more than one month, chest pain lasted more than 4 hours.

Physical Examination

Body temperature: 36.5°C, respiration: 18 beats/min, pulse: 85 beats/min, blood pressure: 180/90mmHg. ECG.

Body: Conscious mind, normal skin and mucous membranes, flat abdomen. Physiological reflexes exist, no elicited pathological reflexes.
Normal: No enlargement of superficial lymph nodes, no cyanosis of lips, no jugular vein enlargement, symmetry of thoracic gallery, clear breath sounds of both lungs, no dry and wet rales, no murmur heard in each valve auscultation area, no tenderness and rebound pain, untouched liver and spleen.

Abnormal: Weakness in lower limbs. High blood pressure. Angina-pectoris.

Tests Results

Blood pressure: 180/90mmHg, no positive signs were found on the other examinations.

ECG diagram II, III, aVF lead ST segment is raised about 0.2 -0.4m V, T wave is inverted.
Cardiac ultrasonography: left ventricular enlargement accompanied by weakened left ventricular overall contractile activity, and left ventricular EF decreased by 29%.

Diagonsis

Coronary atherosclerotic heart disease, Acute inferior myocardial infarction, pump function grade I

Fig. 1. Medical Record Template

Table 1. Accuracy comparison with random errors

Errors	PLSC (%)	Norvig's corrector (%)	Edit distance (%)
1 - 2	94.7	89.5	85.1
2	93.2	81.1	73.4
1 - 3	71.6	64.8	60.1

4 System Construction and Preliminaries

This section first introduces our system structure, and then describes threat models, system goals, notations, and cryptographic preliminaries.

4.1 System Model

There are four principals in the PREMR system. EMR owners is responsible for medical data encryption and index building. They upload encrypted EMRs to the cloud service provider (CSP), and send indexes to the Proxy. Proxy merges the indexes from all EMR owners and encrypts the merged index. Then Proxy uploads the secured index to the CSP. The encrypted EMRs and index are uploaded by EMR owners and Proxy, respectively. Meanwhile, EMR owners distribute decryption keys to authorized users via secure channel.

In our scheme, the EMR storage server is considered as an "honest-but-curious" entity. Specifically, the storage server will honestly implement the protocol, but also curiously analyze the index, stored data and queries to capture more information associated with plaintext EMRs. EMR owners are suppose to be honest because they have the original plaintext records. Proxy is a trustworthy entity who builds up secured index for outsourced data, and generates trapdoors for users' searching queries. Users are untrusted, they may collude with others to get more information about the encrypted EMRs. Secret keys are uncompromised.

4.2 Notations

- R: plaintext EMR set, R = $\{R_1, R_2, \ldots, R_n\}$;
- R′: encrypted EMR set, R′ = $\{R_1', R_2', \ldots, R_n'\}$;
- ID: EMR identifier in plaintext $ID = \{id_1, id_2, \ldots, id_n\}$;
- ID': encrypted EMR identifier, $ID' = \{id_1', id_2', \ldots, id_n'\}$;
- SW: keywords set in plaintext, SW = $\{W_1, W_2, \ldots, W_m\}$;
- SW′: keywords set in ciphertext, SW′ = $\{\ W_1', W_2', \ldots, W_n'\}$;
- $S_{i,j}$: plaintext relevance score of keyword W_i in R_j; S_j: sum of relevance score in R_j in plaintext;
- $S_{i,j}'$: encrypted relevance score of keyword W_i in document R_j; S_j': sum of relevance score in R_j in ciphertext;
- $\mathbb{Z}_{y^2}^*$ is the set of integers range between 1 and y^2.

Our PREMR scheme includes three major processes: EMR index building, encrypted EMR searching and queue-based ciphertext retrieval.

4.3 Cryptographic Preliminaries

In PREMR scheme, we adopt both symmetric key algorithm and homomorphic encryption to guarantee the security of EMR and the value of relevance scores. The symmetric key algorithm (SKA) is used to encrypt keywords, EMR identifiers, and EMRs. The homomorphic encryption (HE) is used to encryption the relevance score of each keyword in every EMR. The algorithms that are involved in the PREMR system are defined as followed.

- SKA = (T, K, ENC1, DEC1) is a symmetric key encryption algorithm, where T is the input data, K is the symmetric key, ENC1 is the encryption algorithm; DEC1 is the decryption algorithm.
- HE = (RS, PK, SK, ENC2, DEC2) is a Paillier-based homomorphic encryption, where RS is the relevance score of a keyword, PK is the public key to encrypt RS, SK is the secret key. ENC2 and DEC2 are the encryption and decryption algorithms. PK and SK are generated with the followed method:

  1. Suppose $p, q \in Z_n$ are two large prime numbers, and gcd(pq,(p − 1)(q − 1)) = 1, Φ(n) = (p − 1)(q − 1). Let n = pq, λ = lcm(p − 1, q − 1).

2. The multiplicative subgroup $Z_n \times Z_n^* \rightarrow Z_{n^2}^*$. $\left|Z_{n^2}^*\right| = \Phi\left(n^2\right) = n\Phi(n)$. g is some element of $Z_{n^2}^*$, $r \in (0, n)$ is a random integer, and $\gcd(r, n) = 1$, $r^{(p-1)} \equiv 1(\text{mod } p)$. $r^{\lambda} = 1 \text{ mod } n$, $r^{n\lambda} = 1 \text{ mod } n^2$.
3. Let $L(x) = \frac{x-1}{n}$, the modular multiplicative inverse $\mu = (\text{L}(g^{\lambda} \text{ mod } n^2))\text{-1mod } n$.
4. The public key is $\text{PK} = (n, g)$ and the secret key $\text{SK} = \lambda$.

5 Encrypted EMR Searching with Privacy Preserving

This section introduces the index building process and EMR searching. Then it describes the relevance score calculation and ranking algorithms.

5.1 EMR Index Building

Before encrypting EMRs, the owners first extract keywords, and build inverted plaintext indexes. Subsequently, EMR owners encrypt and upload the medical records to the cloud server, and at the same time send the plaintext index to the Proxy. Proxy collects indexes from all EMR owners, merges and builds up the secure inverted index.

Plaintext Index Building and EMR Encryption. EMR owners first extract keywords from EMRs, calculate the TF-IDF value for each keyword as its relevance score, and then build the plaintext index **I**. $\mathbf{I} = \{\text{I}_1,\text{I}_2,\text{I}_3\ldots\text{I}_m\}$, $\text{I}_i = (W_i, \bigcup_j < id_j, S_{i,j} >)$, I_i is the inverted index of keyword W_i, id_j is the identifier of the EMR that contains W_i, $S_{i,j}$ denotes the TF-IDF score of keyword W_i, in the EMR with the identifier id_j.

Furthermore, EMR owner implements SKA (*, K_1, ENC1) to encrypt EMRs. Equation (1) describes the encryption process.

$$\begin{gathered} id_j' \leftarrow \text{KA}(id_j, K_1,\ \text{ENC1}) \\ R_j' \leftarrow (R_j, K_1,\ \text{ENC1}) \\ C = \left\{\left(id_1', R_1'\right), \left(id_2', R_2'\right), \ldots, \left(id_n', R_n'\right)\right\} \end{gathered} \tag{1}$$

EMR owners then send **I** to the Proxy, and upload C to the CSP.

Indexes Merging and Encryption. In our system, we support multiple EMR owners to outsource their medical records. Proxy is introduced to handle multiple indexes merging and secure index building, so that even though EMRs are encrypted with different keys the retrieval can still be accurate and efficient. The secure index $\mathbf{I}'$ is generated with the followed steps.

Step 1. Proxy receives multiple indexes from different EMR owners and merges them into a new index based on keywords.
Step 2. Proxy implements SKA$((*, K_2, \text{ENC1})$ to encrypt keywords and EMR identifiers.

$$\begin{gathered} W_i' \leftarrow \text{SKA}(W_i, K_2,\ \text{ENC1}) \\ id_j'' \leftarrow \text{SKA}(W_i, K_2,\ \text{ENC1}) \end{gathered} \tag{2}$$

Comparing Eq. (1) and Eq. (2) we can see that different encryption keys(K1, K2) are used to encrypt the same EMR identifier(id_j), so that the linkability of the index and stored EMR is broken.

Step 3. Proxy runs HE(RS, PK, ENC2) to encrypt the relevance score $S'_{i,j}$. The encryption process is defined in Eq. (3).

$$S'_{i,j} = g^{S_{i,j}} \times r^n mod n^2 \tag{3}$$

At last, Proxy establishes the secure index $\mathbf{I}'$ and upload it to the CSP. The format of the secure index is defined in Eq. (4).

$$\mathrm{I'} = \left\{\mathrm{I}_1^{'}, \mathrm{I}_2^{'}, \ldots, \mathrm{I}_i^{'}\right\}, \mathrm{I}_i^{'} = \left\{W_i^{'}, \bigcup\nolimits_j \left\langle id_j^{''}, S_{i,j}^{'}\right\rangle\right\} \tag{4}$$

5.2 Encrypted EMR Retrieval

When user tries to search a set of keywords, the PLSC algorithm will first correct the misspelled ones. Then user sends the plaintext keywords set $\mathbf{SW} = \{W_1, W_2, \ldots, W_t\}$ to the Proxy. Proxy generate the query trapdoor $\mathbf{SW}' = \{W'_1, W'_2, \ldots, W'_t\}$, where $W'_i =$ SKA(W_i, K_2, ENC1).

Algorithm 1 Ciphertext searching by CSP

```
Input: $SW' = \{W'_1, W'_2, ..., W'_t\}$;
Output: EMR identifiers, $S'_j$
function EMR SEARCHING
$\mathbf{I}'_r = \mathbf{I}'$;
    for ($i = 1$; $i \le t$; $i$++) do
        Search $\mathbf{I}'$;
        if $W'_i \in \mathbf{I}'_i.\ W'_i$ then $\mathbf{I}'_r = \mathbf{I}'_r \cap \mathbf{I}'_i$;
        end if
    end for
    while $\mathbf{I}'_r \neq \emptyset$ do
        for each $\mathbf{I}'_r.id_j$ do
            $\mathbf{I}'_r.\ S'_j = \prod_{i,j} S'_{i,j}$;
        end for
    end while
    return ($\mathbf{I}'_r$)
end function
```

CSP Searching Algorithm. SP searches $\mathbf{SW}'$ in $\mathbf{I}'$. The searching algorithm is described in Alg.1. Search result is the conjunction of EMRs that contain all queried keywords in $\mathbf{SW}'$. Subsequently, CSP sums the encrypted relevance scores of multiple keywords in each EMR. The relevance score calculation is defined in Eq. (5).

$$S_j' = \prod_{i,j} S_{i,j}' = g^{\sum S_{i,j}} \times \prod_i r_i^n mod n^2 \tag{5}$$

CSP returns the search result to the Proxy for relevance score decryption and ranking.

Relevance Ranking Algorithm. After receiving the search results from CSP, Proxy needs to decrypt and rank the summation of relevance scores for each returned EMR. Proxy implements SKA($*$, K_2, DEC1) to get the plaintext keywords W_i and EMR identifiers idi. Then, Proxy runs HE(S_j', SK, DEC2) to decrypt the sum of relevance score. The decryption process is defined in Eq. (6).

$$\begin{aligned} S_j &= \frac{L\left(S_j'^{\lambda} mod n^2\right)}{L\left(g^{\lambda} mod n^2\right)} mod n \\ &= \frac{L\left(g^{\lambda \sum S_{i,j}} \times \prod_i r_i^{\lambda n} mod n^2\right)}{L\left(g^{\lambda} mod n^2\right)} mod\ n \\ &= \sum S_{i,j} \end{aligned} \tag{6}$$

where $\prod_i r_i^{\lambda n} \equiv 1$. Proxy ranks the top-$k$ EMRs based on their $\sum S_{i,j}$ and returns the EMR identifiers back to users. Upon receiving the EMR identifiers, users send downloading requests to the CSP directly.

6 Security Analysis

This section analyzes the data confidentiality and private-preserving of our scheme. We have proved that our scheme can guarantee the security of ciphertext retrieval by using the queue-based search strategy, and can protect the EMR privacy through different encryption algorithms.

Data Confidential. The original EMRs are encrypted before outsourcing to the CSP and the decryption keys are distributed to users via secure channel. Based on the assumption we made in the system model in Sect. 4, EMRs can not be compromised without correct secret keys. Thus, EMR data confidential can be guaranteed.

Indexes are constructed separately by the EMR owners, then merged and encrypted by the Proxy. EMR identifiers in the index and in the outsourced EMRs are encrypted with different keys so that CSP cannot get the relationship of the encrypted EMRs and the encrypted index. Keywords relevance scores of each EMR are encrypted and calculated with the homomorphic encryption, CSP cannot get any information from the keywords and their relevance scores. Therefore, as long as the encryption keys are not compromised, the confidentiality of data, index, keywords and relevance scores can be guaranteed.

Possibility of Privacy Leakage. Queries are encrypted by proxy and then forwarded to CSP. So that, CSP cannot get user information and user privacy is protected. Meanwhile, the file downloading requests and query trapdoors are sent by users and proxy separately. It is impossible for the CSP to guess the exact correspondence between the queried keyword and the downloaded EMRs.

7 Performance Test

The performance test experiments are implemented by C++ programming language on Windows 7 machines, each of which is with an Intel(R) Core(TM) i5 6500 3.2 GHz processor and a 2GB RAM. The performance is evaluated with our own EMR dataset. Our dataset uses more than 3000 medical keywords to generate 2000 EMRs containing various diseases. We compare our scheme with the most relevant researches on searchable encryption: FMS [13], TBMSM [**Error! Reference source not found.**] and Zhong's scheme [8]. In the experiments, the number of keywords in 2000 EMRs varies from 1000 to 3000, and the number of EMRs varies from 100 to 2000.

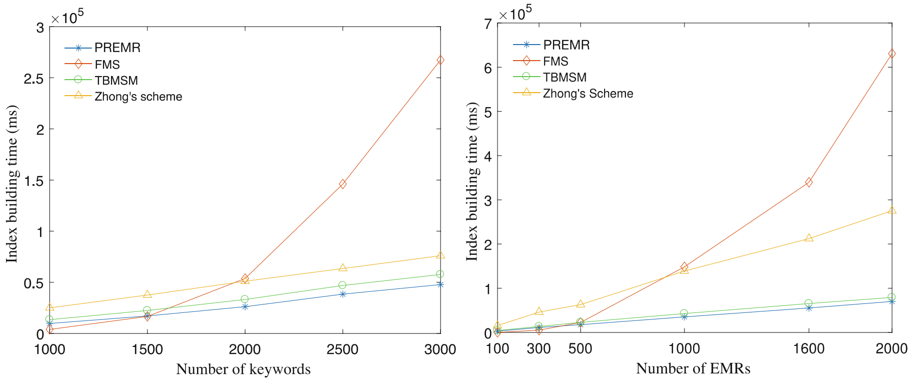
7.1 Index Building Efficiency

We compare the index building time and storage cost among four schemes. Figure 2(a) shows the time overhead required to build an index with the increasing number of keywords. The index building time of FMS grows exponentially since it needs to create an index vector for each document. When the number of keywords exceeds 1500 the index building time of FMS is more than that of other three schemes. While the time cost on building index with other three schemes are stable and increase linearly. The index structure of our PREMR scheme is the inverted index based on keywords. Therefore, the index generation time increases linearly with the increase of keywords. Figure 2. Index Building Timeshows the index generation time with the increase number of EMRs. Our PREMR takes less time to build the index than other three schemes. Compared with other searchable encryption methods, our PREMR is the most efficient one on index building stage.

Figure 3(a) shows the index storage size when the number of index keywords is 1000, 1500, 2000, 2500, and 3000 respectively. When the number of keywords in the index is greater than 1000 or the number of EMRs in the data set is greater than 300, the index storage overhead of our PREMR is less than that of other three schemes. Figure 3(b) shows the required index storage space with the number of EMRs ranges from 100 to 2000. I It indicates that PREMR scheme has better index generation efficiency and less index storage overhead than the other three schemes.

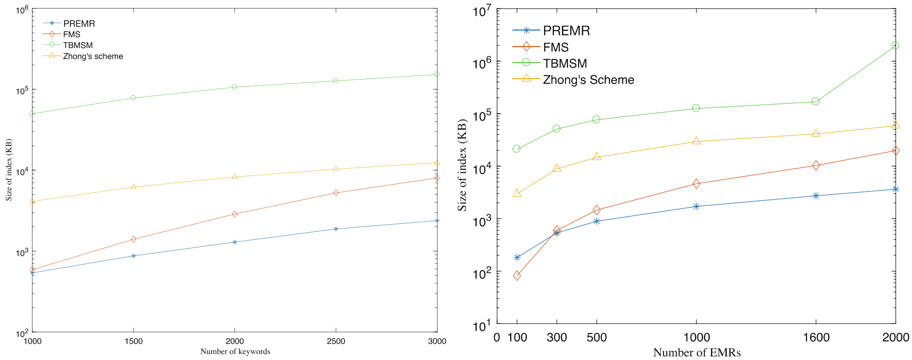
7.2 Trapdoor Generation Time

Figure 4 compares the trapdoor generation efficiency of these four schemes when there are 1000 queries, and the keywords in each query ranging from 10 to 50. Figure 4 shows that the trapdoor generation time of FMS is not affected by the number of queried



(a) Number of keywords (b) Number of EMRs

Fig. 2. Index Building Time



(a) Number of keywords (b) Number of EMRs

Fig. 3. Index Storage Space

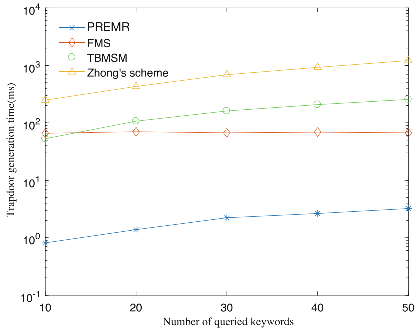


Fig. 4. Trapdoor Generation Time

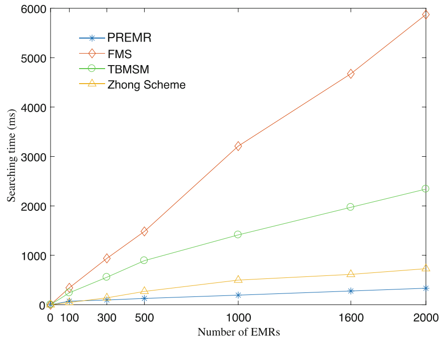


Fig. 5. Search Efficiency Comparison

keywords. This is because that the trapdoor in FMS is a fixed-length one-dimensional vector corresponding to the keywords, even though the number of keywords increases, the trapdoor generation time remains basically unchanged. The trapdoor generation time of PREMR, TBMSM and Zhong's scheme grows linearly with the increase of queried keywords. From comparison result, it shows that the PREMR scheme has a better performance on trapdoor generation efficiency, especially in supporting multiple keywords and simultaneous queries.

7.3 Search Efficiency

Figure 5 shows the search efficiency of compared schemes. All schemes are evaluated with the number of EMRs ranging from 100 to 2000, and the number of keywords in each query is 5. In FMS, a matrix calculation is carried out between the retrieval vector and index vector of each EMR, which increases the search time significantly with the increase of stored EMRs. In TBMSM scheme, a search sequence should be obtained firstly by matching each search keyword with that in the index. So that, the search time in TBMSM increases linearly with the number of keywords in the index. The search efficiency in Zhong's scheme is mainly affected by the mapping operation of the index and query vectors with LSH (Local Sensitive Hash) function. Although our PREMR scheme is also affected by the number of keywords, the search time grows slowly. Form Fig. 5 we can see that our PREMR scheme has less search time than the other schemes. The search time of PREMR is less than 1s even though there are 2000 encrypted EMRs in the database.

8 Conclusion

This paper proposed a privacy-preserving retrieval scheme over encrypted medical records. The proposed scheme can achieve multi-keyword fuzzy search and relevance ranking. In this paper, we use PLSC to support the fuzzy input keywords and improve spelling correction. In addition, homomorphic encryption algorithm is introduced to support keywords relevance scores calculation and ranking securely. Then, the theoretical proofs show that our PREMR scheme can guarantee the security of query vectors and stored EMRs. Finally, we experimentally analyzed and compared the PREMR with three other similar schemes, and the experimental results proved that the PREMR has better performance in index building, query trapdoor generation and search efficiency.

Acknowledgement. This work was supported by the National Natural Science Foundation of China under Project 92067104 and 61872134, and also supported by the Hunan Provincial Natural Science Foundation under Project 2021JJ30140.

References

1. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Proceedings of S&P, Berkeley, CA, USA, pp. 44–55 (2000)

2. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Proceedings of EUROCRYPT, Interlaken, Switzerland, pp. 506–522 (2004)
3. Li, H., Liu, D., Dai, Y., Luan, T.H., Shen, X.S.: Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage. IEEE Trans. Emerg. Top. Comput. **3**(1), 127–138 (2015)
4. Li, R., Liu, A.X., Wang, A.L., Bruhadeshwar, B.: Fast and scalable range query processing with strong privacy protection for cloud computing. IEEE/ACM Trans. Networking **24**(4), 2305–2318 (2016)
5. Lei, X., Tu, G.-H., Liu, A.X., Xie, T.: Fast and secure kNN query processing in cloud computing. In: Proceedings of CNS, pp. 1–9 (2020)
6. Li, X., Li, F., Jiang, J., Mei, X.: Paillier-based fuzzy multi-keyword searchable encryption scheme with order-preserving. Comput. Mater. Continua **65**(2), 1707–1721 (2020)
7. Li, X., Long, G., Li, S.: Encrypted medical records search with supporting of fuzzy multi-keyword and relevance ranking. In: Proceedings of ICAIS, Dublin, Ireland, pp. 85–101 (2021)
8. Zhong, H., Li, Z., Cui, J., Sun, Y., Liu, L.: Efficient dynamic multi-keyword fuzzy search over encrypted cloud data. J. Netw. Comput. Appl. **149**, 102469 (2020)
9. Sun, W., et al.: Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. IEEE Trans. Parallel Distrib. Syst. **25**(11), 3025–3035 (2014)
10. Kabir, T., Adnan, M.A.: A dynamic searchable encryption scheme for secure cloud server operation reserving multi-keyword ranked search. In: Proceedings of SysS, Dhaka, Bangladesh, pp. 1–9 (2017)
11. Liu, Q., Tian, Y., Wu, J., Peng, T., Wang, G.: Enabling verifiable and dynamic ranked search over outsourced data. IEEE Trans. Serv. Comput. **15**(1), 69–82 (2022)
12. Du, L., Li, K., Liu, Q., Wu, Z., Zhang, S.: Dynamic multi-client searchable symmetric encryption with support for boolean queries. Inf. Sci. **506**, 234–257 (2020)
13. Li, H., Yang, Y., Luan, T.H., Liang, X., Zhou, L., Shen, X.S.: Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data. IEEE Trans. Dependable Secure Comput. **13**(3), 312–325 (2016)
14. Pakniat, N., Shiraly, D., Eslami, Z.: Certificateless authenticated encryption with keyword search: enhanced security model and a concrete construction for industrial IoT. J. Inf. Secur. Appl. **53**, 102525 (2020)
15. Wang, C., Yuan, X., Cui, Y., Ren, K.: Toward secure outsourced middlebox services: practices, challenges, and beyond. IEEE Network **32**, 166–171 (2018)
16. Hao, J., Huang, C., Ni, J., Rong, H., Xian, M., Shen, X.S.: Fine-grained data access control with attribute-hiding policy for cloud-based IOT. Comput. Netw. **153**, 1–10 (2019)
17. Stedman, T.L.: The American Heritage Stedman's Medical Dictionary, Edition 2. Houghton Mifflin Company (2002)
18. Xia, Z., Wang, X., Sun, X., Wang, Q.: A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans. Parallel Distrib. Syst. **27**(2), 340–352 (2016)
19. Aritomo, D., Watanabe, C., Matsubara, M., Morishima, A.: A privacy-preserving similarity search scheme over encrypted word embeddings. In: Proceedings of IIWAS, New York, NY, USA, pp. 403–412 (2019)
20. Li, M., Wang, G., Liu, S., Yu, J.: Multi-keyword fuzzy search over encrypted cloud storage data. Procedia Comput. Sci. **187**, 365–370 (2021)

A Data-Centric Approach for Efficient and Scalable CFD Implementation on Multi-GPUs Clusters

Ruitian Li[1], Liang Deng[1](✉), Zhe Dai[1], Jian Zhang[1], Jie Liu[2], and Gang Liu[1]

[1] China Aerodynamic Research and Development Center, Computational Aerodynamic Institute, Mianyang, China
dengliang11@nudt.edu.cn

[2] Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha, China

Abstract. Scalability is a crucial factor determining the performance of massive heterogeneous parallel CFD applications on the multi-GPUs platforms, particularly after the single-GPU implementations have achieved optimal performance through numerous optimizations. A novel Data-Centric hybrid MPI-CUDA CFD model is proposed in this paper to enable efficient scalability of CFD applications on large-scale heterogeneous platforms. Based on the Data-Centric approach, Minimum-cost MPI transfer strategy and the code refactoring technique are realized for a better balance between data transfer and floating-point computation performance, which could significantly improve the scalability and reduce the time-to-solution. Subsequently, those approaches are integrated into the industrial unstructured CFD software, FlowStar, to evaluate their effectiveness. Numerical results demonstrate that Minimum-cost MPI strategy achieves more than 2.0 times performance improvement compared to the traditional Model-Centric implementation, and the code refactoring technique boosts performance by 40% to 50% over the minimum-cost MPI version. Moreover, the Data-Centric implementation on 64 A100 GPUs platform show a speedup ratio of over 120 when compared to the original MPI implementation with 64 ranks.

Keywords: Data-Centric · massive heterogeneous parallel CFD · MPI-CUDA · performance scalability

1 Introduction

Graphics Processing Units (GPUs) have revolutionized the HPC landscape in the past decades [1] and offer tremendous potential for applications in Computational Fluid Dynamics (CFD) [2]. Over the past decade, many CFD codes have been formulated MPI-X hybrid programming models for expressing parallelism to run as efficiently as possible on the modern heterogeneous systems [3]. Here, X refers to the programming model designed specifically for GPUs, such as CUDA, HIP or OpenCL. To run CFD codes on such heterogeneous platforms efficiently, some fine-grained parallel models

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 93–104, 2024.
https://doi.org/10.1007/978-981-99-8211-0_10

have been developed. For example, kernels based on the optimized decoupling strategies [4] which can provide better data locality and make use of the share memory on GPU would hand over higher performance. In the same way, the CFD codes with fine-grained and fast convergent iteration solver [6] would spend less time to reach the solution of fluid flow. Those optimization strategies actually represent the CFD performance optimization direction which aims to reduce time consuming of computational kernels. However, there is a notable disparity in the performance of CFD solver with optimization algorithms when comparing their performance on a single-GPUs machine versus a multi-node and multi-GPUs platform. That is, the algorithm has not shown good scalability when scaled to larger systems or multiple computing units.

Many literatures have reported the issue of poor scalability in CFD applications, which is mainly attributed to the additional data communication overheads that arise when solving CFD problems on multi-node computing systems [6, 7]. Especially on supercomputers, the cost of data transfer is prohibitively high compared to numerical computations, and it is desirable that an efficient application should achieve a better balance between data transfer and floating-point computation performance. However, due to convergence and robustness considerations, the real and large-scale industrial CFD applications require broadcasting the latest information to neighboring zones after each module calculation [8]. This frequent and extensive communication often leads to poor overall performance of CFD applications, especially for the one on multi-GPU heterogeneous platform, which involves the data transfer between the device memory and the host memory, and the communication throughout different host nodes. Due to the strong negative impact of communication on performance, the optimization algorithms for computational kernels mentioned above cannot effectively improve the overall performance of CFD on multi-GPU platform. Currently, there is no effective way to address the weak scalability issue of industrial CFD applications on large-scale multi-GPU platforms.

Data-Centric model is a programming paradigm that emphasizes data as the central element of software design and development [9]. This model emphasizes generating the appropriate data structure and organizing the data itself to build applications. With this approach, developers can create efficient workflows with the help of data sources and data-driven features [10]. Data-Centric model is a powerful approach for high-performance software development [11]. In contrast, Model-Centric methodology emphasizes creating models or functions as the primary focus of the application development process. CFD software are a typical example of Model-Centric development paradigm, where the codes encompass Euler/NS solver and turbulent RANS/LES/DNS module, each of which contain models such as Gradient, Limiter, Flux, and Linear Equation Solvers. These models, taken as a whole, construct the CFD application from Model-Centric perspective. Thus Model-Centric mode is suitable for CFD development and code collaboration, but for the high-performance and scalable implementation, Model-Centric mode may exhibit some performance bottlenecks. A Data-Centric design of CFD may be a potential way to address the scalable problem. However, there is no literature proposing Data-Centric model to improve the overall performance of CFD and Model-Centric mode is still the predominant development method for CFD applications.

In this paper, a Data-Centric CFD focusing on MPI-CUDA framework is invented for the scalable and efficient realization of industrial unstructured CFD on multi-GPU clusters. The contribution of this work is reflected in several aspects. Firstly, this work is the first to employ Data-Centric model to develop and design CFD codes, which has resulted in significant performance improvement compared with Model-Centric one. Secondly, this paper focuses on proposing a feasible and general method to address the current weak scalability issues of industrial CFD on large-scale heterogeneous platforms. Through Data-Centric reorganization of the original codes, a new CFD framework have been developed for a balance between data transfer and floating-point computation. Thirdly, this work introduces a Data-Centric API to focus on high-performance implementation of CFD while retaining the conventional Model-Centric one which is more suitable for the model development and functionality expansion of CFD software. This dual-mode implementation is a first in high-performance CFD, and significantly improves the computational performance and functionality of CFD codes.

The paper is organized as follows. Section 2 describes the numerical foundations and the parallel strategies in CFD. Section 3 presents the techniques of Data-Centric analysis and redesign of MPI-CUDA CFD codes. Results are given in Sect. 4. Conclusion is included in Sect. 5.

2 Numerical Foundations

The general conservative equation of fluid flow can be written as following:

$$\frac{\partial}{\partial t}\int_{\Omega}\overrightarrow{W}d\Omega+\oint_{\partial\Omega}\left(\overrightarrow{F_C}-\overrightarrow{F_V}\right)dS=\int_{\Omega}\overrightarrow{Q}d\Omega \tag{1}$$

Here $\overrightarrow{W}$=[ρ, ρu, ρv, ρw, ρE]^T represents the conservative variables, where $\overrightarrow{F_C}$ and $\overrightarrow{F_V}$ stand for the inviscid and viscous flux respectively, and $\overrightarrow{Q}$ is the source item. And ρ is the fluid density, u, v and w refer to the velocity component at x, y and z coordinate, E stands for the internal energy, and t is the time item. After the discretization of Eq. (1), one can get the discretize conservative equation:

$$\frac{d\overrightarrow{W_I}}{dt}=-\frac{1}{\Omega_I}\left[\sum_{m=1}^{N_F}\left(\overrightarrow{F_C}-\overrightarrow{F_V}\right)_m\Delta S_m-\left(\overrightarrow{Q}\Omega\right)_I\right] \tag{2}$$

Using special numerical algorithms to solve above discrete equations can give us the numerical solution of fluid flow. Here the cell-centered FVM discretization is selected while the implicit linear iteration, Lower-Upper Symmetric Gauss-Seidel (LUSGS), is employed to solve compressible laminar or turbulent problems with Venkatakrishnan's Limiter function and Spalart-Allmaras one-equation model (SA). The gradient information is evaluated with Node-Based Green-Gauss approach while the Roe scheme and the central discretization are used for the convective solution reconstruction process and the viscous flux calculation respectively.

The coarse-grained parallel framework has been developed through MPI framework. In order to achieve strong robustness and fast convergence in CFD computation, MPI implementation introduces multiple information exchange modules among the ghost cells within neighbor domains, such as the update of ghost node data for Node-based

Green-Gauss gradient, the update of ghost limit and gradient information. To further obtain acceleration on heterogeneous platform, fine-grained parallel models should be developed. For the face-based loops in right-hand side(RHS) calculation, the reduction strategy [15] is used here to process the face data races while the balancing cell-color decoupling model [16] is employed for the implicit left-hand side(LHS) iteration. After the fine-grained remolding of those computational tasks, like gradient terms, and the application of some optimization strategies, such as the mesh renumbering [17] and the share memory model, researchers could develop an efficient single-GPUs implementation. Combining the coarse-grained MPI framework and the fine-grained single-GPU CUDA implementation results in the original MPI-CUDA parallel implementation, as shown in Fig. 1.

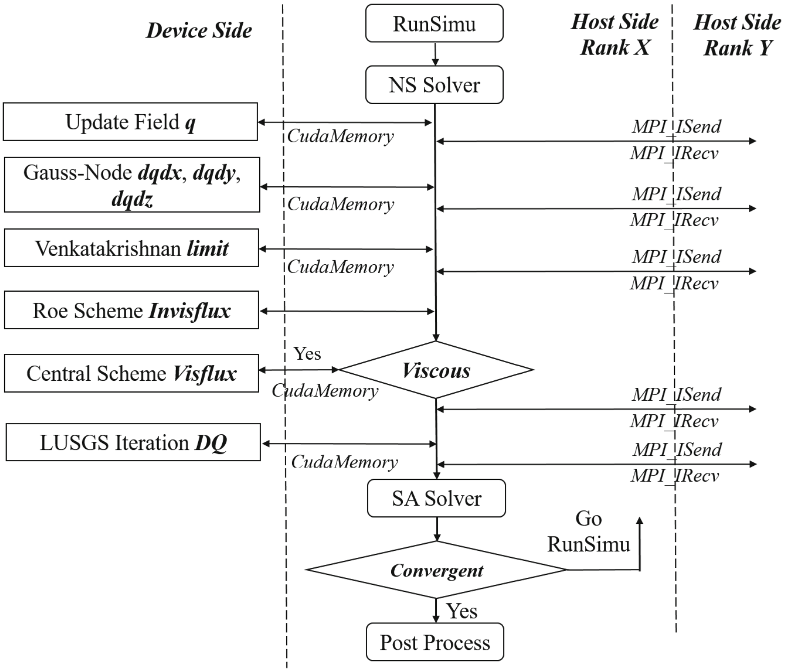


Fig. 1. Framework for MPI-CUDA implementation of CFD codes

3 Data-Centric MPI-CUDA CFD Implementation

The original MPI-CUDA implementation that develops from Model-Centric prospective would result in the weak scalability problem attributed to the additional data communication overheads that arise inside or at the end of computational models, including gradient, limit, flux and SA model. In this section, the Data-Centric viewpoint would be used to analyze and reconstruct MPI-CUDA CFD codes with a better balance between data transfer and floating-point computation performance.

3.1 Minimum-Cost MPI Transfer Strategy

Data-Centric analysis of MPI modules is presented firstly. The schematic diagram of the domain decomposition for MPI framework is presented in Fig. 2, where the original computational domain is divided into the sub-regions and the ghost cells' arrays are set to transfer the latest information among the neighbor sub-regions. Typically, each of MPI ranks would process the computation in one of sub-regions and employ the *MPI_Isend* and *MPI_Irecv* functions to accomplish those ghost layers' information exchange. In Model-Centric MPI-CUDA implementation, the existence of above data transfer would put a negative effect on the whole parallel performance since *MPI_Isend* and *MPI_Irecv* functions must be carried out at the host-node side, which bring the idle time of GPUs' cores.

In the original MPI-CUDA implementation, almost all computational tasks are designed to perform on GPUs and thus the calculating data, such as the flow fluid variables and etc., are all stored in the device memory, namely the video memory of GPUs. The calculating variables, such as the fluid flow velocity, the gradient and the limit, are set to equal $\mathrm{N} = nTCell + nBFace$, where *nTCell* is the number of the real cells to be solved and *nBFace* is the length of the ghost cells' array. The ghost cells' array is set to store the latest boundary information from the neighbor rank. Since all computational tasks are finished on GPUs, thus only the ghost cells' array needs to be transferred back to the host node for the MPI information transfer, while the data in the real cells always stay on GPUs side. After the transfer of *nBFace* array to the host memory, the host node would run *MPI-bqs* module to assemble the *bqs* array from the *nBFace* ghost cells' array for *MPI_Isend* function. Similarly, *MPI-bqr* module would process the *bqr* data received by *MPI_Irecv* function from the neighbor ranks to get the updated *nBFace* array. Thus the MPI module in MPI-CUDA implementation contains three sub-modules, the *cudaMemcpy* operation for the exchange of the *nBFace* array, the assembly between the *nBFace* array and the *bqs*/*bqr* array and the *MPI_Isend* and *MPI_Irecv* communication, both mainly finished by the host node.

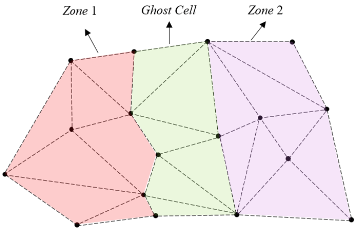


Fig. 2. Domain Decomposition for Parallel Implementation of MPI

For the host side *MPI_Isend* and *MPI_Irecv* function, the *nBFace* array is a crude data where only after finishing the assembly of the *bqs* array from the *nBFace* array, *MPI_Isend* function can start their communication among ranks. One way to decrease the running time of those MPI modules is to parallel assemble *bqs*/*bqr* array on GPU. There are two types of kernels to assemble *bqs*/*bqr* array, including of the information process on the ghost cells and the ghost nodes. The *bqr*/*bqs* of the ghost cells can be processed parallel since only numerical exchange happens in those modules while for MPI transfer of the ghost nodes for the Node-based Green-Gauss approach, the *AtomicAdd* operation would be employed to process the accumulation for the node value. Then *MPI_Isend* and *MPI_Irecv* function could see the processed ready data after the *cudaMemcpy* operation and start *MPI_Isend* and *MPI_Irecv* immediately.

Another way to improve the MPI transfer performance from data level is the transfer merging of the assembly of the *bqs*/*bqr* array for multiple variables in the fluid flow. Since the ghost cells indirectly stay in *q* [5] [N] and the merging assembly collects those indirect data into a continuous *bqr*/*bqs* array. Those data encapsulation process is also finished on GPUs and the host node only calls one *cudaMemcpy* and one subsequent *MPI_Isend* and *MPI_Irecv* to accomplish all MPI information transfer. For the fluid variables *q* [5] [N] and the similar data, this transfer merging can reduce the number of calls of MPI module from five to one, while for the gradient data, the reduction is from fifteen (five gradient variables multiple three coordinate components) to one. The parallel data processing and the data transfer merging make up the Minimum-cost MPI Transfer Strategy.

3.2 Code Refactoring Strategy

The Minimum-cost MPI Transfer Strategy focuses on reducing time comsuming of MPI modules in CFD and the coresponding Data-Centric reconstruction is confined to MPI modules. This section would give a global Data-Centric analysis and attempt to create efficient CFD workflows for a better balance between data transfer and floating-point computation performance with the help of global data relationships.

The Model-Centric programming flowchart in Fig. 1 could reveal some data relationships in CFD. For the NS solver, after the update of the fluid flow variables *q* [5] [N], those variables would keep unchanged in the NS solver until the program has obtained the new *DQ* [5] [N] values through the LUSGS iteration and updated *q* [5] [N] again, as shown in Eq. (3):

$$q[j][i] = q[j][i] + DQ[j][i], \quad 0 \leq i < nTCell,\ 0 \leq j < 5 \tag{3}$$

The time step array *dt*[N], used in the LUSGS iteration, is evaluated from *q* [5] [N], as presented in Eq. (4):

$$dt[i] = f_{time}(q[5][N]), \quad 0 \leq i < nTCell \tag{4}$$

The gradient information *dqdx* [5] [N], *dqdy* [5] [N] and *dqdz* [5] [N] are also calculated from *q* [5] [N], while the limiter array *Limit* [5] [N] depends on *q* [5] [N], *dqdx* [5] [N], *dqdy* [5] [N] and *dqdz* [5] [N], as shown in Eq. (5) and Eq. (6):

$$(dqdx[j][i],\ dqdy[j][i],\ dqdz[j][i]) = f_{grad}(q[5][N]), \quad 0 \leq i < nTCell,\ 0 \leq j < 5 \tag{5}$$

$$\begin{aligned} Limit[j][i] = f_{limit}(&q[5][N],\ dqdx[5][N],\ dqdy[5][N], \\ &dqdz[5][N]), \quad 0 \leq i < nTCell,\ 0 \leq j < 5 \end{aligned} \tag{6}$$

Then based on *q* [5] [N], *dqdx* [5] [N], *dqdy* [5] [N], *dqdz* [5] [N] and *Limit* [5] [N], the inviscid flux on the faces, *Invisflux* [5] [N], is calculated through Eq. (7):

$$\begin{aligned} Invisflux[5][i] = f_{Invflux}(&q[5][N],\ dqdx[5][N],\ dqdy[5][N],\\ &dqdz[5][N],\ Limit[5][N]),\quad 0 \le i < nTCell \end{aligned} \tag{7}$$

If the flow problems are involving of the viscous force, the viscous effect would be added into *Visflux* [5] [N], where the temperature *T*[N] and its gradient *dTdx*[N], *dTdy*[N] and *dTdz*[N] should be evaluated from *q* [5] [N] firstly before *Visflux* [5] [N] calculation, as shown in Eq. (8) and Eq. (9):

$$(T[i],\ dTdx[i],\ dTdy[i],\ dTdz[i]) = f_{temp}(q[5][N]),\quad 0 \le i < nTCell \tag{8}$$

$$\begin{aligned} Visflux[j][i] = f_{visflux}(&q[5][N], dqdx[5][N], dqdy[5][N],\\ &dqdz[5][N], Limit[5][N],\ T[i],\\ &dTdx[i], dTdy[i], dTdz[i]),\quad 0 \le i < nTCell,\ 0 \le j < 5 \end{aligned} \tag{9}$$

After the accumulation of *Invisflux* [5] [N] and *Visflux* [5] [N] into *flux* [5] [N] through Eq. (10), NS solver would run LUSGS iteration module to obtain a new *DQ* [5] [N], as shown in Eq. (11).

$$flux[j][i] = Invisflux[j][i] + Visflux[j][i],\quad 0 \le i \le nTCell,\ 0 \le j < 5 \tag{10}$$

$$\begin{aligned} DQ[j][i] = LUSGS(&DQ[5][N], q[5][N], dt[5][N],\\ &flux[5][N]),\quad 0 \le i < nTCell,\ 0 \le j < 5 \end{aligned} \tag{11}$$

Above formulae briefly summarize the data relationships among the main variables occurred in the NS solver while the more specific knowledge is not presented here for due to the limited space and one could find those complex formulae, such as f_{grad} and f_{limit} from CFD books [12].

Those data relationships indicate that the computation of NS solver must follow the order shown in Fig. 1 since the later module needs the front one's data. However, this restriction of computational order may hinder the further performance improvement for MPI-CUDA CFD while the flowchart of the solver here solidifies into the cycle of the computation on GPUs, the MPI information transfer between GPUs and the host node and between the host nodes, and the computation on GPUs again. Those serial executions would bring the result that GPUs computing cores would keep inactive when *cudaMemcpy*, *MPI_Isend* and *MPI_Irecv* are in their busy time. This will cause a strong imbalance between communication units and computation cores and lead to an inefficiency of multi-GPUs platform.

Further exploration of above data relationships may lead to new discoveries. Actually, the most noteworthy feature of above data relationships is that the main variables, including of *q* [5] [N], *dqdx* [5] [N], *dqdy* [5] [N], *dqdz* [5] [N], *Limit* [5] [N], *flux* [5] [N], *DQ* [5] [N] and etc., only change their values in one certain range in CFD and then keep their values constant until the next nonlinear iteration of CFD enters into this range again. And another feature is that almost all variables to be computed in the flowchart of

CFD all have a direct dependency on *q* [5] [N]. Based on the Data-Centric analysis, the working range of computational modules in NS solver can be zoomed to some extent, where the solver can run at a different way to finish its communication and computation tasks. Firstly, the time step module to calculate the array *dt*[N] can be carried out at the range from the start of the NS solver to the end of the flux calculation. Secondly, the limit module is divided into two parts, *Limit*-1 and *Limit*-2, and the data relationships are shown in Eq. (7) and Eq. (8) respectively:

$$(qmax[j][i],\ qmin[j][i],\ esp[j][i]) = f(q[5][N]),\quad 0 \le i < nTCell,\ 0 \le j < 5 \tag{12}$$

$$\begin{aligned} Limit[j][i] = f(&qmax[5][N],\ qmin[5][N],\ esp[5][N],\ dqdx[5][N],\\ &dqdy[5][N],\ dqdz[5][N]),\quad 0 \le i < nTCell,\ 0 \le j < 5 \end{aligned} \tag{13}$$

Limit-1 part can be performed at larger range from the start of NS solver to the start of *Limit*-2 calculation and only depends on *q* [5] [N] data. The inviscid flux calculation can be processed in the same way. Thirdly, based on the Eq. (8), the temperature information can be calculated at the range from the start of the NS solver to the start of the viscous flux calculation. Finally, the gradient computation for each variable is independent with others and can be processed parallel. Those relationships constitute final Data-Centric data relationships.

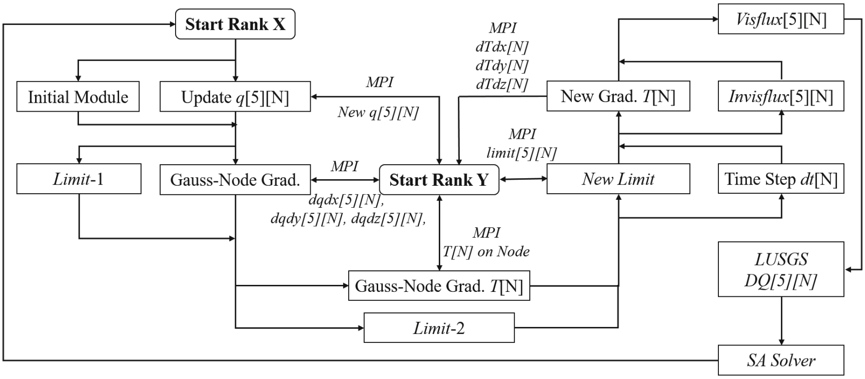


Fig. 3. Data-Centric Code Refactoring for NS Solver in CFD

The Data-Centric data relationships make it possible for the code refactoring for NS solver in CFD as shown in Fig. 3. Two MPI rank X and Y are depicted where the modules are designed to process parallel for the balance of communication and computation. For example, the first part of limit calculation, *Limit*-1, is shifted forward to carry out parallel with MPI transfer for the latest gradient values. In this way, the communication in MPI transfer module, including of *cudaMemcpy* and *MPI_Isend* and *MPI_Irecv* functions, would go parallel with the computation in *Limit*-1. With the multiple adjustment, MPI transfer process at the end of *q* [5] [N], *dqdx* [5] [N], *dqdy* [5] [N], *dqdz* [5] [N], *Limit* [5] [N], *dTdx*[N], *dTdy*[N] and *dTdz*[N] calculation all can be performed parallel with the computation running on GPUs. Another strategy of the code refactoring in NS solver is developed for the Node-based Green-Gauss gradient module for overlap of the MPI transfer of the ghost nodes. The computational stream would change from Stream Default to the stream allocated by the fluid flow variables and the communication in one fluid flow variable would be covered up by the computation from other fluid flow variables.

SA solver in CFD employs above method to develop Data-Centric data relationships and the corresponding code refactoring has been organized in the same way. Actually, the flowchart of SA solver is similar to the one in NS solver, while the solving variable changes from the fluid flow q [5] [N] to the single SA variable sa[N]. Due to the limit space this work will not go into details of the code refactoring in SA solver here.

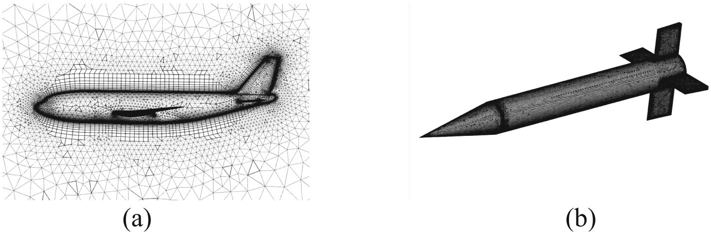
3.3 Dual-Mode API for CFD Solver

The program structure of heterogeneous parallel CFD can be generalized into following parts, where the geometric topology information refers to the mesh and the boundary conditions and the computational kernels stand for the main computational modules in CFD, such as the gradient and the limit and etc. The application API calling layer would determine the organization of solver and MPI information transfer among ranks would be accomplished by MPI kernels. Above Minimum-cost MPI Transfer Strategy would build a new set of MPI kernels besides the original one, while the code refactoring strategy would introduce a more efficient application API calling layer for multi-GPUs' implementation. The main body of CFD, including of the computational kernels and the geometric topology information will keep unchanged all the time. Therefore, two running API would be established in CFD where the original Model-Centric API is used to develop and debug new codes to expand the function of CFD while Data-Centric API is appropriate for high-performance execution.

4 Results and Discussion

In this section, numerical experiments will be carried out using industrial CFD software, FlowStar, to verify the optimization effect of above Data-Centric CFD models. FlowStar is a CFD framework that utilizes mathematical models and numerical algorithms to simulate and analyze fluid mechanics and heat transfer phenomena. It provides a comprehensive suite of tools for solving complex problems related to turbulence, multiphase flows, combustion, and more. And all above parallel models have been integrated into FlowStar. Two industrial cases, the full-aircraft CHN-T1 airplane [13] and the Army-Navy Basic Finner(ANF) missile [14], are selected to test the performance of different parallel CFD implementations. The geometry of CHN-T1 and ANF are presented in Fig. 4-a and Fig. 4-b respectively, and the mix-element unstructured grid is used to match the complex geometric profile in those models. Results are computed by the GPU server configured with 64 Intel Xeon 8268 CPU and 64 A100 80G GPUs.

The weak scaling test is carried out firstly. Since the practical industrial unstructured grid is rather than difficult to generate a set of grid in proportion, each GPUs card would handle with approximately equaling cells to test the weak scaling. Table 1 gives the configuration of the weak scaling test with a series of grids of CHN-T1, 6.5, 17.3, 49.4 and 162.2 million. Three versions of MPI-CUDA implementations are performed, where the primitive refers to the one developed from Model-Centric CFD codes. The time costs for the first 100 iterations are recorded to compare the performance difference. Results show that with the increase of the computing GPUs cards and keeping the constant computing load on each GPUs, the difference among time-consuming results for a certain MPI-CUDA strategy, like the code refactoring, was rather small, which proved the fine weak scaling for the MPI-CUDA implementation. What's more, those results among different MPI-CUDA implementations show that the minimum-cost MPI strategy could improve about 2.0 times performance compared with the primitive one and the code refactoring can further bring 40% to 50% performance improvement compared with the minimum-cost model. For example, the 6.5 million CHN-T1 computed on 2 A100 nodes and the 162.2 million on 52 A100 nodes would spend almost the same time, 32.2 ms, for the first 100 iterations and the Data-Centric optimizations can decrease the running time dramatically, from about 95 ms to 45 ms and further 32 ms.


(a)

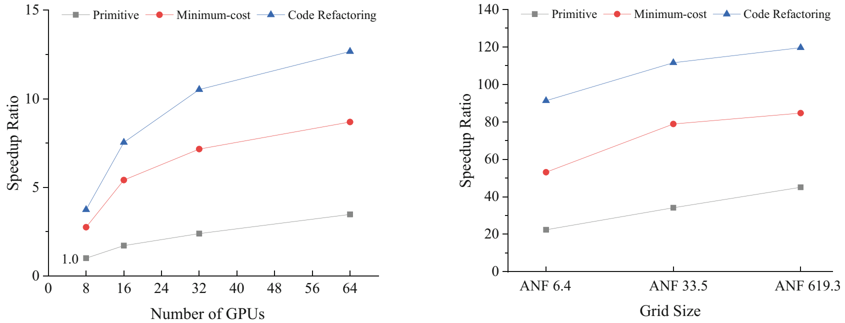
(b)

Fig. 4. Geometry and Grid for CHN-T1 aircraft and ANF missile

Table 1. Weak Scaling Test with Different Strategies on CHN-T1 Grids (100 iterations)

GPUs Numbers	2	6	16	52
CHN-T1 Grid/million	6.5	17.3	49.4	162.2
Cells per GPUs/million	3.25	2.88	3.08	3.11
Primitive MPI-CUDA(ms)	95.27	82.46	92.43	97.51
Minimum-Cost MPI Strategy(ms)	46.51	41.37	50.42	45.75
Code Refactoring Strategy(ms)	32.18	28.34	33.19	32.22

The CHN-T1 case with 162.2 million grid is employed to carry out the strong scaling test with the range of 8 to 64 A100 nodes. The speedup results are presented in Fig. 5 where the performance benchmark is set to the primitive MPI-CUDA running on 8 A100 nodes. The results on 64 A100 node show that for the primitive MPI-CUDA implementations, only the 46% parallel efficiency could be obtained compared with the benchmark on 8 A100 nodes. This poor scalability could be ameliorated by the Data-Centric models. For example, the Data-Centric MPI-CUDA version on 64 A100 nodes can increase the parallel efficiency to 158% compared with the benchmark or the primitive MPI-CUDA running on 8 A100 nodes. In other words, the Data-Centric optimization could bring the super linear acceleration over the primitive MPI-CUDA implementation. Meantime, the performance improvement in the implementation of the minimum-cost strategy and the code refactoring model also can be observed from the results presented in Fig. 5 (a).



(a) (b)

Fig. 5. (a) Strong Scaling Test with Different Strategies on CHN-T1 162.2 million Grids and (b) Speedup Ratio of MPI-CUDA Strategies Compared with MPI Implementation on Different Grid Sizes

After the assessment of the scalability of the Data-Centric optimization, various parallel versions are employed to the large-scale test with the ANF grid. Figure 5 (b) presents the speedup ratio of the ANF case on 64 A100 nodes with three set of ANF grids, including of 6.4, 33.5 and 619.3 million cells, compared to the original MPI parallel implementation with 64 ranks running on Intel Xeon 8268 CPU. The results show that with the increase of grid cells computed on GPUs, the application could obtain a higher speedup ratio. And the maximum speedup ratio could be found in ANF case with 619.3 million cells, which is about three times parallel performance of the primitive MPI-CUDA implementation.

5 Conclusion

This paper introduces a Data-Centric approach to develop efficient and scalable MPI-CUDA CFD applications on large-scale multi-GPUs platforms. The Data-Centric approach includes Minimum-cost MPI strategy and the code refactoring technique for a better balance between communication and computation. The industrial unstructured CFD software, FlowStar, is employed to verify the performance. Results exhibit impressive speedup improvements and better scalability of CFD codes when compared to traditional Model-Centric CFD implementations. Tests show that Minimum-cost MPI strategy can bring about 2.0 times performance improvement over the Model-Centric MPI-CUDA CFD and the code refactoring technique can further take 40% to 50% faster acceleration than the one based on the minimum-cost MPI strategy. The large-scale test displays that Data-Centric implementation on 64 A100 GPUs platform could produce over 120 speedup ratio compared to the original MPI parallel implementation with 64 ranks and 3 times faster than the Model-Centric MPI-CUDA one.

Acknowledgements. This work was supported by Sichuan Science and Technology Program (Grant No. 2023YFG0152), National Numerical Wind Tunnel Project, and the National Key Research and Development Program of China (Grant No. 2020YFB1506700).

References

1. Heldens S., Hijma P., Werkhoven B.V.: the landscape of exascale research. ACM Comput. Surv. (CSUR) **53**, 1–43 (2020)
2. Afzal, A., Ansari, Z., Faizabadi, A.R.: Parallelization strategies for computational fluid dynamics software: state of the art review. Arch. Comput. Methods Eng. **24**, 337–363 (2017)
3. Kedward, L, Allen, C.B.: Summary of investigations into finite volume methods on GPUs. In: AIAA SCITECH 2022 Forum, vol. 0028 (2022)
4. Zhang, J., Dai, Z., Li, R., Deng, L., Liu, J., Zhou, N.: Acceleration of a production-level unstructured grid finite volume CFD code on GPU. Appl. Sci. **13**(10), 6193 (2023)
5. Zhang, J., Deng, L., Li, R., Li, M., Liu, J., Dai, Z.: Achieving high performance and portable parallel GMRES algorithm for compressible flow simulations on unstructured grids. J. Supercomput. **79**, 1–25 (2023)
6. Hashimoto, T., Yasuda, T., Tanno, I.: Multi-GPU parallel computation of unsteady incompressible flows using kinetically reduced local navier-stokes equations. Comput. Fluids **167**, 215–220 (2018)
7. Lei, J., Li, D., Zhou, Y.: Optimization and acceleration of flow simulations for CFD on CPU/GPU architecture. J. Braz. Soc. Mech. Sci. Eng. **41**, 1–15 (2019)
8. Gomes P., Economon T.D., Palacios R.: Sustainable high-performance optimizations in SU2. In: AIAA Scitech 2021 Forum, vol. 0855 (2021)
9. Ziogas, A.N., Ben-Nun, T., Fernández, G.I.: A data-centric approach to extreme-scale ab initio dissipative quantum transport simulations. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–13 (2019)
10. Ben-Nun T., de Fine Licht J., Ziogas A. N.: Stateful dataflow multigraphs: A data-centric model for performance portability on heterogeneous architectures. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–14, (2019)
11. Ziogas A.N., Ben-Nun T., Fernández G.I.: Optimizing the data movement in quantum transport simulations via data-centric parallel programming. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–17 (2019)
12. Blazek J.: Computational Fluid Dynamics: Principles and Applications. Butterworth-Heinemann, Oxford (2015)
13. Yu, Y.G., Zhou, Z., Huang, J.T.: Aerodynamic design of a standard model CHN-T1 for single-aisle passenger aircraft. Acta Aerodynamica Sinica **3**, 505–513 (2018)
14. Bhagwandin, V.A., Sahu, J.: Numerical prediction of pitch damping stability derivatives for finned projectiles. J. Spacecr. Rocket.Spacecr. Rocket. **51**(5), 1603–1618 (2014)

Research on Psychological Testing Methods of Criminal Suspects Based on Multi-features of EEG

Yijie Peng and Xiaofan Zhao(✉)

School of Cybersecurity, People's Public Security University of China, Beijing 100038, China
zhaoxiaofan@ppsuc.edu.cn

Abstract. P300 is a commonly used indicator for testing the psychology of criminal suspects, but it has problems such as weak signal and large amount of processed data. Aiming at such problems, based on experiments to simulate real case data, a psychological test method for criminal suspects based on multi-feature extraction of EEG signals in time domain, frequency domain, and time-frequency domain was proposed. In order to achieve psychological testing of criminal suspects in public security investigations, used the existing data to test and adjusted the model. In the time domain, the signal-to-noise ratio was improved by superimposing and averaging, and P300 was extracted. The amplitude and latency of the components were taken as the time domain features. In the frequency domain, the relationship between the EEG power and frequency reflected by the power spectrum estimation was used as the frequency domain features. In the time and frequency domain, the wavelet approximation coefficients of the corresponding frequency band extracted by the Mallat algorithm was used as the frequency domain features. Time-frequency domain features were selected through F-score. Finally, SVM was used as the classifier. The optimal penalty factor and kernel function were selected through cross-validation and dynamic grid. The results show that the method of multi-feature extraction can reflect the essential characteristics of the suspect's EEG signal, reduce the amount of data processing, and have a higher classification accuracy.

Keywords: P300 · feature extraction · ICA · Mallat · F-score · SVM

1 Introduction

The P300 is an EEG indicator that is commonly used in psychological testing techniques for criminal suspects. The P300 is an event-related potential (ERP), an evoked potential associated with human attention to events. It reveals the subject's response to an external stimulus with a relatively low probability of occurrence, and is called P300 because it generally occurs about 300 ms after the stimulus occurs [1].

Fund Project: Double First-Class Innovation Research Project for People's Public Security University of China (No.2023SYL07).

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 105–117, 2024.
https://doi.org/10.1007/978-981-99-8211-0_11

It was shown that P300 was not influenced by the physical properties of the stimulus, but related to the frequency of the stimulus, the meaning of the information contained in the stimulus, the subject's expectations, and therefore can be used as an important psychometric test for criminal suspects indicator [2]. In P300-based tests, researchers select photographs of crime scenes, crime tools, and other case-related objects as detection stimuli [3]. Since P300 and spontaneous EEG signals are slightly different in terms of generation and characteristics, they both have bioelectric signal characteristics and can be studied using the same signal processing methods.

Therefore, this paper is based on the EEG signal data processing method. With the help of machine learning techniques, based on laboratory simulated case data, we study and construct a new rapid detection of deception features based on brain cognitive potential and adjust the model by applying the existing data to test and to realize the psychological testing of criminal suspects in public security investigation, which is useful for This is of great theoretical significance and practical value to prevent and combat crimes, and to quickly identify suspects in unexpected situations. This has important theoretical significance and practical value for preventing and combating crime, and for rapid identification of suspects in unexpected situations.

2 Related Work

In the research related to the psychological testing of criminal suspects, feature extraction is a crucial aspect, and scholars have proposed many methods for EEG signal feature extraction methods, which are usually divided into the following the methods are usually divided into the following aspects:

Time Domain Feature Extraction. Geometric parameters of the time domain waveform of EEG signals, such as peak, wave area, latency, and signal energy, are often extracted. For example, Liang Zhou et al. [4] used wave amplitude, wave area and latency as P300 features for lie detection experiments, and Junfeng Gao et al. [5] used F-score to select time-domain features and found that amplitude and peak difference had a more positive role in classification.

Frequency Domain Feature Extraction. The EEG signal has strong frequency characteristics, and its frequency information can be extracted as features. The frequently used extraction methods are Fourier transform and power spectrum estimation. Mu et al. [6] have used AR model for power spectrum estimation to achieve frequency domain feature extraction of EEG signals.

Time-Frequency Domain Feature Extraction. EEG signals are non-stationary random signal, its transient changes contain both time domain and frequency domain information, if only EEG signals are non-stationary random signals, and the transient changes contain both time and frequency domain information. Therefore, the combined time-frequency feature analysis method is often used. The commonly used The wavelet transform algorithm, wavelet packet decomposition transformation, Hilbert yellow transform algorithm and other methods are often used to extract time-frequency features. The wavelet transform algorithm, wavelet packet decomposition transform, Hilbert yellow transform algorithm, etc. are often used to extract time-frequency features. Yang L. C. [7]

proposed P300 recognition algorithm based on wavelet decomposition and support vector machine, and Wu T. et al. [8] proposed the spontaneous EEG signal feature extraction based on EMD and Hilbert transform method.

In addition, there are many feature extraction methods, and A. Moura [9] concluded that different methods should be chosen in different situations by comparing the advantages and disadvantages of several of the most popular EEG signal feature extraction methods today.

Based on the above research, this paper proposes a multi-feature extraction method using P300 as the basic basis of the psychological test for criminal suspects. The wave amplitude and latency of P300 are used as time domain features, and the power spectrum estimation and Mallat algorithm are used to extract the frequency domain features and time-frequency domain features. The existing features are combined into a time-frequency-time-frequency fusion feature vector, and the SVM is used as the classifier to select the optimal penalty factor and kernel function by cross-validation and dynamic grid. Finally, the effectiveness of the classification model in suspect identification is verified based on the existing EEG data. The overall model architecture is shown in Fig. 1.

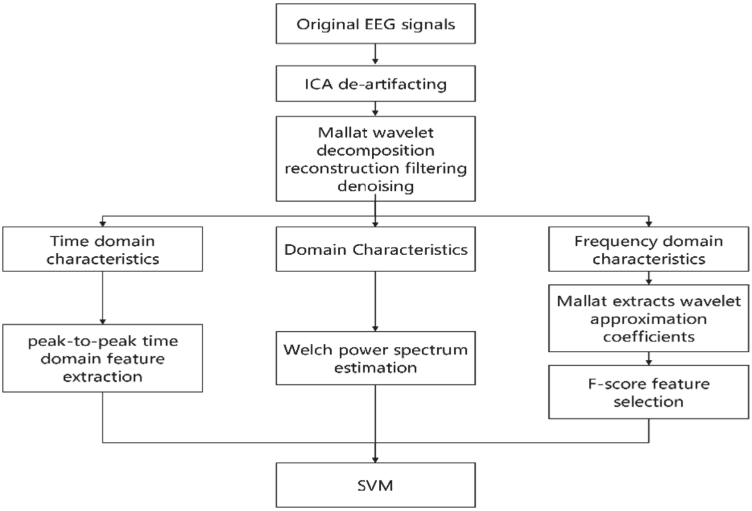

Original EEG signals
ICA de-artifacting
Mallat wavelet decomposition reconstruction filtering denoising
Time domain characteristics
Domain Characteristics
Frequency domain characteristics
peak-to-peak time domain feature extraction
Welch power spectrum estimation
Mallat extracts wavelet approximation coefficients
F-score feature selection
SVM

Fig. 1. Model architecture diagram.

3 Psychological Testing for Criminal Suspects

In this paper, the study of psychological testing methods for criminal suspects was developed in the form of a simulated crime [10]. All subjects in the test were randomly divided into three groups. Subjects in the innocent group took the memory test directly. Subjects in the informed group were asked to determine the correct answers to the four topics by means of questioning and subjects to ensure that the subjects in that group were indeed informed. Subjects in the implementation group performed a simulated crime before the test began. The mock crime was conducted in an elaborate room, where the subject was instructed to enter the room and steal something he or she thought was valuable, without being told what the item would be. After the subjects had "stolen" the items, they were given an EEG test.

Subjects first read and signed an informed consent form, after which the test procedure was explained to them. There were four sets (blocks) of tests with three breaks. The first and second sets of tests were about what the stolen item was (envelope) and what was inside the stolen item (check), while the third and fourth sets of tests were about the value of the stolen item ($800) and the location of the stolen item (in the drawer). Each set of tests began with a prompt about the problem involved in that set of words. The stimulus sequences were referenced to the DPCTP paradigm [11].

EEG data were acquired using Neuroscan's 64-conductor ERP recording system, with electrode positions referenced to the international 10–20 system. Horizontal electrooculography (HEOG) was recorded with an electrode located 1.5 cm lateral to the left and right orbits, and vertical electrooculography (VEOG) was recorded with an electrode located approximately 1 cm superior and inferior to the left eye. The average electrode of the bilateral mastoid was used as the reference electrode, and the midline of the scalp was grounded between Fz and Cz. The scalp resistance at each electrode was adjusted to less than 5 kΩ. The signal was amplified by AC, and continuous EEG and EOG were recorded with an A/D sampling frequency of 1000 Hz (Fig. 2).

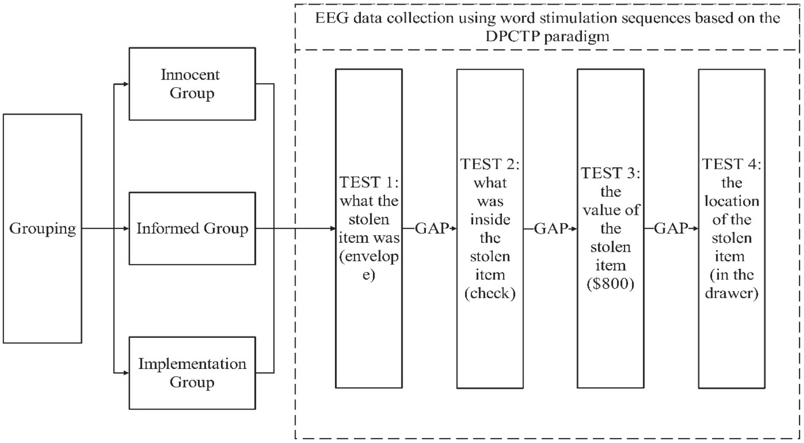

EEG data collection using word stimulation sequences based on the DPCTP paradigm
Grouping
Innocent Group
Informed Group
Implementation Group
TEST 1: what the stolen item was (envelop e)
GAP
TEST 2: what was inside the stolen item (check)
GAP
TEST 3: the value of the stolen item ($800)
GAP
TEST 4: the location of the stolen item (in the drawer)

Fig. 2. Simulated Criminal Mind Test.

4 Data Pre-processing Methods

4.1 ICA De-artifacting

In this paper, the artifacts are separated from the EEG signal by the signal separation algorithm-Independent Component Correlation Algorithm (ICA) [12]. The linear hybrid ICA model is shown in Fig. 3.

Assume that there are M source signals and N observed signals can be obtained through N electrodes, i.e., N leads, each of which is a linear combination of M source signals. For simplicity, assume that M = N, i.e., the number of electrodes and the number of dissociated source signals are the same. In matrix terms, let $S = (S_1, S_2, \ldots, S_M)^T$, $X = (X_1, X_2, \ldots, X_M)^T$, A be the mixing matrix, then X = AS.ICA is to find the transformation matrix B, also known as the unmixing matrix, and the N-dimensional output vector Y = WX = WAS is obtained by linear transformation of X.

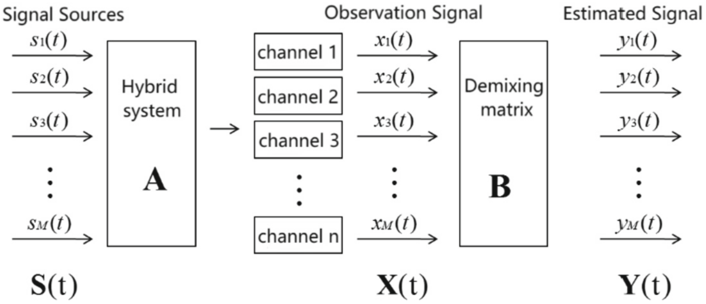


Fig. 3. Linear mixed ICA model.

4.2 Wavelet Transform Filtering and Denoising

In this paper, the Mallat algorithm [13, 14] is used to complete the decomposition of the EEG signal. Assuming that the signal to be decomposed is, the finite N-layer decomposition shown in Eq. (1):

$$\begin{aligned} x(t) &= A_1 + D_1 \\ &= A_2 + D_2 + D_1 = A_N + \sum\nolimits_{j-1}^{N} D_j \end{aligned} \tag{1}$$

where A_N is the approximate component of the low-frequency part of the signal, and D_j is the detail component of the high-frequency part under the j-layer decomposition.

The detail and approximate components of the corresponding frequency bands can be obtained through layer-by-layer decomposition, and each frequency band has its own wavelet coefficients, i.e., the detail coefficient corresponding to the detail component D1 is cD1, and the approximate coefficient corresponding to the approximate component A1 is cA1. The decomposition and reconstruction of the signal x(t) can be completed by processing the wavelet coefficients of each layer after decomposition. The single-step decomposition process is shown in Fig. 4.

Mallat algorithm signal decomposition formula [15]:

$$c_{jk} = \sum\nolimits_n h_0(n - 2k)c_{j-1,n} \tag{2}$$

$$d_{jk} = \sum\nolimits_n h_1(n - 2k)c_{j-1,n} \tag{3}$$

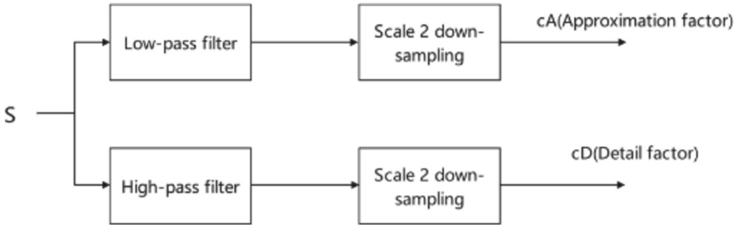


Fig. 4. Single-step decomposition process of wavelet decomposition.

In the equation, c_{jk}, d_{jk} are the signal scale coefficients and wavelet coefficients, respectively, and j is the number of signal decomposition layers.

The Mallat reconstruction formula is [15]:

$$c_{j-1,k} = \sum\nolimits_n h_0(k-2n)c_{j,n} + \sum\nolimits_n h_1(k-2n)d_{jn} \tag{4}$$

5 Feature Extraction Methods

5.1 Peak-To-Peak Time Domain Feature Extraction

The peak-to-peak is used to extract the amplitude and latency of P300 by searching for the most positive 100 ms on average from 300 ms to 900 ms, and the midpoint of this 100 ms determines the latency of P300. The distance between the most positive 100ms and the most negative 100 ms is the p-p wave amplitude of P300 [16].

Due to the low signal-to-noise ratio and high randomness of the EEG signal, the P300 signal is not easily determined by the influence of adjacent stimuli. In order to reduce the random noise and improve the signal-to-noise ratio, the EEG signal can be processed by superimposed averaging method before using peak-to-peak extraction of wave amplitude and latency.

Assume that the EEG signal generated in a single session is:

$$\begin{gathered} x(n,i) = p(n,i) + u(n,i) \\ i = 1, 2, \ldots, N \end{gathered} \tag{5}$$

N is the total number of stimuli, $x(n, i)$ is the original mixed EEG signal with power P, $p(n, i)$ is the pure EEG signal containing P300, and $u(n, i)$ is the noise with variance σ^2, mean 0; the signal-to-noise ratio of a single stimulus is P/.

Averaged over N superpositions:

$$\begin{gathered} \frac{1}{N}\sum\nolimits_{i=1}^{N} x(n,i) = \frac{1}{N}\sum\nolimits_{i=1}^{N} p(n,i) + \frac{1}{N}\sum\nolimits_{i=1}^{N} u(n,i) \\ = p(n) + \frac{1}{N}\sum\nolimits_{i=1}^{N} u(n,i) \end{gathered} \tag{6}$$

$p(n)$ is the EEG signal containing P300 averaged over N times. The power of $p(n)$ is still P after superposition averaging, and the variance of the noise becomes σ^2/N, and the power signal-to-noise ratio is N · P/, which improves the signal-to-noise ratio by a factor of N [14].

5.2 Welch Power Spectrum Estimation for Extracting Frequency Domain Features

The Welch algorithm minimizes the variance of the spectral estimate without affecting the resolution by segmenting the data and adding windows [17].

Based on the probability statistical theory, it is known that if the data of length N is divided into L segments, the length of each segment is M = N / L, and each segment is independent of each other, the estimated variance will be only 1/ L before the segmentation, which can be consistently estimated [18]. The algorithm is given in Eq. (7):

$$G_M^L(\omega) = \sum\nolimits_{l=1}^{L} G^{M,l}(\omega)/L \tag{7}$$

Finally, the power spectrum is calculated for each segment of the data and the results are averaged for each segment.

5.3 Mallat Extracts Time-Frequency Domain Features

Due to the complexity of EEG signals, the essential features of EEG signals cannot be obtained from time domain or frequency domain alone, and the features can be extracted by using time-frequency analysis method combining time domain and frequency domain. The wavelet transform has good time-frequency properties, and the time-frequency information of the effective frequency band can be obtained after the signal decomposition by Mallat algorithm, and the wavelet approximation coefficients and detail coefficients calculated by the decomposition can be used as time-frequency domain features [19]. Among them, the wavelet approximation coefficients respond to the low-frequency information and the wavelet detail coefficients respond to the high-frequency information.

Mallat is decomposed at each layer to obtain the wavelet coefficients corresponding to the detail and approximation components of the corresponding frequency bands: the detail coefficients $\mathrm{cD_j}$ and the approximation coefficients $\mathrm{cA_j}$, where *j* is the number of corresponding layers (Fig. 5).

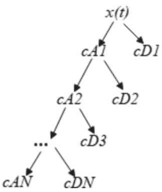


Fig. 5. Wavelet decomposition.

6 EEG Signal Processing Process

6.1 Pre-processing

In this paper, the experimental environment is Matlab2016b, and in addition, EEGLAB, ERPLAB, LIBSVM and other toolboxes are installed for EEG data processing and classification. The main steps of data pre-processing are segmentation, artifact removal, filtering and noise removal.

According to the principle of P300 applied to the psychological test for criminal suspects, the EEG signals generated by the suspect and the innocent person in response to the probe stimulus can be analyzed to distinguish the two. Therefore, by extracting the EEG signals of subjects in response to the probe stimuli in segments, 100 epochs can be extracted from a single subject per experiment.

The process of artifact removal by ICA is as follows: first find the unmixing matrix B to complete the separation of independent signals; identify the artifact components; set the rows representing the artifact components in the output signal (independent source estimation signal) Y to zero and correct Y to Y'; correct the observed signal X' = AY' to obtain a relatively clean EEG signal.

As shown in Fig. 6, the artifacts of the blue EEG signal were well corrected by the above operation well corrected by the above operation.

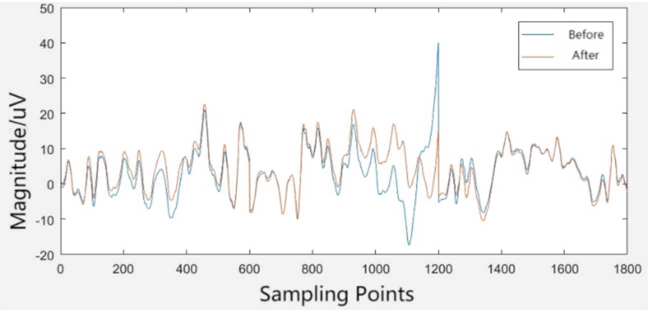


Fig. 6. Artifact removal

After removing the artifacts, the signal is filtered and denoised using Mallat algorithm for wavelet decomposition and reconstruction.

The data in this paper are sampled at a reduced sampling rate of 500 Hz, and the highest frequency of the EEG signal is 250 Hz. The general frequency range of P300 is 1–10 Hz [1], and the signal needs to be decomposed in 5 layers to reach this frequency range. 5 layers of decomposition are in the frequency range of 0–7.8125 Hz. As shown in Fig. 7.

The reconstruction is performed based on the obtained layer 5 approximation coefficients, and the same time-domain superposition averaging process is applied to the obtained reconstructed signal. As shown in Fig. 8, it can be seen that a good filtering and denoising effect is achieved after wavelet transform processing.

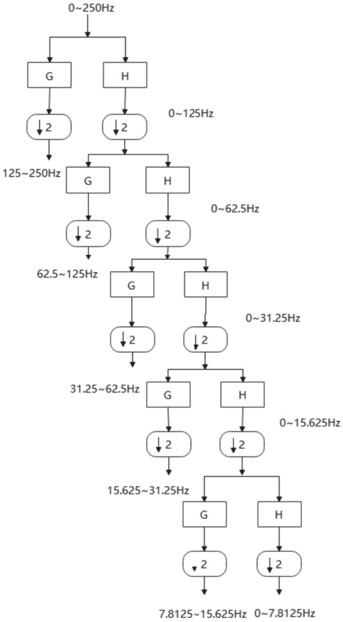


Fig. 7. EEG signal decomposition process

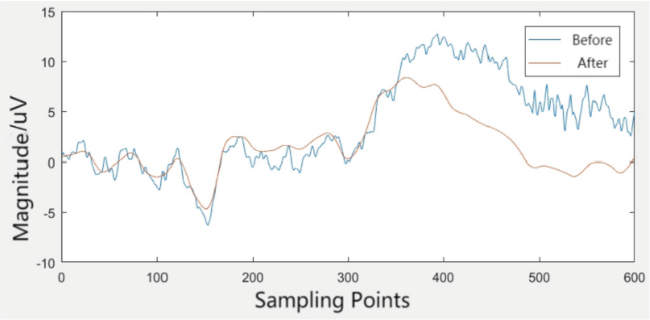


Fig. 8. Denoising results of wavelet decomposition and reconstruction

6.2 Multi-feature Extraction

According to the above, the segmentation process has been completed in the preprocessing. Every 10 identical detection stimulus responses of each subject are superimposed and averaged in 1s length. In this paper, we use 41 sets of experimental data, and each experiment generates 100 epochs for a single subject, and 410 probe stimulus waveforms are obtained after superimposed averaging.

It has been shown that not all electrodes induce significant P300 components. In addition, the amount of data to be processed can be appropriately reduced by electrode selection. In this paper, the PZ, FZ, and CZ electrodes that induce the P300 component are selected by comparison. The wave amplitude and latency of P300 were extracted by peak-to-peak method.

Welch power spectrum estimation of EEG signals from the FZ channel. Figure 9 show the power spectrum estimates of EEG signals with and without P300 components, respectively. It can be seen from the coordinate plots that the power spectrum estimates of EEG signals with P300 components are significantly larger than those of EEG signals without P300 components in the low frequency band, and the low frequency power spectrum density values are extracted as the frequency domain features.

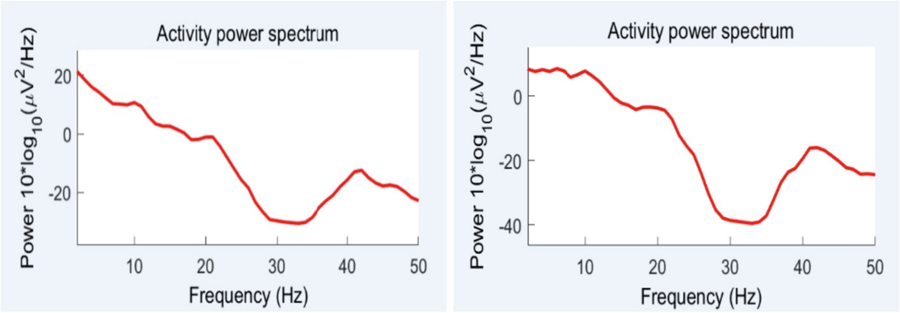


Fig. 9. Estimation of suspect power spectrum (left) and Innocent power spectrum estimation(right)

The general frequency range of P300 is from 1 to 10 Hz, which corresponds to the frequency band in which the approximation coefficients are located in the fifth layer wavelet preprocessing. The EEG signal was subjected to a 5-layer wavelet transform during preprocessing, and the segmentation of the EEG was completed before the wavelet transform, so the 5th layer approximation coefficients cA5 of each epoch could be extracted and averaged. Each subject can obtain 31 approximate coefficients as time-frequency domain features. However, the number of these 31 approximate coefficients is large and not all of them are favorable for classification, so the F-score method is used for feature selection.

F-score is a category separability evaluation index based on intra-class spacing, mainly used for dichotomous classification. The essence is to select the validity features with small intra-class variation and large inter-class variation. The F-score value of the i-th feature is calculated as:

$$\begin{gathered}\mathrm{F}_i = \left(x_i^+ - \bar{x}_i\right)^2 + \left(x_i^- - \bar{x}_i\right)^2 / \\ \left[\frac{1}{N_+ - 1}\sum_{k=1}^{N_+}\left(x_{k,i}^+ - \bar{x}_i^+\right)^2 + \right. \\ \left. \frac{1}{N_- - 1}\sum_{k=1}^{N_-}\left(x_{k,i}^- - \bar{x}_i^-\right)^2\right]\end{gathered} \tag{8}$$

N_+ and N_- are the sample numbers of positive and negative classes, $\bar{x}_i$, $\bar{x}_i^+$, $\bar{x}_i^-$ are the average values on the whole dataset, on the positive class dataset, and on the negative class dataset, respectively, and $x_{k,i}^+$ and $x_{k,i}^-$ are the feature values of the i-th feature of the k-th positive class sample point and the negative class sample point, respectively.A larger F$_i$ indicates that the feature has a stronger discriminative power [7].

The 12 features were extracted by F-score feature selection, as shown in Fig. 10. The cA5 approximation coefficients with and without the P300 component can reflect the difference to some extent, and are used as the time-frequency domain features of the EEG signal.

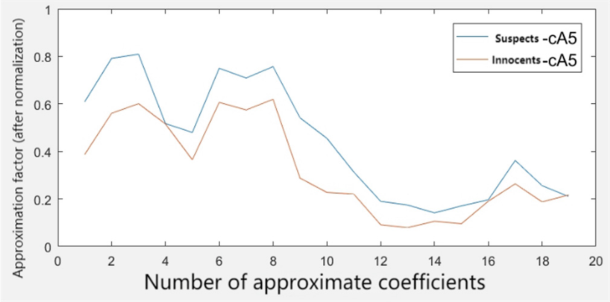


Fig. 10. Approximation coefficient

After extracting the features in time domain, frequency domain, and time-frequency domain and filtering the F-score features, each multi-feature set has 23-dimensional features, including 6-dimensional time domain features, 5-dimensional frequency domain features, and 12-dimensional time-frequency domain features.

6.3 SVM-Based Classification

In this paper, SVM algorithm is used to classify the extracted features, and the algorithm is implemented based on LIBSVM toolbox. Since the informants theoretically belong to the innocents, this paper divides the three groups of subjects into two categories, the innocent group and the informed group belong to the innocent category, and the experimental group belongs to the suspect human. The Radial Basis Function (RBF) [20] is chosen as the kernel function with low complexity and simple operation. The feature values were normalized to between 0 and 1 before classification. The SVM classification model was trained with a combination of a penalty factor c and a kernel parameter σ to optimize the performance of the SVM. 10-fold cross-validation and dynamic grid search were used, and the initial values of the grid search parameters were 2^{-4}, the termination values of the parameters were 2^4, and the step size was set to 0.1. Finally, the total classification accuracy was 83.4146%. The accuracy of suspect classification is 87.5%.

7 Summary

In this paper, we propose a multi-feature extraction method, in which the wave amplitude and latency of P300 are used as the time domain features, the power spectrum estimation is used as the frequency domain features, and the approximate coefficients after wavelet decomposition are used as the time-frequency domain features in the time-frequency domain. Through multi-feature extraction, the P300 essential features are comprehensively reflected, and the data volume is greatly reduced to guarantee the classification effect of SVM. In addition, in the process of data processing, this paper adopts the methods of denoising and improving the signal-to-noise ratio, such as ICA to remove eye movement artifacts, wavelet decomposition reconstruction, and superposition averaging, etc., which makes the features of P300 more prominent and improves the classification accuracy.

References

1. Wang, P., Shen, J., Shi, J.: P300 feature extraction algorithm basedon wavelet transform and temporal energy entropy. Chin. J. Sci. Instrum. **32**(06), 1284-1289 (2011). (in Chinese)
2. Fengjuan, R.: Research on Polygraph Algorithm Based on P300. Shaanxi Normal University (2014). (in Chinese)
3. Pu, X.: Experimental Research on P300 Polygraph. Zhejiang Normal University (2006). (in Chinese)
4. Liang, Z., Yang, W., Liao, S., Zou, H.: An experimental study on the application of P300 in simulated theft lie detection. Chin.J. Clin. Psychol. (01), 34–36 (1999). (in Chinese)
5. Gao, J., Tian, H., Yang, Y., et al.: A novel algorithm to enhance P300 in single trials: application to lie detection using F-score and SVM. PLoS ONE **9**(11), e109700 (2014)
6. Mu, Z., Hu, J.: Research of EEG identification computing based on AR model. In: 2009 International Conference on Future BioMedical Information Engineering (FBIE), Sanya, pp. 366–368 (2009)
7. Yang, L., Li, J., Yao, Y., Wu, X.: A P300 detection algorithm based on f-score feature selection and support vector machines. J. Biomed. Eng. **25**(01), 23–26+52 (2008). (in Chinese)
8. Ting, W., Guozheng, Y., Bingfeng, Q.: EEG feature extraction based on empirical mode decomposition and hilbert transform in brain computer interface. Beijing Biomed. Eng. **30**(04), 381–386 (2011). (in Chinese)
9. Moura, A., Lopez, S., Obeid, I., et al.: A comparison of feature extraction methods for EEG signals. In: 2015 IEEE Signal Processing in Medicine andBiology Symposium (SPMB), Philadelphia, PA, pp. 1–2 (2015)
10. Winograd, M.R., Rosenfeld, J.P.: Mock crime application of the complex trial protocol (CTP) P300-based concealed information test. Psychophysiology **48**(2), 155–161 (2011)
11. Labkovsky, E., Peter Rosenfeld, J.: A novel dual probe complex trial protocol for detection of concealed information. Psychophysiology **51**(11), 1122–1130 (2014)
12. Bing, Y.: Research on De-artifacting of Brain Wave Signal. Nanjing University of Posts and Telecommunications (2014). (in Chinese)
13. Wang, H.: Research on Denoising Algorithm for Multi-channel EEG Signal. Changchun University of Science and Technology (2020). (in Chinese)
14. Yan, M.: Research on Multi-domain Fusion Technology Based on P300-EEG Classification. Changchun University of Science and Technology (2020) (in Chinese)
15. Zhong, L., Wei, G.: Wavelet decomposition and reconstruction denoising based on the mallat algorithm. Electron. Des. Eng. **20**(02), 57–59 (2012). (in Chinese)
16. Meijer, E.H., Smulders, F.T., Merckelbach, H.L., Wolf, A.G.: The P300 is sensitive to concealed face recognition. Int.J. Psychophysiol. **66**(3), 231–237 (2007)
17. Xiaoqing, X., Genmin, Z.: Window function selection and algorithm analysis in Welch power spectrum estimation. Comput. Age **02**, 1–4 (2018). (in Chinese)

18. Luo, M., Liu, S.: Realization of power spectrum estimation based on Welch algorithm. J. Beijing Technol. Bus. Univ. (Nat. Sci. Edn) (03), 58–59+66 (2007). (in Chinese)
19. Wang, P.: Feature extraction of EEG signal based on wavelet transform and multi-domain fusion. Zhejiang University (2011). (in Chinese)
20. Wang, Z.: Research on P300 Signal Recognition Technology in BCI System Based on SVM. Tianjin University (2007). (in Chinese)

Insider Trading Detection Algorithm in Industrial Chain Based on Logistics Time Interval Characteristics

Fulin Chen[1], Kai Di[2(✉)], Hansi Tao[2], Yuanshuang Jiang[2], and Pan Li[1]

[1] School of Cyber Science and Engineering, Southeast University, Nanjing 211189, Jiangsu, China
[2] School of Computer Science and Engineering, Southeast University, Nanjing 211189, Jiangsu, China
dikai@seu.edu.cn

Abstract. Insider trading behavior is becoming increasingly prevalent with the rapid development of the industrial chain. Insider trading refers to the illegal behavior of conducting insider trading by obtaining insider information. The existing insider trading detection methods of industrial chain do not consider the problems of inefficient industrial chain data characteristics and long trading time span, resulting in poor algorithm effect. Therefore, in order to solve the above problems, this paper proposes an algorithm for detecting insider trading in the industrial chain based on logistics time interval characteristics. Firstly, aiming at the problem of inefficiency of industrial chain data characteristics, this algorithm proposes a logistics index construction method for describing the whole process of insider trading behavior; Secondly, aiming at the problem of long time span of transaction, a dynamic sliding window method is proposed; Finally, the isolation forest algorithm is improved to identify the abnormal data. Verified under the real data set, the results show that compared to using the isolation forest methods, the F1 value of the insider trading behavior detection problem of the industry chain can be improved by 20.68% by using the logistics time interval feature.

Keywords: Industrial chain · Insider trading detection · Anomaly detection

1 Introduction

With the development of the global supply chain, the safe and stable operation of the industrial chain plays a vital role in the economic development of a country. At present, the environment of the industrial chain trading market is complex, and there are various trading risks [1]. Insider trading behavior is one of the risks. Insider trading refers

This work was supported by the National Key Research and Development Program of China (No. 2022YFB3304400), the National Natural Science Foundation of China (Nos. 62076060, 61932007), the Key Research and Development Program of Jiangsu Province of China (No. BE2022157), and the Defense Industrial Technology Development Program (JCKY2021214B002).

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 118–129, 2024.
https://doi.org/10.1007/978-981-99-8211-0_12

to traders directly or indirectly using inside information to buy and sell commodities and obtain improper economic benefits. Inside information refers to the non-public information obtained by the staff of some financial institutions or regulatory departments of the industry chain due to the convenience of their position or position. Since insider information can affect the price in the trading market, users who have access to insider information can spread it to others [2], thereby indirectly obtaining excess profits. This behavior affects the fairness of the industrial chain trading market and threatens the healthy development of the industrial chain trading market. In the supervision process of insider trading, it shows the characteristics of strong concealment of insider trading [3]. Therefore, this paper designs an insider trading behavior detection algorithm of industrial chain based on the characteristics of logistics time interval to mine the hidden insider trading behavior.

At present, the detection of insider trading behavior of industrial chain usually uses the regularized abnormal indicators of dealer trading patterns and the data-driven model methods [4] to identify the abnormal behavior of dealers. For the methods of regularized abnormal indicators, due to the characteristics of physical delivery of the industrial chain, commodities can be traded not only online but also offline, so the whole process of circulation of commodities cannot be supervised, which leads to the inapplicable income measurement indicators based on informed trading. For the data-driven model methods, the direct use of logistics features has a high dimension and the model learning is difficult, which makes the detection effect of insider trading behavior not good. Therefore, in order to help regulators efficiently supervise the insider trading behavior in the industrial chain trading market, this paper studied the problem of insider trading behavior detection based on logistics characteristics in the industrial chain trading market.

The main contributions of this paper are summarized as follows:

1) Firstly, aiming at the problem of low efficiency of industrial chain data characteristics, the algorithm proposed a construction method of logistics indicators to describe the whole process of insider trading behavior according to the three dimensions of own trading mode, commodity trading mode and dealer mode;
2) Secondly, aiming at the problem of long time span of transaction, a method of using dynamic sliding window is proposed to extract the logistics characteristics within the time interval, and then judge whether the time interval is abnormal;
3) Finally, the isolation forest algorithm is improved to identify the abnormal data in the abnormal window. The experimental results show that compared to using the isolation forest methods, the F1 value of the algorithm in this paper is increased by **20.68%** in identifying the insider trading behavior of traders in the industrial chain trading market.

2 Related Work

In terms of methods, insider trading detection in the industrial chain market can be divided into model-based and data-driven insider trading detection methods [4].

For the model-based insider trading detection methods, Fama et al. proposed the event study method to measure the normal returns and abnormal returns before and after a certain event [5], which can be used to judge insider trading. Easley et al. mathematically described the trading process of informed and uninformed people, and proposed a model to estimate the probability of informed trading, which is used to estimate the probability of whether a transaction is an informed trader [6]. Mienna proposed a statistical model based on long time series data sets [7] in view of the complex trading strategies of insider traders and the difficulty of calculating additional returns in traditional econometrics models. Cline et al. considered the partial observability of insider trading and proposed a bivariate probability model to detect the behavior of illegal insider trading [8].

For data-driven insider trading detection methods, Deng et al. proposed a Gradient boosted decision tree (GBDT) based approaches for insider trading detection with differential evolution (DE) for parameter initialization [9, 10]. Esen et al. proposed a clustering-based insider trading detection method, which takes the outlier value of trading behavior as the suspicion degree of insider trading behavior through K-means and hierarchical clustering method and verifies it through the event study method [11]. Islam proposed the method of using Long Short-Term Memory network (LSTM) in deep learning to learn the structured and unstructured features of illegal insider trading events [12]. Seth et al. proposed a multi-stage insider trading detection method including deep neural network, consensus model and statistical methods to identify illegal insider trading behaviors through event analysis and detection of unstructured and structured data [13]. Lauar et al. proposed to build a training data set based on news events before insider trading events and proposed an augmentation method to expand the size of the data set, and used XGBoost to predict insider trading events [14].

However, the above methods do not consider the problems of inefficiency of industrial chain data characteristics and long transaction time span, so that the previous index modeling cannot be applied, resulting in poor algorithm effect. Therefore, this paper designs an insider trading behavior detection algorithm of industrial chain based on the characteristics of logistics time interval to mine the hidden insider trading behavior.

3 Problem Formulation and Analysis

In this section, the problem of insider trading detection in the industrial chain trading market is formally defined. The trading of industrial chains is different from other electronic transactions, which involves the transportation of goods logistics and has a large time span, so it is necessary to combine the characteristics of logistics to detect insider trading behavior. Next, the insider trading detection problem based on the characteristics of logistics time interval in the industrial chain is defined in detail.

Suppose there are a number of dealers that have traded during the time interval $[t_a, t_b)$, and we use sets $A = \{a_1, \ldots, a_i, \ldots, a_N\}$ to represent these trading accounts. A trader $a_i \in A$, trading in M types of commodities, we define it as $C^i = \left\{c_1^i, \ldots, c_j^i, \ldots, c_M^i\right\}$. Among them, where K transactions on the j -th commodity are represented as $R_j^i = \left\{r_{j,1}^i, \ldots, r_{j,k}^i, \ldots, r_{j,K}^i\right\}$, then the k -th transaction can be represented as $r_{j,k}^i = \left\langle s_{j,k}^i, e_{j,k}^i, p_{j,k}^i, v_{j,k}^i, f_{j,k}^i \right\rangle$, where $f_{j,k}^i = 1$ represents the selling transaction behavior and $f_{j,k}^i = -1$ represents the buying transaction behavior.

The problem of insider trading behavior detection [15] is defined as: judging whether the trading behavior $r^i_{j,k}$ in the trading behavior data R^i_j of dealers is insider trading behavior. The variables involved are listed in Table 1.

Table 1. Variable description

variables	Description
a_i	Trading account a_i
$r^i_{j,k}$	The the k -th transaction made by trading account a_i on the trading behavior of j -th commodity categories
$s^i_{j,k}$	The start time of the k-th transaction of the transaction account a_i on the j -th commodity
$p^i_{j,k}$	Commodity price of the k -th transaction of the transaction account a_i on the j -th commodity
$v^i_{j,k}$	The number of items of the k -th transaction of the transaction account a_i on the j -th commodity
$e^i_{j,k}$	The completion time of the k -th transaction of the transaction account a_i on the j -th commodity
L	Sliding window time length
H	Detecting sequence length

4 Algorithm Design

4.1 Interval Logistics Index Construction Algorithm

The main function of interval logistics characteristic index construction is to use indicators [16] under different dimensions to describe insider trading behavior and normal trading behavior in the industrial chain trading market. The specific steps of interval logistics characteristic index construction algorithm are as follows:

Using Sliding Window to Divide the Logistics Behaviour. The logistics behavior of the trading account $\boldsymbol{a_i}$ that detects the time interval of $[\boldsymbol{t_a}, \boldsymbol{t_b})$ on the $\boldsymbol{j}$-th commodity is divided into N segments using the unit time interval length T, where T is expressed as the Eq. 1:

$$T = \frac{t_b - t_a}{N} \tag{1}$$

Therefore, the time interval sequence obtained by partitioning is expressed in Eq. 2:

$$S = \{\ [t_a, t_a + T), [t_a + T, t_a + 2T), \ldots, [t_a + (N-1)T, t_b)\} \tag{2}$$

According to the characteristics that the logistics interval will span multiple units of time T, a sliding window of length L is used to extract the logistics characteristics of this logistics behavior, and the time interval of the sliding window is expressed as Eq. 3:

$$SS = \{\ [t_a, t_a + L), [t_a + T, t_a + T + L), \ldots, [t_a + NT - L, t_b)\} \tag{3}$$

According to different types of logistics behaviors of traders, logistics behaviors related to logistics characteristics in the sliding window within $[t_x, t_{x+L})$ can be divided into buying logistics behaviors and selling logistics behaviors, in which logistics behaviors entering the warehouse are represented by B_{t_x} in formula 4, and logistics behaviors exiting the warehouse are represented by S_{t_x} in Eqs. 4 and 5.

$$B^i_{j,t_x} = \{\ (t^i_{j,k}, e^i_{j,k}, p^i_{j,k}, v^i_{j,k}, f^i_{j,k}) | t_x \le e^i_{j,k} < t_x + L, f^i_{j,k} = 1\} \tag{4}$$

$$S^i_{j,t_x} = \{\ (t^i_{j,k}, e^i_{j,k}, p^i_{j,k}, v^i_{j,k}, f^i_{j,k}) | t_x \le s^i_{j,k} < t_x + L, f^i_{j,k} = -1\} \tag{5}$$

Calculating Logistics Indicators of Incoming Warehouse and Outgoing Warehouse. The total value of goods corresponding to logistics behaviors of incoming warehouse and outgoing warehouse are expressed as Eqs. 6 and 7:

$$U^i_{j,t_x} = \sum_{R^i_{j,k} \in B_{t_x}} p^i_{j,k} * v^i_{j,k} \tag{5}$$

$$V^i_{j,t_x} = \sum_{R^i_{j,k} \in S_{t_x}} p^i_{j,k} * v^i_{j,k} \tag{7}$$

For different types of logistics behavior, the contributions to the logistics index are different. Therefore, the commodity value of logistics behaviors of incoming warehouse and outgoing warehouse accounting for their total logistics behaviors are defined as Eqs. 8 and 9, respectively.

$$\alpha^i_{j,t_x} = \frac{U^i_{j,t_x}}{U^i_{j,t_x} + V^i_{j,t_x}} \tag{8}$$

$$\beta^i_{j,t_x} = \frac{V^i_{j,t_x}}{U^i_{j,t_x} + V^i_{j,t_x}} \tag{9}$$

Calculating the Characteristic Indicators of Three Dimension. Next, in order to better describe the indicators proposed, we have made the following definitions:

Definition 1. Total value of goods in interval: it represents the total value of goods in and out of warehouse on the logistics characteristics within the sliding window, and the outlier of the timing relationship of the trader's logistics behavior on this kind of goods. The logistics behavior in this time zone, the total value of the goods that are shipped out and shipped into the warehouse are expressed as Eq. 10:

$$O^i_{j,t_x} = U^i_{j,t_x} + V^i_{j,t_x} \tag{10}$$

In the formula, the first item represents the total value of all goods entering the warehouse in the time range of $[t_x, t_{x+L})$, which is expressed as the commodity price $p^i_{j,k}$ multiplied by the quantity $v^i_{j,k}$ of all purchasing logistics behaviors, and the second item represents the total value of all goods exiting the warehouse.

Definition 2. Ratio of interval commodity trading to all traded commodities: it represents the ratio of the value of commodities in the sliding window to the sum of the value of commodities traded by all users in the sliding window and is expressed as Eq. 11:

$$P^i_{j,t_x} = \alpha^i_{j,t_x} \frac{U^i_{j,t_x}}{\sum_j U^i_{j,t_x}} + \beta^i_{j,t_x} \frac{V^i_{j,t_x}}{\sum_j V^i_{j,t_x}} \tag{11}$$

In the formula, α^i_{j,t_x} and β^i_{j,t_x} represent the proportion of logistics activities of incoming warehouse and outgoing warehouse respectively; In the first term, the numerator represents the total value of the goods j of the incoming warehouse in the time range $[t_x, t_{x+L})$, and the denominator represents the sum of the total value of all the goods of the incoming warehouse in the time range. In the second term, the numerator represents the total value of the goods j of the outgoing warehouse in the time range $[t_x, t_{x+L})$, and the denominator represents the sum of the total value of all the goods of the outgoing warehouse in the time range.

Definition 3. The ratio of interval commodity trading to all dealers: it represents the ratio of the sum of commodity values of the logistics behavior of the dealer and all dealers on this commodity, which is expressed as Eq. 12.

$$Q^i_{j,t_x} = \alpha^i_{j,t_x} \frac{U^i_{j,t_x}}{\sum_i U^i_{j,t_x}} + \beta^i_{j,t_x} \frac{V^i_{j,t_x}}{\sum_i V^i_{j,t_x}} \tag{12}$$

In the formula, the numerator represents the total value of goods stored in the warehouse within the time range. The denominator represents the sum of the total value of the inventory of all the traders who traded in the commodity during the time period.

4.2 Interval Anomaly Detection Algorithm for Logistics Characteristics

The main function of abnormal interval detection based on interval logistics characteristic indexes is to detect abnormal interval according to the logistics characteristic indexes and the surrounding normal indexes. The specific steps of the algorithm are given in Algorithm 1, and we describe its detailed process as follows:

1) Construct index sequence composed of sliding window and surrounding time interval. For the r-th index F^i_{j,r,t_x} of the j-th commodity logistics of the i-th trader in the time range $[t_x, t_{x+L}]$, the logistics index within the detection time interval H is expressed as Eq. 13:

$$FS^i_{j,r,t_x} = \left\{ F^i_{j,r,t_x - \frac{H}{2}L}, F^i_{j,r,t_x - \frac{H}{2}(L-1)}, \ldots, F^i_{j,r,t_x}, \ldots, F^i_{j,r,t_x + \frac{H}{2}(L-1)}, F^i_{j,r,t_x + \frac{H}{2}} \right\} \tag{13}$$

When detecting whether each sliding window is abnormal with the surrounding sliding window, the length of L time interval is used to select in turn, so as to ensure that the sliding window to be detected does not overlap. During sequence exploration, it is necessary to ensure that it does not exceed the boundaries of the original indicator sequence [17].

2) Calculate the anomaly index corresponding to the sliding window. For the r-th index outlier of the j-th commodity logistics of the i-th trader in the time range [t_x, t_{x+L}] is expressed as Eq. 14:

$$S^i_{j,r,t_x} = \frac{F^i_{j,r,t_x} - median\left(FS^i_{j,r,t_x}\right)}{MAD_{FS^i_{j,r,t_x}}} \tag{14}$$

The function of *median* represents the Median of samples in the time series, and represents the Median Absolute Deviation (MAD) of samples [18].

3) Determine whether the sliding window is abnormal. For each feature, its outliers are calculated, and its total outliers are expressed as Eq. 15. ρ_r represents the parameters of this indicator:

$$S^i_{j,t_x} = \sum_r \rho_r S^i_{j,r,t_x} \tag{15}$$

Whether the interval is abnormal is judged according to the relationship with threshold δ_s. The robustness of the anomaly detection algorithm is enhanced by weighted fusion of abnormal degrees of multiple dimensional indicators [19].

4.3 Abnormal Logistics Feature Detection Based on Isolation Forest

In this paper, the method based on isolation forest [20] is used to perform anomaly detection on the data in the detected anomaly interval, which can improve the effectiveness of the whole anomaly detection algorithm and enhance the interpretability of the algorithm [21]. The specific steps of the abnormal logistics feature detection algorithm based on isolation forest are given in Algorithm 2, and we present its specific description as follows:

1) The newly constructed feature vector for each transaction is expressed as $\left\langle p^i_{j,k}, v^i_{j,k}, f^i_{j,k}, An^i_{j,k}\right\rangle$;
2) Mark the transaction $R^i_{j,k} = \left\langle s^i_{j,k}, e^i_{j,k}, p^i_{j,k}, v^i_{j,k}, f^i_{j,k}\right\rangle$ to determine whether it is within the m-th anomaly interval [t^m_s, t^m_e], and if so, it is marked as:

$$An^i_{j,k} = \begin{cases} 1, t_s \leq s^i_{j,k} < t_e f^i_{j,k} = 1 or t_s \leq e^i_{j,k} < t_e f^i_{j,k} = -1 \\ 0, otherwise \end{cases} \tag{16}$$

The feature vector is input into the isolation forest algorithm to determine whether each logistic behavior of the user is abnormal.

5 Experiments

5.1 Experimental Environment

The experiments were run on a PC with windows10 operating system, the Processor was AMD Ryzen 7 3800X 8-Core Processor 3.89 GHz. The experimental code was implemented using Python 3.8.13.

The experimental data comes from the real industry chain data. Due to the risk of leakage of logistics data for traders' privacy, therefore, all the data provided were desensitized according to the industry chain electronic transaction sensitive information desensitization and encryption regulations. All traded commodity names and dealer names in this paper are represented using a post-desensitized string.

5.2 Performance Indices and Benchmark Algorithms

Experimental Indicators. For classification problems, the following indicators are usually considered [21]:

1) Accuracy rate: all predicted dealer associations are consistent with the actual user associations.
2) Precision rate: the proportion of actual transactions that are abnormal among all transactions that are predicted to be abnormal.
3) Recall rate: the proportion of actual transactions predicted to be abnormal among all transactions that are abnormal.
4) F1 value: Considering the above two indicators.
5) Running time.

Comparison Algorithm. We have chosen the following three comparative algorithms:

1) Using the Isolation Forest algorithm (IForest) for detection [20], randomly selecting features from the original features to construct an isolated tree, and calculating the outliers of the sample through the path length of the sample. The shorter the path length, the greater the outlier value;
2) The Histogram Based Outlier Score (HBOS) method [22] is used to divide each dimension of the data into intervals, and the outlier value of the sample is calculated through the density of the interval where the sample is located. The lower the density, the greater the outlier value;
3) The anomaly detection method of K Nearest Neighbor (KNN) [23] is used to calculate the outliers by calculating the distance between the sample and the surrounding points. The larger the distance, the larger the outlier.

Parameter Settings. The effectiveness of the algorithm was verified by randomly sampling the original data set from 50% to 100% to obtain different sizes of data sets. 50 abnormal transactions are generated per day, where the number of items in each abnormal transaction is set to 100. Sequence detection length H = 6.

Table 2. Experimental index results under different transaction sample numbers

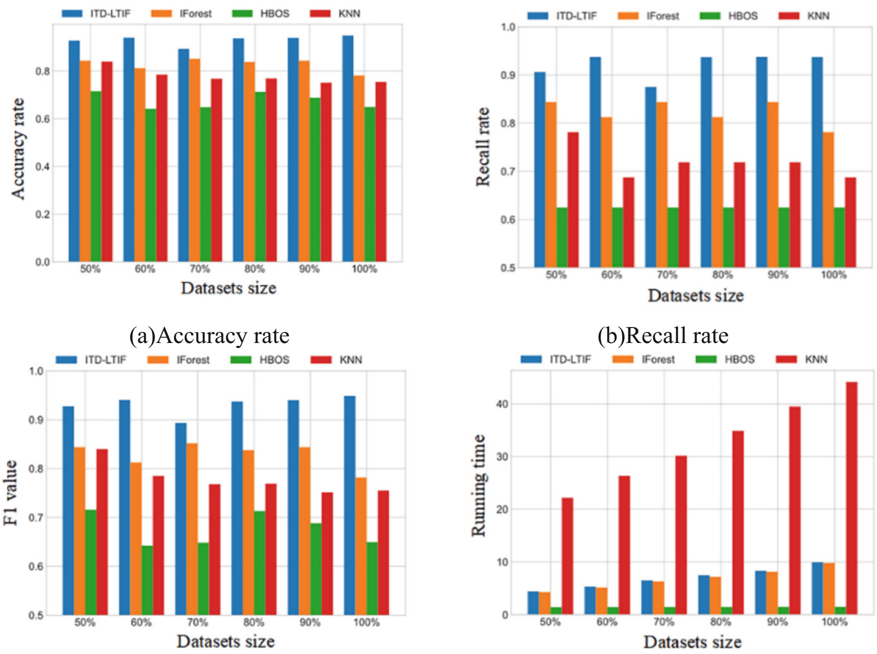
Datasets size %	Algorithm name	Accuracy rate	Recall rate	F1 value	Running time/second
50%	ITD-LTIF	**0.9277**	**0.9062**	**0.9168**	4.419
	IForest	0.8438	0.8438	0.8438	4.291
	HBOS	0.7158	0.6250	0.6673	**1.454**
	KNN	0.8401	0.7812	0.8096	22.185
70%	ITD-LTIF	**0.8934**	**0.8750**	**0.8841**	6.484
	IForest	0.8517	0.8438	0.8477	6.311
	HBOS	0.6485	0.6250	0.6365	**1.473**
	KNN	0.7682	0.7188	0.7427	30.172
100%	ITD-LTIF	**0.9488**	**0.9373**	**0.9430**	9.956
	IForest	0.7817	0.7812	0.7814	9.817
	HBOS	0.6498	0.6250	0.6372	**1.503**
	KNN	0.7550	0.6875	0.7197	44.129

5.3 Experimental Result

As shown in Table 2 and Fig. 1(a)(b)(c), the insider trading detection method based on time interval characteristics proposed in this paper is superior to other comparison algorithms in terms of precision, recall rate and F1 value on transaction data sets of different sizes. Compared with the isolation forest algorithm, the proposed algorithm improves the accuracy index by ***9.94%***, ***15.74%***, ***4.89%***, ***11.84%***, ***11.37%***, ***21.37%*** respectively. In terms of the recall rate, the proposed algorithm improves by ***7.39%***, ***15.38%***, ***3.69%***, ***15.34%***, ***11.12%***, ***19.98%*** respectively; In terms of F1 value, the algorithm proposed in this paper improves by ***8.65%***, ***15.55%***, ***4.29%***, ***13.59%***, ***11.24%***, ***20.68%*** respectively.

As shown in Table2 and Fig. 1(d), the proposed insider trading detection algorithm based on logistics time interval characteristics is not significantly different from the isolation forest method in terms of time, and the algorithm execution time shows an increasing trend with the increase of transaction datasets size. The histogram based anomaly detection algorithm takes the least time, and with the increase of the size of the transaction data set, the execution time of the algorithm has little effect. The anomaly detection method based on k-nearest neighbor method takes the longest time and the execution time of the algorithm increases significantly with the increase of the size of the transaction data set.

Results Analysis. In terms of precision, recall rate and F1 index, the insider trading detection algorithm based on logistics time interval characteristics proposed in this paper is superior to other comparison algorithms, but there is no obvious change in the trend, which is due to the randomness of sample sampling, and different sizes of transaction data sets are different in the distribution of normal trading behavior. As for the running time of the algorithm, the insider trading detection algorithm based on the characteristics



(a)Accuracy rate

(b)Recall rate

(c)F1value

(d)Running time

Fig. 1. Variation of experimental indicators with the proportion of datasets size

of logistics time interval proposed in this paper has little difference. This is because although the algorithm proposed in this paper needs more time to detect the additional time interval, the isolation forest method has a higher time dimension and requires more time. Therefore, the running time of the two algorithms is not greatly different. Histogram-based anomaly detection takes the least time due to its simple detection method and fast computation, so the increase of its dataset size is not enough to have a noticeable effect on it. The k-nearest neighbors algorithm takes the longest time, because it takes a lot of time to calculate the distance between its K neighboring points. With the increase of the proportion of the datasets size, the algorithm needs to spend more running time on transaction detection.

6 Conclusions

This paper studies the detection of insider trading behavior based on logistics characteristics in the industrial chain trading market, and proposes an insider trading detection algorithm based on logistics time interval characteristics. Firstly, according to the characteristics of long logistics time span, the dynamic sliding window is used to extract the logistics transaction behavior in the time interval. Secondly, three characteristics under the dimension of own trading mode, commodity trading mode and dealer mode are used to describe. Then, after the abnormal logistics time intervals are identified, the overlapping abnormal time intervals are optimized into non-overlapping abnormal intervals by using the method of reducing the sliding window to improve the accuracy of the algorithm. Finally, the method based on isolation forest was used to judge the abnormal transaction behavior of the extracted effective logistics features, so as to improve the effect of the algorithm. Through the experiments on the logistics data sets of real trading behavior of dealers, the results show that the proposed algorithm improves the F1 value by 20.68% compared with the direct using of isolation forest.

References

1. Angelopoulos, J., Sahoo, S., Visvikis, I.D.: Commodity and transportation economic market interactions revisited: new evidence from a dynamic factor model. Transp. Res. Part E: Logist. Transp. Rev. **133**, 101836 (2020)
2. Baklarz, A., Bogusz, J., Martysz, C.: Models of Propagation of Inside Information. Acta Physica Polonica A **138**(1) (2020)
3. Adams, B.J., Perry, T., Mahoney, C.: The challenges of detection and enforcement of insider trading. J. Bus. Ethics **153**(2), 375–388 (2018)
4. Hilal, W., Gadsden, S.A., Yawney, J.: Financial fraud: a review of anomaly detection techniques and recent advances. Expert Syst. Appl. **193**, 116429 (2022)
5. Fama, E.F., Fisher, L., Jensen, M.C., et al.: The adjustment of stock prices to new information. Int. Econ. Rev. **10**(1), 1–21 (1969)
6. Easley, D., Kiefer, N.M., O'hara, M., et al.: Liquidity, information, and infrequently traded stocks. J. Finance **51**(4), 1405–1436 (1996)
7. Minenna, M.: Insider trading, abnormal return and preferential information: supervising through a probabilistic model. J. Bank. Finance **27**(1), 59–86 (2003)
8. Cline, B.N., Posylnaya, V.V.: Illegal insider trading: commission and sec detection. J. Corp. Finan. **58**, 247–269 (2019)
9. Deng, S., Wang, C., Wang, M., et al.: A gradient boosting decision tree approach for insider trading identification: an empirical model evaluation of China stock market. Appl. Soft Comput. **83**, 105652 (2019)
10. Deng, S., Wang, C., Fu, Z., et al.: An intelligent system for insider trading identification in Chinese security market. Comput. Econ. **57**(2), 593–616 (2021)
11. Esen, M.F., Bilgic, E., Basdas, U.: How to detect illegal corporate insider trading? A data mining approach for detecting suspicious insider transactions. Intell. Syst. Account. Finan. Manage. **26**(2), 60–70 (2019)
12. Islam, S.R., Khaled Ghafoor, S., Eberle, W.: Mining illegal insider trading of stocks: a proactive approach. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 1397–1406. IEEE, Seattle (2018)
13. Seth, T., Chaudhary, V.: A predictive analytics framework for insider trading events. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 218–225. IEEE, Atlanta (2020)
14. Lauar, F., Arbex Valle, C.: Detecting and predicting evidences of insider trading in the Brazilian market. In: Dong, Y., Ifrim, G., Mladenić, D., Saunders, C., Van Hoecke, S. (eds.) Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track. ECML PKDD 2020. LNCS, vol. 12461, pp. 241–256. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67670-4_15
15. Donoho, S.: Early detection of insider trading in option markets. In: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 420–429. Association for Computing Machinery, New York (2004)
16. Tangwongsan, K., Hirzel, M., Schneider, S., et al.: General incremental sliding-window aggregation. Proc. VLDB Endowment **8**(7), 702–713 (2015)

17. Blázquez-García, A., Conde, A., Mori, U., et al.: A review on outlier/Anomaly detection in time series data. ACM Comput. Surv. **54**(3), 56:1–56:33 (2021)
18. Howell, D.C.: Median Absolute Deviation. Encyclopedia of Statistics in Behavioral Science. Wiley, New York (2005)
19. Sun, H., He, Q., Liao, K., et al.: Fast anomaly detection in multiple multi-dimensional data streams. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 1218–1223 (2019)
20. Ounacer, S., Bour, H.A.E., Oubrahim, Y., et al.: Using isolation forest in anomaly detection: the case of credit card transactions. Periodicals Eng. Nat. Sci. **6**(2), 394–400 (2018)
21. Han, S., Hu, X., Huang, H., et al.: ADBench: anomaly detection benchmark. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
22. Kalaycı, İ., Ercan, T.: Anomaly detection in wireless sensor networks data by using histogram based outlier score method. In: 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 1–6 (2018)
23. Ying, S., Wang, B., Wang, L., et al.: An improved KNN-based efficient log anomaly detection method with automatically labeled samples. ACM Trans. Knowl. Disc. Data **15**(3), 34:1–34:22 (2021)

Link Attributes Based Multi-service Routing for Software-Defined Satellite Networks

Xueyu Lu, Wenting Wei(✉), Liying Fu, and Dong Zhang

State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China
luxueyu@stu.xidian.edu.cn, wtwei@xidian.edu.cn

Abstract. Satellite networks are the potential complementary of terrestrial networks, which are expected to provide full-coverage and broadband access anywhere, anytime. As satellite networks scale up, Software-Defined Satellite Network (SDSN) is a promising paradigm due to its higher flexibility in network management. However, in the SDSN with highly time-varying characteristics, the traditional terrestrial routing strategy can hardly meet the QoS requirements for diverse services. In this paper, we propose a Link-Attributes-based multi-service On-Demand Routing (LAODR) algorithm under SDSN architecture. It quantifies the reliability of the Inter-Satellite Links (ISL) and provides a fine-grained state description of the dynamic topology. Furthermore, we select the K-shortest path as the solution space and reasonably allocate link resources based on LAODR to meet the diverse service demands of users. We implement LAODR and conduct experiments by using real network topologies. The results validate that LAODR not only satisfies the QoS requirements of different types of services but also outperforms other routing algorithms in terms of mean end-to-end latency, packet loss ratio, throughput and node congestion degree.

Keywords: Software-Defined Satellite Network · Link attributes · Multi-service routing · Reliable routing

1 Introduction

As the world welcomes its 8 billion inhabitants, the Internet is penetrating people's daily lives [1]. Despite the convenience the Internet offers, 34% of the global population still does not have access to it, particularly those in remote or disadvantaged areas [2]. Clearly, global coverage cannot be solved by terrestrial networks alone. Fortunately, satellite communication is an ideal long-distance communication technology with wide coverage, low affect by terrain, landscape, and natural disasters. Satellite networks, which are the convergence of satellite communication and Internet technology, are expected to be a high-capacity transmission solution providing seamless global coverage. They can not only improve ubiquitous access to global networks, but also respond quickly to emergency communication needs. How to efficiently exploit their potential for applications becomes an important issue.

To ensure reliable communication, satellite network routing design is a fundamental technology. The traditional offline routing algorithms lack the dynamic self-adaptive

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 130–142, 2024.
https://doi.org/10.1007/978-981-99-8211-0_13

capability for satellite networks. As the satellite network expands and the ISLs become increasingly intricate, the restrictions of these algorithms become increasingly apparent. By obtaining the state information of satellite networks, it is possible to design dynamic routing strategies that are suitable for network topology changes. But the frequent signaling exchanges between satellites are likely to cause additional network burdens. Furthermore, the range of service types in satellite networks creates different Quality of Service (QoS) requirements. To efficiently utilize the limited resources on board, it also poses challenges for multi-QoS routing design [3, 4] in satellite networks.

In traditional satellite distributed network architectures, the control and data planes are unified. Satellite nodes must not only forward data packets, but also implement network control functions such as traffic state monitoring, link state maintenance, and route calculation, thus consuming valuable on-board payload and inter-satellite link resources. To meet rising traffic demands and network heterogeneity, Software Defined Network (SDN) can be used to simplify the management of communication networks for future satellite Internet architectures. The Software Defined Satellite Network (SDSN) architecture is a promising solution for monitoring and managing the network more flexibly and facilitating network expansion [5–12].

Currently, researches on SDN-based satellite network routing are focused on the network architecture. However, hierarchical-based SDNs need to consider the reliability of routing policies. On the one hand, the timeliness of routing tables, where the higher-level satellites need to accurately capture the network topology of the lower-level satellites promptly. On the other hand, the robustness of routing policies, where the inter-layer links need to be stable to ensure the effective update of the routing policies of the higher-level satellites. In addition, the SDSN routing algorithms proposed by researchers mainly focus on the guarantee of different QoS. For example, the Software-Defined Routing Algorithm (SDRA) obtains the optimal routing path through a centralized routing policy with only a single QoS goal as the optimization point, while most of the literature does not study the differentiated services for different service requirements deeply enough.

In this paper, we propose a Link-Attributes-based On-Demand Routing (LAODR) scheme in SDSN to enhance the adaptiveness and reliability during data transmission. Specifically, we refer to the typical two-layer architecture in the design of the SDSN framework, consisting of a GEO satellite and a ground computing center acting as the controller. In the LAODR, we take service adaption as the main goal to achieve on-demand routing. Meanwhile, by quantifying the dynamic attributes of links, the control plane can sense the dynamic changes of the network topology in time to ensure the reliability of the routing strategy and achieve dynamic topology adaption. To evaluate the performance of the LAODR algorithm, we developed a satellite network simulation platform based on STK and OMNeT++. Simulation results demonstrate that the proposed algorithm in this paper outperforms the basic algorithms in terms of latency, packet loss ratio, and throughput.

The contribution of this paper is the proposal of the LAODR algorithm under the SDSN architecture, which quantifies the reliability of Inter-Satellite Links, provides a fine-grained state description of the dynamic topology, and efficiently meets diverse service demands while outperforming other routing algorithms in terms of QoS metrics.

The rest of this paper is organized as follows. The framework of the Software-Defined Satellite Network is constructed in Section 2. In Section 3, a link-attributes-based multi-service on-demand routing algorithm is proposed. In Section 4, we give the simulation results and performance analysis. In Section 5, we conclude this paper.

2 SDN-Enabled LEO/GEO Satellite Network Model

Software Defined Network (SDN) will play an important role in the future development of satellite Internet by decoupling the control plane and data plane and simplifying the management of the network. We use a typical multilayer SDSN centralized control framework in this paper, which contains GEO control plane, ground control plane and LEO data plane [13]. The control plane consists of GEO satellites and Ground Computing Center (GCC), where GEO satellites can take advantage of their natural coverage characteristics to collect global traffic information and formulate routing policies, and GCC can take advantage of computing power resources to process the acquired information and mine the routing laws. The data plane is composed of LEO satellites, which only need to provide data transmission services based on the routing table issued by GEO satellites. The architecture is shown in Fig. 1.

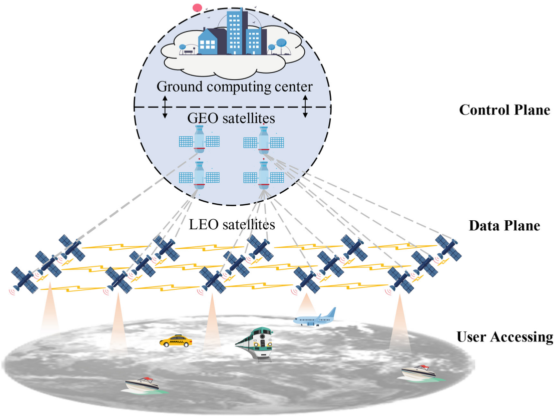


Fig. 1. Software-Defined Satellite Network Architecture.

Control Plane: GEO satellites and GCC. Their main functions are traffic scheduling and access user's path assignment. GEO satellites are responsible for collecting the link traffic state of LEO satellites, such as time slot, satellite location, link load, remaining capacity, etc., to make multi-service routing decisions based on link attributes, and further send network topology information and routing decision results to GCC continuously. The GCC trains the routing model based on the data sent by GEO, predicts the future routing paths from the past traffic and routing laws, and uploads the routing results to GEO satellites with a certain frequency. Then, the GEO satellites integrate its own and the received GCC routing scheme to get a unique routing result that adapts to the state of the satellite network, and sends it forward to the data plane. Based on this architecture, the GEO satellites are used as the primary controller and the GCC as the secondary controller to ensure the timeliness and accuracy of routing decisions, and alleviate the limitation of on-board computing resources.

Data Plane: LEO satellites. Their main functions are request upload, network traffic status upload and data transmission. The LEO satellites periodically upload the network link status information and transmission request to the GEO satellites, and transmits the packets according to the routing table returned from the GEO satellites. Since the LEO satellites transfer the decision of routing to the GEO satellites, it greatly reduces the demand for onboard computing resources.

On the one hand, the GCC utilizes the periodic predictability of satellite topology change and collects the routing decision results of GEO satellites in the early stage for regular analysis; in the later stage, the regular routing forwarding strategy can be uploaded to GEO satellites to assist GEO satellites' routing decision. On the other hand, the data forwarding of LEO is still dominated by GEO's decision, which ensures the timeliness of the routing strategy and reduces the impact of long-distance ISL on routing reliability. In addition, since the routing decisions are made at the GEO satellites, the data forwarding of the LEO satellites, the training and updating of the GCC model, and the routing strategy formulation by the GEO satellites can occur in parallel, minimizing the impact of the model update on the routing performance.

3 Design of LAODR

In this section, a satellite network description is given to analyze the properties of ISN first. Then, link utilities are quantified to portray the reliability of links, which are used as a decision metric for target optimization in the routing model. Finally, a Link-Attributes-based multi-service On-Demand Routing (LAODR) algorithm is designed to achieve adaptive routing for dynamic topologies and multiple services.

3.1 Description of the LEO Satellite Network

Satellites often establish communication links with surroundings via microwave/laser Inter-Satellite Links (ISLs). Generally, each node is interconnected with four surrounding satellites to establish ISLs. Among them, the satellite establishes two ISLs in the same plane, called Intra-plane ISL, and the two other interplanetary links with satellites in different planes, called Inter-plane ISL. If a satellite enters the polar region, its Inter-plane ISL will be disconnected due to antenna tracking limitations, while the Intra-plane ISL mostly remains connected. The Inter-plane ISL is also temporarily broken when the angle of view or distance between two satellites changes too rapidly, which happens between two counter-rotating orbits when two planes are close or crossed.

We use Iridium constellation as the study object for LEO satellite routing, and a network topology schematic is established as shown in Fig. 2. The Iridium constellation consists of 6 orbits, each containing 11 LEO satellites. It should be noted that the polar region boundary is assumed to be 70° in this paper, and once the satellites enter the polar region, the Inter-plane ISLs are broken, while the Intra-plane ISLs continue to be maintained. Therefore, the Inter-plane ISLs in the red region do not exist. Also, the Inter-plane ISLs between Plane 1 and Plane 6 do not exist due to the reverse seam.

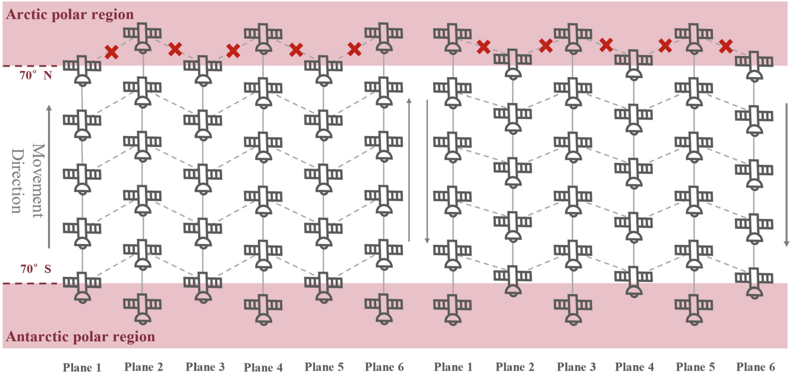


Fig. 2. Satellite Network Topology.

3.2 Quantification of Dynamic Link Attributes

Due to the dynamic nature of satellite networks, the ISL's state changes with the motion of satellites, and ISL's attributes such as Signal-to-Noise Ratio (SNR) [14], link duration [15] and buffer queue [16] affect the reliability of routing paths. Existing studies only describe the link states as simply on and off, which can easily cause unreliability of routing paths due to untimely and incomplete updating of link state information. These dynamic attributes can be quantified as the utility of ISLs to improve the adaptability of satellite routing to dynamic topologies [17]. We define these dynamic link attributes (SNR, link duration and buffer queue) as $\{U_S, U_L, U_B\}$, respectively.

First, to ensure the correct reception of data, the SNR of the receiving satellite should be greater than the reception threshold, as in (1). Second, to ensure the stability of transmission, ISLs with longer link duration should be selected as much as possible, as in (2). Third, satellites must have sufficient buffer queues to store and process packets, as in (3). Therefore, the dynamic characteristics of ISLs can be characterized to further quantify the impact of link attributes on communication quality.

$$U_S = P_r\left(SNR_{ij} > \gamma_0\right) = \int_{\gamma_0}^{\infty} SNR_{ij}dx = \int_{\gamma_0}^{\infty} \frac{\left|h_{ij}(t)\right|^2 L_{ij}^{-\gamma}(t)G}{N_0}dx \qquad (1)$$

where SNR_{ij} is the SNR of ISL between satellite i and j, the SNR_{ij} threshold is γ_0, h_{ij} is the channel characteristic, L_{ij} is the ISL's length, G represents the state of satellite, which is constant if it works normally, otherwise 0, and N_0 is the link noise power.

After a satellite enters the polar region, the Inter-plane ISL will be broken and only the Intra-plane ISL will continue to be maintained. Therefore, the duration of the link depends greatly on the latitude position of the satellite in the absence of sudden satellite failure. Let the starting moment of the link connection be t_{ij}^{on}, the disconnection moment be t_{ij}^{off}, and the current moment be t, with $t_{ij}^{on} \leqslant t \leqslant t_{ij}^{off}$. From the maximum link duration $l_{ij}^{\max} = t_{ij}^{off} - t_{ij}^{on}$ and the link duration $l_{ij}^{\Delta} = t_{ij}^{off} - t$, we can obtain:

$$U_L = \begin{cases} \frac{l_{ij}^{\Delta}}{l_{ij}^{\max}} & i, j \text{ in different orbits} \\ 1 & i, j \text{ in the same orbit} \end{cases} \tag{2}$$

Based on the queuing theory *M/M/1/N/*∞ model, where N is the capacity of the satellite buffer queue, assume that the arrival of packets obeys Poisson distribution, and set the packet flow rate λ, the satellite processing rate μ, the existing queue length n_{ed}, the current service packet size m, service capacity ρ. From the sojourn time of the service W_S, the minimum sojourn time of the service $W_{\min}$ and the packet loss ratio of the service P_B, we can calculate:

$$U_B = \frac{W_{\min}}{W_S}(1 - P_B) = \left(\frac{m}{\mu}\right) / \left(\frac{n_{\text{ed}} + m}{\mu}\right) \cdot \left(1 - \frac{(1-\rho)\rho^N}{1-\rho^{N+1}}\right) \tag{3}$$

The above three link dynamic attribute utilities are combined into a link utility U_{ij} to characterize the link reliability. The link utility U_{ij} can be expressed as follows:

$$U_{ij} = U_{S_{ij}}^{w_s^{ij}} \cdot U_{L_{ij}}^{w_l^{ij}} \cdot U_{B_{ij}}^{w_b^{ij}} \tag{4}$$

where w_s, w_l, w_b are the contribution weights of each attribute utility to the link utility calculated by the entropy value method, with $w_s^{ij} + w_l^{ij} + w_b^{ij} = 1$.

The link utility calculated in this part can well evaluate the dynamic properties of ISLs. It can predict the trend before link disconnection, reconstruction or node congestion occurs, which evaluates the reliability of the link to reduce the retransmission problem caused by packet loss and realize the self-adaptation to dynamic topology.

3.3 Link-Attributes-Based Multi-service On-Demand Routing

With the increasing number of satellites, the Satellite Internet will carry a richer range of services, which have different needs for Quality of Service (QoS). So how to design a differentiated routing scheme for services has become a key issue.

Table 1. QoS requirements for different services.

Category	Bandwidth	Latency	Reliability	Applications
Voice Stream	Low	High	Medium	IP Phone
Video Stream	High	Low	Medium	Video on Demand
Data stream	Medium	Medium	High	FTP, File Transfer

The service or QoS classifications defined by various standardization organizations are not the same, and it is difficult to achieve interoperability of multiple QoS routes without uniform classifications. We mapped the typical classifications and completed a brief service classification based on QoS requirements, as shown in Table 1. For example, voice and calls belong to delay-sensitive services; video belongs to bandwidth-sensitive services; and file transfer belongs to reliability-sensitive services.

Three-dimensional vectors (B, D, R) are used to indicate the comprehensive sensitivity of each type of service, where they represent path bandwidth, delivery delay, and path reliability, respectively. If the available bandwidth of each link is denoted as B_{ij}, the bandwidth occupied by the task B cannot exceed the minimum value of B_{ij}, $B \leqslant \min\left(B_{ij}, B_{jk}, \cdots, B_{mn}\right)$. If the link delay between two adjacent satellites is d_{ij}, the path delivery delay $D = \sum d_{ij}$. The link utility U_{ij} is obtained from the previous section, and the reliability of the path $R = U_{ij} \times U_{jk} \times \cdots \times U_{mn}$.

```
Algorithm 1: LAODR
Input: satellite latitude and longitude, network information (traffic, available bandwidth,
queue length, packet loss rate, service requirements).
Output: next-hop nodes of different business types ($nxt\_A,\ nxt\_B,\ nxt\_C$ ).
Initialize satellite network environment and network load;
for $s_i \in S$ do
    for $s_j \in S$ do
        Calculate the link attributes $U_{S_{ij}}, U_{L_{ij}}, U_{B_{ij}}$ of the link $e_{ij}$;
        Get the link utility $U_{ij} = {U_{S_{ij}}}^{w_s} \cdot {U_{L_{ij}}}^{w_l} \cdot {U_{B_{ij}}}^{w_b}$;
        Quantifying link latency $d_{ij}$, link available bandwidth $B_{ij}$, and link reliability $U_{ij}$;
    end for
end for
for $s_i \in S$ do
    for $s_j \in S$ do
        Compute the optimal set of paths $Path\{s_i \rightarrow s_j\}$;
        for $p_k \in Path\{s_i \rightarrow s_j\}$ do
            Calculate the path delay $D$, path bandwidth $B$, and path reliability $R$;
            Define optimization goals $\{\min\ D,\ \ \min\ B_I - B,\ \ \min\ R_I - R\}$;
            Choose the $p_k$ that minimizes $Z_k = W_B \cdot (B_I - B) + W_D \cdot D + W_R \cdot (R_I - R)$;
        end for
        Get the next hop node $nxt = p_k[1]$ of $s_i \rightarrow s_j$;
    end for
end for
Store the $nxt\_A,\ nxt\_B,\ nxt\_C$ of all satellite node pairs;
```

To achieve on-demand routing of services and full utilization of resources, the Multi-Objective Planning (MOP) model can be established, where the bandwidth, delay and reliability requirements of a service are b_n, d_n, r_n, as well as the ideal bandwidth, delay and reliability of the path are B_I, D_I, R_I We use the eigenvector method to solve the MOP model by assigning different weights $\boldsymbol{w} = [W_B, W_D, W_R]$ to different types of service QoS metrics. And transforming the MOP problem into a single-objective planning problem:

$$\begin{aligned} &\min Z = W_B \cdot Z_B + W_D \cdot Z_D + W_R \cdot Z_R \\ &\text{s.t.} \begin{cases} Z_B = B_I - B, \quad Z_D = D, \quad Z_R = R_I - R \\ \quad\; B \geqslant b_n, \quad D \leqslant d_n, \quad R \geqslant r_n \end{cases} \end{aligned} \tag{5}$$

Finally, to achieve a trade-off between reducing the computational complexity and ensuring the adaptation to the satellite topology, the paths are selected optimally, i.e., K shortest paths. The Dijkstra algorithm is used to calculate the set of the first optional paths between the source and destination nodes. The optimal paths satisfying (5) are solved iteratively to obtain on-demand routing policies for different service types. The designed LAODR algorithm implements adaptive routing for dynamic topologies and multiple services, and the overall algorithm pseudo-code is as Algorithm 1.

4 Evaluation

4.1 Experimental Setup

We first use the Standard Object Database (SOD) in the STK11.2 simulator to construct a satellite topology that meets the practical application and obtain the real-time latitude and longitude data of each satellite. In the control plane, four GEO satellites are deployed at equal intervals for global control and one GCC is located at Beijing for routing algorithm training and updating. In the data plane, the Iridium system, which is widely used in simulations, is deployed. Then, based on OMNeT + +5.6.2, we establish the algorithm simulation and verification platform, import the scenario of STK simulation by Python, build the control node and forwarding node, and control the disconnection and reconstruction of the ISL. Each packet generation rate is set to obey uniform distribution from 200 Kbps to 2000 Kbps and different tasks are labeled with sensitivity labels, where the percentages of delay-sensitive, bandwidth-sensitive and reliability-sensitive services are 0.2: 0.3: 0.5, respectively. The weight matrix of different services $\boldsymbol{w} = [W_B, W_D, W_R]$ in (5) is calculated by the eigenvector method. The main simulation parameters in this paper are shown in Table 2.

The designed algorithm LAODR results are compared with existing algorithms (e.g., classical Dijkstra's algorithm, IADR algorithm considering only link utility [17]) to verify and analyze five performance metrics. To ensure the reliability of the results, the average value of five experiments is taken as the simulation result.

Dijkstra: packets are calculated based on dijkstra algorithm to get the shortest path between node pairs, which has the minimum number of hops, but the performance is significantly degraded due to congestion when the traffic load is high.

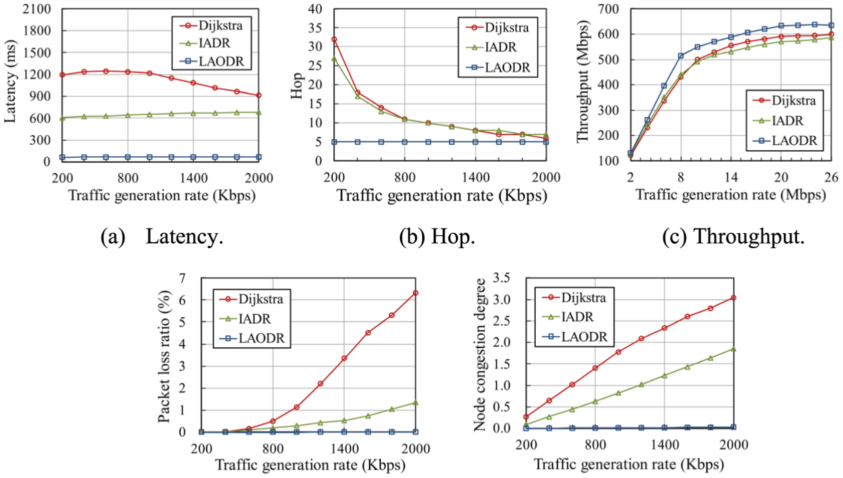
IADR (ISL Attributes-based Dynamic Routing): To improve the adaptability and reliability of LEO satellite network routing, IADR quantifies the link utility based on ISL attributes such as SNR, link duration and buffer queue. A routing path optimization model is constructed based on the multi-attribute decision scheme.

Table 2. Simulation parameters setting.

Parameter	Value	Parameter	Value
Polar region boundary	70°	Laser beam divergence half-angle	5e–3 rad
ISL Bandwidth	20 Mbps	Tracking error angle θ	1 mrad
ISL propagation delay	15 ms	SNR threshold γ_0	20 dB
Packet length	2 Kbytes	Transmitting power G	4 dBm
Buffer queue size	800 packets	Noise power N_0	1e–14 dBm
Switch processing latency	0.1 ms	Simulation time	100 s
Traffic generation rate	200–2000 Kbps	Routing calculation time step	1 s

4.2 Performance Evaluation Under Different Traffic Loads

Figure 3 (a) gives the comparison curves of the time delay for different traffic loads. Both IADR and LAODR algorithms increase slowly as the traffic increases, while the latency of Dijkstra's algorithm increases and then decreases. Dijkstra is prone to network congestion when the traffic is high, resulting in packet drops. The IADR and LAODR take the "buffer queue" attribute of the link into account, which can better balance the traffic across the network. LAODR algorithm has the lowest latency and compare to Dijkstra and IADR with 94.07% and 89.74% latency reduction. Figure 3 (b) shows the average hop count for different traffic loads. Dijkstra has a large instability on hop under different traffic sizes, and when the traffic volume increases, there is a sharp decay in the hop count, which is due to the large packet loss. The proposed LAODR algorithm has the most stable hop count with 59.02% and 57.27% reduction compared to Dijkstra and IADR. Figure 3 (c) compares the throughputs under different traffic loads. Since the three algorithms do not deliberately pursue network load balancing under low traffic, all have similar network throughput in the early stage. LAODR makes full use of LEO satellite resources to improve the data transmission efficiency of the satellite network, so the data transmission per second is improved by 8.25% and 10.13% compared with Dijkstra and IADR. Figure 3 (d) compares the packet loss performance. All three algorithms have a low packet loss ratio when the traffic is small. Since the Dijkstra scheme pursues the smallest transmission delay, all tasks are assigned to the shortest distribution path, which easily causes link overload and increases burst link congestion and causes packet loss. Thus, with the increase of traffic, the packet loss performance of Dijkstra decreases significantly. The IADR algorithm can select links with long link durations, and therefore, the packet loss ratio grows slowly and with smaller values. It is worth noting that the proposed LAODR algorithm not only considers the link stability, but also optimizes the traffic distribution of the network in multi-service on-demand routing. As a result, a low packet loss ratio can still be guaranteed when the traffic is high. The satellite congestion is evaluated in Fig. 3 (e). As the traffic increases, the node congestion degrees show an increasing trend. It is observed that Dijkstra has the largest congestion, IADR is the second and the proposed LAODR scheme has the smallest. LAODR takes each node load into account, thus alleviating the traffic imbalance problem. Compared with other algorithms, LAODR can utilize more nodes for pathfinding and thus has the best congestion performance. LAODR reduces node congestion by 99.26% and 98.59% compared to Dijkstra and IADR.



(a) Latency. (b) Hop. (c) Throughput.

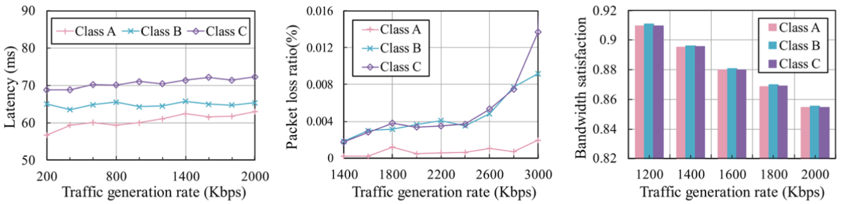
(d) Packet loss ratio. (e) Node congestion degree.

Fig. 3. Performance comparison of each algorithm under different traffic loads.

4.3 Performance Evaluation of Different Services

Meanwhile, the performance of the LAODR algorithm for different services is compared. The following Class A represents delay-sensitive services, Class B represents bandwidth-sensitive services, and Class C represents reliability-sensitive services.

Figure 4 (a) shows the latency of different services in LAODR. It can be found that Class A has higher delay requirements and therefore have lower routing delay compared to other types of services. Class C services have lower delay requirements and therefore have a higher delay, and they choose paths with larger hop counts to provide more choice for Class A. Figure 4 (b) shows the packet loss performance of different services. When the traffic is small, Class A has the smallest number of packets and higher priority making the packet loss performance excellent, while Class B does not require a high packet loss ratio, so the packet loss performance is poor. As the traffic load increases, the overall packet loss performance of Class C is gradually inferior to that of other services because the traffic volume of class C services is larger. Overall, the packet loss ratio of Class C is less than 0.015%, which has good routing reliability. Figure 4 (c) shows the bandwidth satisfaction for different services. This performance metric provides a good representation of the bandwidth enhancement space for different service routing paths. Class B is bandwidth-sensitive service, which requires more bandwidth and has better bandwidth satisfaction than other types of services.



(a) Latency. (b) Packet loss ratio. (c) Bandwidth satisfaction.

Fig. 4. Comparison of routing performance of different services of LAODR.

In summary, the paths assigned by the LAODR can better meet the QoS requirements of users and have good sensing capability for the link on/off and node failure. In contrast, the Dijkstra and IADR methods rarely consider user requirements and show poor identification ability for delay-sensitive and reliability-sensitive services.

5 Conclusion

This paper focuses on the problem of designing adaptive routing algorithms under the Software-Defined Satellite Networks (SDSN) architecture. Considering the dynamic characteristics of satellite network topology and the differentiated service demands of users, the LAODR scheme provides a fine-grained portrayal of link reliability attributes, based on which a multi-objective on-demand routing model is established to realize adaptive routing for dynamic topology and multiple services. The SDN framework is fused with the routing model to achieve efficient ISN traffic control and load balancing. Simulation results demonstrate that the proposed LAODR routing algorithm has superior traffic control performance and flexibility in routing, enabling efficient utilization of network resources to fulfill the varying service requirements of users. In the future, further investigation into the design of the routing prediction algorithm of the auxiliary controller GCC in the SDSN framework can be conducted to explore how to extract relevant regular big data and generate periodic routing policies.

Acknowledgments. This work was supported in part by the National Key R&D Program of China under Grant 2018YFE0202800, the National Natural Science Foundation of China under Grant 62102302, the Fundamental Research Funds for the Central Universities under Grant XJSJ23088 and the Youth Innovation Team of Shaanxi Universities.

References

1. Liu, Y., Fang, X., Ming, X., et al.: Decentralized beam pair selection in multi-beam millimeter-wave networks. IEEE Trans. Commun. **66**(6), 2722–2737 (2018)
2. International Telecommunication Union. Measuring Digital Development, Facts and Figures 2022. https://www.itu.int/itu-d/reports/statistics/2022/11/24/ff22-internet-use. Accessed 21 Jun 2023
3. Tomovic, S., Radusinovic, I., Prasad, N.: Performance comparison of QoS Routing algorithms applicable to large-scale SDN networks. In: IEEE Eurocon-international Conference on Computer as A Tool, pp. 1–6. IEEE (2015)
4. Roth, M., Brandt, H., Bischl, H.: Distributed SDN-based load-balanced routing for low earth orbit satellite constellation networks. In: 2022 11th Advanced Satellite Multimedia Systems Conference and the 17th Signal Processing for Space Communications Workshop (ASMS/SPSC), pp. 1–8. IEEE (2022)
5. Martinello, M., Ribeiro, M.R.N., De Oliveira, R.E.Z., et al.: Keyflow: a prototype for evolving SDN toward core network fabrics. IEEE Network **28**(2), 12–19 (2014)
6. Farris, I., Taleb, T., Khettab, Y., et al.: A survey on emerging SDN and NFV security mechanisms for IoT systems. IEEE Commun. Surv. Tutorials **21**(1), 812–837 (2019)
7. Papa, A., Cola, T.D., Vizarreta, P., et al.: Design and evaluation of reconfigurable SDN LEO constellations. IEEE Trans. Netw. Serv. Manage. **17**(3), 1432–1445 (2020)
8. Bertaux, L., Medjiah, S., Berthou, P.: Software defined networking and virtualization for broadband satellite networks. IEEE Commun. Mag. **53**(3), 54–60 (2015)
9. Kreutz, D., Ramos, F., Verissimo, P.E., et al.: Software-defined networking: a comprehensive survey. Proc. IEEE **103**(1), 14–76 (2015)
10. Nazari, S., Du, P., Gerla, M., et al: software defined naval network for satellite communications (SDN-SAT). In: 2016 IEEE Military Communications Conference, pp. 360–366. IEEE (2016)
11. Du, P., Nazari, S., Mena, J., et al: Multipath TCP in SDN-enabled LEO Satellite Networks. In: 2016 IEEE Military Communications Conference, pp. 354–359. IEEE (2016)
12. Tao, J., Liu, S., Liu, C.: A traffic scheduling scheme for load balancing in SDN-based space-air-ground integrated networks. In: 2022 IEEE 23rd International Conference on High Performance Switching and Routing (HPSR), pp. 95–100. IEEE (2022)
13. Wang, F., Jiang, D., Wang, Z., et al.: Fuzzy-CNN based multi-task routing for integrated satellite-terrestrial networks. IEEE Trans. Veh. Technol. **71**(2), 1913–1926 (2022)
14. Du, J., Jiang, C., Jian, W., et al.: Resource allocation in space multiaccess systems. IEEE Trans. Aerosp. Electron. Syst. **53**(2), 598–618 (2017)
15. Wang, J.F., Li, L., Zhou, M.T.: Topological dynamics characterization for LEO satellite networks. Comput. Netw. **51**(1), 43–53 (2007)

16. Tang, F., Zhang, H., Fu, L., et al.: Multipath cooperative routing with efficient acknowledgement for LEO satellite networks. IEEE Trans. Mob. Comput.Comput. **18**(1), 179–192 (2018)
17. Han, Z., Zhao, G., Xing, Y., et al: Dynamic routing for software-defined LEO satellite networks based on ISL attributes. In: 2021 IEEE Global Communications Conference (GLOBECOM), pp.1–6. IEEE (2021)

A Fuzzy Logical RAT Selection Scheme in SDN-Enabled 5G HetNets

Khitem Ben Ali[1,2](✉) and Faouzi Zarai[2]

[1] Lamsade Lab, University of Dauphine PSL, Paris, France
khitem.enis@gmail.com
[2] NTS'com Research Unit, University of Sfax, Sfax, Tunisia

Abstract. Mobile communication systems are witnessing an ongoing-increase in connected devices and new types of services. This considerable increase has led to an exponential augmentation in mobile data traffic volume. The dense deployment of small base stations and mobile nodes in traffic hotspots is considered one of the potential solutions aimed at satisfying the emerging requirements in 5G/Beyond 5G wireless networks. However, the ultra-densification poses challenges for the mobility management, including frequent, unnecessary and ping-pong handovers, with additional problems related to increased delay and total failure of the handover process. In this paper, we propose a new handover management approach using the Software Defined Networking (SDN) paradigm to overcome performance limitations linked to handover taking place at dense femtocell environments. With the exploitation of SDN, data plane and control plane are separated thus the HO decision can be made at the SDN controller. In addition, in order to reduce the complexity and delay of handover process, a Fuzzy logic system is used to decide whether a target candidate is suitable for handover. Simulation results validate the efficiency of our proposal.

Keywords: 5G · Macrocell · Femtocell · SDN · MIH · Handover · RAT Selection · Fuzzy logic

1 Introduction

Nowadays, there is an exponential deployment of radio networks characterized by an increasing number of users and panoply of services. Next Generation Wireless Networks (NGWNs) will combine existing and new technologies such as GPRS, UMTS, LTE, WiMAX and other backbone internet to provide high throughput anytime, anywhere for multimedia services. Mobile users will be able to communicate through different radio access technologies (RAT) and roam from one RAT to another by using multimode user equipement. This new mobile broadband technology is characterized by networks having different coverage ranges such as macro-cell, small-cell and atto-cell [1, 2], and diverse technologies interacting with varied types of entities. Although this new technology will bring several advantages in many areas, issues regarding mobility management are still a big challenge that needs to be solved in the future B5G/6G mobile networks. The outstanding issues include the challenges related to mobility routing, handover decisions,

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 143–153, 2024.
https://doi.org/10.1007/978-981-99-8211-0_14

handover authentication and control parameters settings, and more other mobility issues. So, to manage such a network and overcome these drawbacks, flexibility will be the key feature of this mobile generation. This architectural flexibility will be released by implementing the Software Defined Network (SDN) in the 5G mobile network [3]. SDN is based on the separation between the control plane, and the data plane allowing the handling of the traffic by means of software [4]. This separation helps improve scalability, flexibility, reliability, and simplification of network management [4, 5]. SDN architecture transforms network devices (e.g. switches) into dummy devices with no intelligence functions such as routing, major processing, and mobility management [6]. On the other hand, the IEEE 802.21 standard proposes a media independent handover (MIH) [7] specification for achieving seamless handover for mobile users in the same or in different networks. The main functionality offered by MIH is a seamless connection to different RATs. The control messages are relayed by the Function (MIHF) located in the protocol stack between the layer 2 wireless technologies and IP at layer 3. The MIH Information Service (MIIS) offers a variety of criteria and services that can be used to avoid network scanning. But sometimes scanning avoidance leads to inconsistent handover, that is, increased handover failure rates.

While there are many open challenges in 5G HetNets, our focus here is on identifying a solution to the problem of handover management. Understandably, handover procedures for existing networks are needed to support the macrocell/femtocell integrated network. Understandably, this situation may result in a large accumulation of unnecessary and frequent handovers, and also increase the risk of handover failure. In this paper, we propose a solution to optimize the handover in5G HetNets. In order to avoid unnecessary handoff and reduce the excessive interference, we present a handover strategy between Femtocell and Macrocell base on QoS level under SDN/MIH based 5G network. Our solution resides in creating a novel multi-criteria network selection mechanism. RSSI of Base Station, mobile user's movement direction, and base station available capacity are factors used in this work to improve handover decision while sustaining perceived network performance.

The main contributions of this work are as follows: i) Proposal of an interworking architecture that integrates MIH and SDN paradigms and enhances the handover procedure. The optimized architecture involves new components for the management of the handover. (ii) Proposal of an optimized network selection process divided into two stages, which are pre-selection and network selection. The pre-selection eliminates the non-potential candidate networks dependent on the mobility profile of the mobile node. The network selection process is based on multiple parameters using fuzzy logical model. (iii) The feasibility of the proposed RAT selection scheme in handover management is verified by extensive numerical simulations. The HO performances are evaluated in terms of average throughput of UE, average ping-pong HO rate, average handover failure HOF rate, and average HO delay. Compared with the performance of other existing HO strategies, our algorithms are more significant. The rest of the paper is organized as follows. Section 2 provides a background on the related research topic. Section 3 presents the system model. In Sect. 3.1, the MIH/SDN optimized handover scheme is described in detail. Section 4 presents simulation settings and provides the results and discussion. Finally, Sect. 5 concludes our study.

2 Related Works

In the conventional RSS-based solutions, HO is triggered when the user obtains a higher SINR from a nearby network while its current signal quality is degrading. In [8], the authors propose to use macro-assisted small cells based on the split of Control and User planes where small cells are considered as data only carriers. This solution improved the HO failure and energy consumption; however, it has a scalability problem at the macro cell level as this latter is expected to handle the control plane of a big number of femtocells. More intelligent techniques, such as Fuzzy rule-based algorithms, have been used in order to determine the best network, to reduce unnecessary HOs and improve throughput, taking into account bit error rate, delay, jitter and bandwidth as QoS parameters for HO decision [9]. However, the existing Fuzzy TOPSIS solutions deal with only QoS parameters and were not used in the case of HOD algorithms for HetNets with D2D communications. Other study has been carried out on the channel scanning method. In [10], the authors proposed an algorithm for vertical handover decision-making able to choose the best-optimized access network. This algorithm uses two major approaches namely the Bio-geographical Based Optimization (BBO) method and the Markov chain method. The proposed algorithm uses the IEEE 802.21 standard to acquire different handover decision information. It was proved via the simulation that this algorithm is able to select the best network candidate accurately based on the requirements of the connection in accordance with the requirements of the application and the preferences of the users.

Wang et al. [11] proposed an SDN-based architecture for future wireless networks. The proposed SDN controller is able to predict a user's movement path such that the relevant point of attachment is able to do handover in advance. Then, they proposed a novel self-healing approach to mitigate different failures occurring in backhaul to boost the robustness and reliability of future wireless networks. Experimental results verified that the proposed SDN-based architecture reduces handover latency compared to conventional scheme. However, in this paper, the authors don't provide details of their proposed handover scheme. Monira et al. in [12], proposed an SDN-based 5G handover solution to optimize handover delay by addressing the diversity of 5G networks with the help of SDN. In this work, the authors developed an authentication mechanism where a centralized authentication server establishes mutual trust among the domain controllers to ensure the credibility of the connected network components. However, this study did not deal the handover decision and network selection algorithms. Meanwhile, in [13–15], the researchers suggested various SDN-based HO algorithms for LTE-Advanced/5G HetNets. In [15], the authors proposed an SDN-based QoS VHD where the network with highest RSS and highest QoS score is selected for HO in dense HetNets. In this proposed solution, network context information such as RSS, Bandwidth, BLER, Jitter and Delay, user context information such as the user speed and service context information such as the application type are taken into consideration in the decision. Simulation results indicated that the proposed approach could reduce the signaling overhead, handover delay and handover dropping probability. However, additional decision parameters are required to ensure better QoS.

Hocine et al. in [16], dealt with the problem of energy saving during vertical handover in 5G communication systems. The proposed approach is based on MIH framework including modifications to make MIH more collaborative in energy saving. However, in this proposal the study of mobility management aspects is absent specially the handover decision and the best network selection. Khitem et al. in [17], proposed a new solution that involves resource allocation and vertical handover process based on MIH. In this solution, new elements are added in the core network to enhance the vertical handover procedure and to support the resource management process in next-generation wireless networks. Simulation results indicated that the proposed scheme could optimize the vertical handover process and improve the overall network performance, such as the fairness index, call blocking rate, etc. However, the handover decision strategy does not consider the users' preferences. Moreover, the execution of this scheme takes a lot of time, which causes long handover latency

3 Handover Approach in 5G HetNets

3.1 The Integrated 5G MIH-SDN Controller Architecture

In our approach, we focus on the adoption of the Logically Centralized Physically Distributed (LC-PD) [6] control plane architecture with MIH cooperation, as shown in Fig. 1, into the 5G mobile network. Our proposed framework incorporates an additional functional unit into SDN to assist in handover discovery, and the decision of candidate networks based on the networks' QoS parameters.

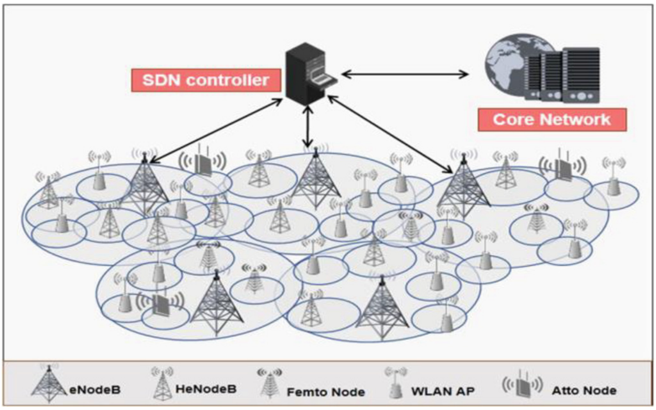


Fig. 1. SDN-based 5G Network Architecture

The proposed software-defined handover architecture is based on MIH and SDN to optimize handover in 5G networks. In addition, this architecture involves new component for the management of the handover. Figure 2 shows the proposed architecture, which consists of four units:

- MIH Enable Multi Interface Mobile Node MN: This logical device can use any of the available wireless networks supported by its interfaces. It consists of all functionalities necessary for an end user to access B5G network. The Negotiator Service

Monitor (NSM) define a mapping between the terminal context of the WiFi and LTE that enables the translation function to define values for WiFi context information-elements (resp., for LTE context information exchange) based on values related to a LTE association (resp., for WiFi association).

- SDN Controller with MIHF extension: We have introduced extension Handover Management Unit (HMU) for handover decision-making and network selection. HMU is introduced between the MIH layer and the upper layer in the protocol stack of the SDN controller. It can sense the MN's conditions in real time. Thus, it can use switch to collect the QoS information calculated by the network side before MN's computational capacity cannot support the services. Based on collected information, HMU determines in advance the need for the handover and chooses the best access network among heterogeneous networks, leading to the success of the handover process.
- Information Server (IS): providing static information about RATs like MAC address, location, and service provider's name.
- OpenFlow switches: associate diverse types of PoAs to the SDN controller. OpenFlow switches are responsible for the data forwarding. They consist of one or more flow tables. A flow entry consists of the source address, destination address, session id, port number, time, etc. [18]. An OpenFlow switch updates its flow tables based on instructions that are provided via an OpenFlow protocol from the SDN controller.

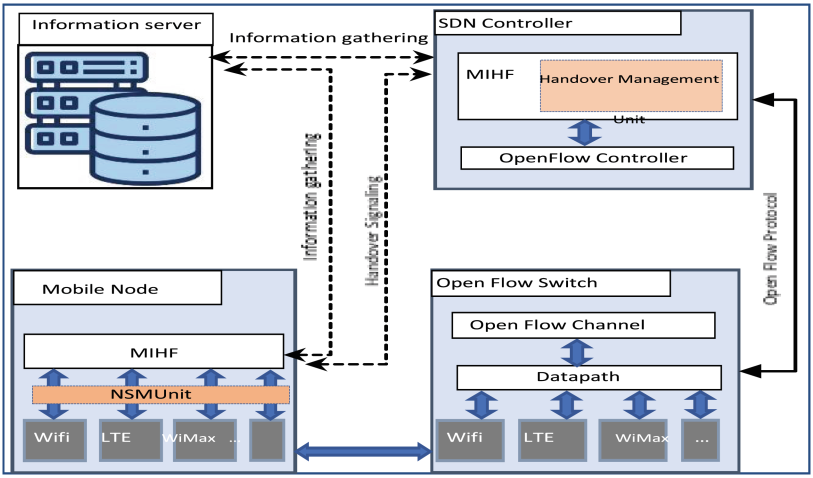


Fig. 2. Proposed Vertical Handover Framework

3.2 A Proposed Multi-criteria RAT Selection Scheme

The core idea behind the proposed architecture is based on mobility management to accommodate more calls while reducing the ping pong effect, handover delay and handover failure ratio and satisfying at the same time applications expectations. Thus, our proposed QoS management approach has been designed based on the following foundations:

Pre-selection Network: In this phase, the HMU performs the pre-selection process. The SDN controller first calculates the QoS score of the qualified networks. To judge whether the performance meets the user's requirement, QoS is quantitatively measured by some important parameters for user experience. Nonetheless, to define a cost function that takes into account MN's requirements, we involve another important metric which is the application type-based priority (P). Indeed, according to the temporal characteristics, the data in 5G can be divided into three categories [19]: Complex real-time data (CRT): whose time requirements are strict and have the highest priority, Soft real-time data (SRT)are more tolerant to changes in the timeout and Non-real-time data (NRT): non-real-time data is not time-sensitive and have the least priority. So, Q the denotation of QoS score can be computed as:

$$\begin{aligned} Q_{i,j} &= w_{RSRP} * \ln RSRP_{i,j} + w_d * \ln \frac{1}{d_{i,j}} + w_B * \ln B_{i,j} \\ &+ w_p \ln P_{i,j} + w_{BER} * \ln BER_{i,j} + w_{jit} * \ln JIT_{i,j} \end{aligned} \tag{1}$$

With:

$$w_{RSRP} + w_d + w_B + w_P + w_{BER} + w_{JIT} = 1$$

where W_{RSRP}, W_d, W_B, W_P, W_{BER} and W_{JIT} are the weighting factors of reference signal received power (RSRP), delay, bandwidth, application type-based priority, block error rate (BLER) and jitter, respectively. In (1), specially, factors that have negative effects on QoS are expressed in the form of reciprocal in "ln" function to reduce the QoS value.

After QoS calculation, a RAT is considered to be a candidate if it has a QoS score greater than the QoS score of the current RAT.

Network Selection and Fuzzy Logic Controllers: The SDN controller needs to select only the potential candidate based on multi-criteria metrics. The considered criteria are as follow. On the one hand, the network conditions which refer the characteristics of each RAT such as RSS and load. The RSS for non-3GPP networks or RSRP for 3GPP networks of the available RATs: is a measurement used for evaluating the signal quality of the neighbor base stations. The traffic load of the network: the traffic load of the cellular base stations and/or Wi-Fi Access Points (APs) (in terms of available bandwidth). It is the ratio between the number of resources used in the network and the total number of resources in the network for a period of time t. On the other hand, the mobile node conditions which refer to the parameters of each MN such as velocity and the requested type of service (ToS). To identify the ToS, we consider the two principles variables, requested Tolerated delay and Data-Rate (bit rate). The speed of the vehicle is a crucial decision parameter. Fast moving vehicle may cross over WLAN coverage rapidly. Thus, handing it over from a cellular network to a WLAN could cause quick successive handovers which may result in high signaling overheads and delays.

To identify the best suitable radio access network, we adopt fuzzy logic algorithmin our proposed scheme. Fuzzy logic is an ideal tool for dealing with uncertainty cases, when the inputs are rough estimated values [12]. The fuzzy logic controller is composed of four elements. These are fuzzification, rule base, inference mechanism and defuzzification. The proposed block diagram of a fuzzy logic control system is shown in Fig. 3. The fuzzifier under takes the transformation (fuzzification) of the input values to the degree that these values belong to a specific state (e.g., low, medium, high, etc.) as shown in Table 1. After that, the inference mechanism correlates the inputs and the outputs using simple "IF…THEN…" rules. Then, the output degrees for all the rules of the inference phase are being aggregated. The output of the decision making process, comes from the defuzzification procedure.

In the proposed scheme, fuzzy logic controller is applied on the following criteria: speed, type of service and network load. We assume three types of networks such as LTE-A macro cell, LTE-A femtocell and Wi-Fi. Every time that the algorithm is triggered, all the available eNB, Home eNB (HeNB) and APs are evaluated.

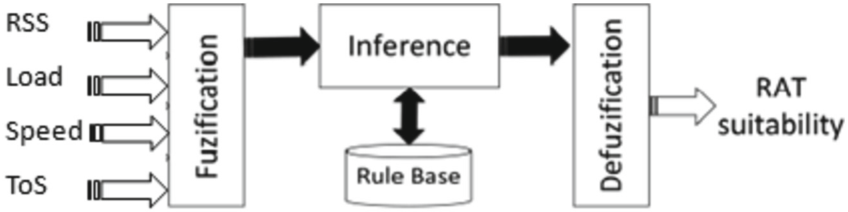


Fig. 3. Block diagram of fuzzy logic control system

For each output, there is a set of four triangular membership functions that represent the four following linguistic variables: "not acceptable NA", "probably not acceptable PNA", "probably acceptable PA" and "acceptable A". The fuzzy controller of the proposed scheme takes four inputs: (i) load factor, ld; (ii) Signal Strength factor, RSS; (iii) speed factor, SP; and (iv) data rate, DR; v) Tolerated delay, TD to identifying the type of service factor and output the selection decision, Sd. Below is the description of the structure of the proposed fuzzy logic controller.

Table 1. Values of decision variables.

Decision variables	Low	Medium	High
MN Speed (Km/h)	<40	From 40 to 60	From 80 to 140
RSS (dBm)	From −140 to −70	From −70 to −60	From −60 to −44
Load (%)	<30	From 30 to 70	From 70 to 100
Data rate (Mb/s)	<0,5	From 0,5 to 1,4	From 1,5 to 2
Tolerable delay (ms)	<150	From 150 to 30	From 300 to 500

Membership Functions: Trapezoidal and triangular membership functions are chosen for simplicity. The membership functions for input and output linguistic parameters are shown in Fig. 4. The values of the membership functions have been chosen based on commonly used values of membership functions in various literatures. For the fuzzy controller, the term sets for ld, SP, DR, TD, and Sd are defined as follows:

i) U(ld) = {Low, Medium, High}
ii) U(RSS) = {Low, Medium, High}
iii) U(SP) = {Low, Medium, High}
iv) U(DR) = {Low, Medium, High}
v) U(TD) = {Low, Medium, High}
vi) U(Sd) = {NA, PNA, PA, A}

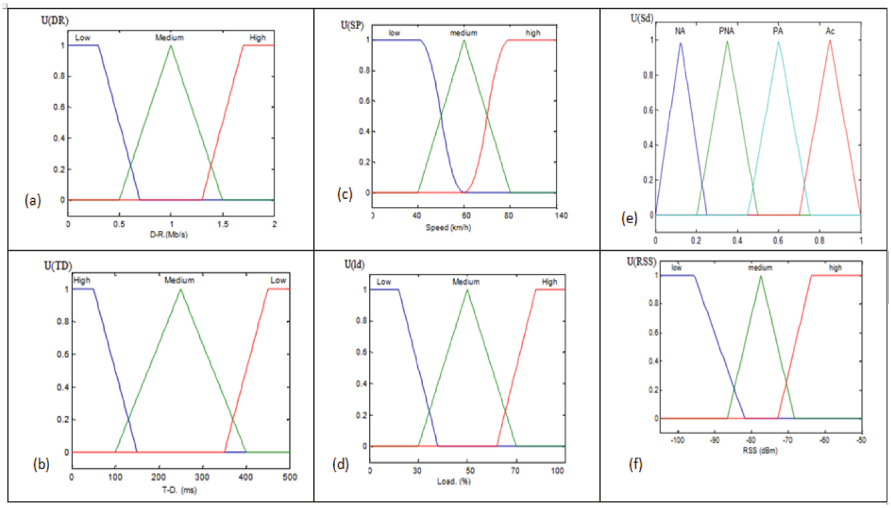


Fig. 4. Membership functions for (a) Data rate, DR (b) Tolerated delay, TD (c) Speed, SP (d) load factor, lc (e) selection decision (sd)

Examples of fuzzy inference system (FIS) rules:

(a) If D-R is low and T-D is medium then eNB is PA, HeNB is PA and AP is PA
(b) If D-R is high and T-D is high then eNB is A, HeNB is PA and AP is PNA
(c) If Speed is high then eNB is A, HeNB is PNA and AP is PNA

The strategy of the rules is the following. The RAT, which is characterized by high RSS and low load, is advantageous for the MN with high speed and QoS requirements choice. On the other hand, high mobility MN are preferably placed in larger cells and small cells are avoided to minimize the unnecessary handover. On the contrary, MN characterized by low or medium speed will be served by femtocells, in order to offload the traffic of the macro cells. Finally, the defuzzification process aggregates all the outcomes of all the rules and ends up to a certain degree of the output value, i.e., RAT suitability. The network with the highest RAT suitability will be selected. The suitability value ranges from 0 to 1 (0 to 100% respectively).

4 Performance Evaluation

In this section, the performance of the proposed handover management scheme is evaluated through a simulation analysis using MATLAB tool. This software is a suitable environment for our simulations because it has basic functions already present and necessary to evaluate the process of various traffic models. We adopt the conventional used hexagonal cells. The total number of macro cells is 24, and the radius of each is 1 Km. For the smaller cells, the total number equal to 500 APs distributed randomly in the network, and the radius of each varies from 50 to 200 m. Mobile Nodes are distributed randomly around AP, and they move randomly with a velocity varies from 5 km/h to 140 km/h. More details on the configuration parameters used in this simulation are given in Table 2. In order to make the simulation more accurate, we ran the simulation 10 times and averaged the results. To evaluate the performance of the proposed MIH/SDN-based vertical handover approach, we compare it by our previous handover mechanism based on utility function [17].

Table 2. Simulation Parameters.

Parameters	Values
Number of macrocells	24
Macrocell coverage	1000 m
Number of small cells	50–500
femtocell coverage	250
LTE BW/Data rate	20 MHz/100 Mbps
LTE range of RSRP	From −140 dBm to −44 dBm
Resource blocks (RBs)	100 RBs and 180 kHz per RB
802.11p BW/Data rate	10 MHz/6 Mbps
802.11p range of RSS	From −90 dBm to −30 dBm
Number of MN	125–1250
Vehicles speed (km/h)	20–140
Mobility model	Random walk model
Minimum association RSRP	−112 dBm

First, we measure the handover delay according to the increase of handover request arrival rate for the proposed and the existing solutions. Handover delay is the time it took from the disconnection of the MN from the ancient PoA until the MN correctly receives the first packet from the new PoA. Then, we measure the handover failure rate and ping-pong effect for all types of traffic classes.

Figure 5 presents the impact of the handover request arrival rate (request/second) and the delay occurred following the proposed and the existing handover approaches. We remark that our proposed handover algorithm gets significantly lower delay than the

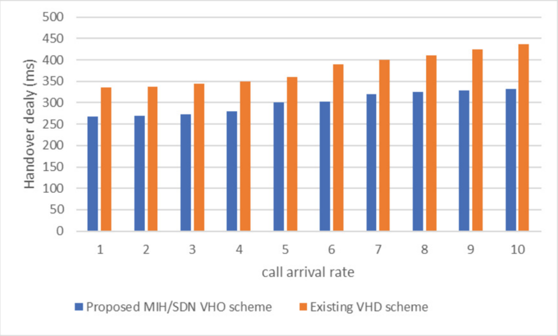


Fig. 5. Comparison of handover delay versus call arrival rate

other handover procedure. By analysing Fig. 5, we note that MIH/SDN-based vertical handover approach provides a 26% decrease in handover delay compared to the previous handover approach. This best result can be justified by the fact that the utilization of the SDN technology, when the density of networks is significantly important, reduces the complexity of the handover process.

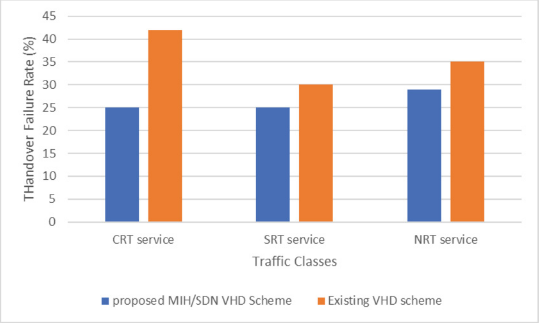


Fig. 6. Comparison of HO Failure Rate versus all Traffic Classes

Figure 6 depicts the handover failure ratios of the proposed and our previous handover mechanism. We observe from this figure, that the MIH/SDN-based vertical handover solution outperforms the previous approach by having less handover failure ratio. We show that the novel approach registers a decrease of 30% compared to the other solution.

5 Conclusion

In this work, we focus on proposing a solution to the problem of handover management in 5G HetNets. We have proposed a novel multi-criteria network selection mechanism. The objectives of the proposed approach are to decrease handover failure and delay and to distribute traffic load uniformly among available network to improve the average system resource utilization. The proposed algorithm is based on fuzzy logic scheme to support the decision making process. Simulation results demonstrate that, compared to existing works, the proposed approach significantly reduces the handover delay and failure.

References

1. Alraih, S., Shayea, I., Behjati, M., et al.: Revolution or evolution technical requirements and considerations towards 6G mobile communications. Sensors **22**(3), 744–762 (2022)
2. Shayea, I., Ergen, M., Azmi, M.H., Çolak, S., Nordin, A.R., Daradkeh, Y.I.: Key challenges, drivers and solutions for mobility management in 5g networks: a survey. IEEE Access **8**(1), 172534–172552 (2020)
3. ONF TR-502: SDN architecture Issue 1 June (2014)
4. Bannour, F., Souihi, S., Mellouk, A.: Distributed SDN control: survey, taxonomy, and challenges. IEEE Commun. Surv. Tuts. **20**(1), 333–354 (2018)
5. Tadros, C.N., Rizk, R.M., Mokhtar, B.: Software defined network-based management for enhanced 5G network services. IEEE Access **8**(1), 53997–54008 (2020)
6. Khan, S., Ali, M., Sher, N., Asim, Y., Naeem, W., Kamran, M.: Software defined networks (SDNs) and Internet of Things (IoTs): a qualitative prediction for 2020. Int. J. Adv. Comput. Sci. Appl. **7**(11), 385–404 (2016)
7. IEEE 802.21 Standard: IEEE standard for local and metropolitan area networks—part 21: media independent handover, IEEE STD 802. 21-2008 (2009)
8. Zhang, J., Feng, J., Liu, C., Hong, X., Zhang X., Wang, W.: Mobility enhancement and performance evaluation for 5G ultra dense networks. In: 2015 IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, pp. 1793–1798 (2015)
9. Hwang, W.-S., Cheng, T.-Y., Wu, Y.-J., Cheng, M.-H.: Adaptive handover decision using fuzzy logic for 5G ultra-dense networks. Electronics **11**(20), 1–15 (2022)
10. Baghla, S., Bansal, S.: An approach to energy efficient vertical handover technique for heterogeneous networks. Int. J. Inf. Technol. **10**(1), 359–366 (2018)
11. Lee, J., Yoo, Y.: Handover cell selection using user mobility information in a 5G SDN-based network. In: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, pp. 697–702 (2017)
12. Monira, S., Kabir, U., Jahan, M., Paul, U.: An efficient handover mechanism for SDN-based 5G HetNets. DUJASE **6** (2), 49–58 (2021)
13. Rizkallah, J., Akkari, N.: SDN-based vertical handover decision scheme for 5G networks. In: 2018 IEEE Middle East and North Africa Communications Conference, Jounieh, Lebanon, pp. 1–6 (2018)
14. Monir, N., et al.: Seamless Handover Scheme For MEC/SDN-based vehicular networks. J. Sens. Actuator Netw. **11**(9), 1–16 (2022)
15. Shah, S.D., Gregory, A.M., Li, A.S.R., Fontes, D.R., Hou, L.: SDN-based service mobility management in MEC-enabled 5G and beyond vehicular networks. IEEE Internet Things J. **9**(15), 13425–13442 (2022)
16. Hocine, A., Moez, E., Lyes, K.: Enhanced MIH (media independent handover) for collaborative green wireless communications. Int. J. Commun. Syst. **30**(7), 1–15 (2017)
17. Khitem, B.A., Zarai, F., Khdhir, R., Obaidat, M.S., Kamoun, L.: QoS aware predictive radio resource management approach based on MIH protocol. IEEE Syst. J. **12**(2), 1–12 (2018)
18. Sharma, V., You, I., Leu, F-Y., Atiquzzaman, M.: Secure and efficient protocol for fast handover in 5G mobile Xhaul networks. J. Netw. Comput. Appl. **102**(15), 38–57 (2018)
19. Liyanage, M., Porambage, P., Ding, A. Yi, K.: Driving forces for multi-access edge computing (MEC) IoT integration in 5G. ICT Express **7**(2), 127–137 (2021)

SSR-MGTI: Self-attention Sequential Recommendation Algorithm Based on Movie Genre Time Interval

Wen Yang[1,2](✉), Ruibo Yue[2], Yawen Chen[3], and Jun Zhao[4]

[1] Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering,
China Three Gorges University, Yichang 443002, China
yangwen0720@163.com

[2] College of Computer and Information Technology, China Three Gorges University,
Yichang 443002, China

[3] University of Otago, Dunedin 9016, New Zealand
yawen@cs.otago.ac.nz

[4] Hubei Three Gorges Polytechnic, Yichang 443000, China

Abstract. As an important part of the recommendation system, movie recommendation system can recommend movies to users accurately according to their preferences. Traditional movie recommendation systems simply treat user-movie interactions as a time-ordered sequence, without considering the time intervals between movies of the same genre. The genre time interval can reflect the user's preference for a particular genre and determine whether the algorithm can fully capture the user's interests and the time characteristics of the movie, which plays an important role in the accuracy of the movie recommendation. Therefore, in this paper, we propose a Self-Attention Sequential Recommendation algorithm based on Movie Genre Time Interval (SSR-MGTI). Specifically, a multi-head self-attention mechanism is used to model the same genre time interval information. Then, an absolute position is added to the multi-head self-attention mechanism model to solve the problem that multi-head self-attention mechanism does not consider the sequence. In addition, the convolutional neural network is used to convert the model from linear to non-linear and extract local information of user-movie interaction sequences. It is interesting to show that the proposed SSR-MGTI can accurately predict the movie that the user will watch next time. Experimental results on MovieLens and Amazon datasets demonstrate the superiority of our SSR-MGTI over state-of-the-art movie recommendation methods.

Keywords: Movie recommendation system · Genre time interval · Multi-head self-attention mechanism · Convolutional neural network

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 154–166, 2024.
https://doi.org/10.1007/978-981-99-8211-0_15

1 Introduction

Personalized recommendation is one of the most popular recommendation methods at present, which can tailor the recommended content to the users according to their unique preferences. Personalized recommendation algorithm mainly includes collaborative filtering-based recommendation algorithm, content-based recommendation algorithm, and sequential recommendation algorithm. The collaborative filtering-based recommendation algorithm [16] can recommend items according to a certain similarity (similarity between users or similarity between items) through the behavior of groups. The content-based recommendation algorithm [7] only utilizes the basic information (e.g., gender, age) of the user and the user-item interactions to forecast the user's preferences without taking into account the information of other users. The sequential recommendation algorithm [2, 19] attempts to predict the user's next new item by exploiting their historical behavior sequences. The sequential recommendation algorithm is particularly important in movie recommendation, since it can model the relationship between historically watched movies as a dynamic sequence to find the hidden information between movies.

Most of current sequential recommendation algorithms rank movies by interaction timestamps. However, these sequential recommendation algorithms only model the time series, ignoring the temporal information hidden in the timestamp itself. For example, Markov chain [9, 21, 22] assumes that the next movie is related to the previous movies, which only considers the time sequence, without considering time itself. Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) [21] model time series through hidden states. However, both CNN and RNN only compress temporal information into fixed hidden vectors, thus ignoring the temporal relationship between the various movies. The recently emerged "self-attention" mechanism (Self-Attention) can allocate different weights to the information according to their importance [1], but Self-Attention does not take the sequence of the series.

In this paper, we propose a Self-Attention Sequential Recommendation Algorithm based on Movie Genre Time Interval (SSR-MGTI). Specifically, we add absolute position to the multi-head self-attention mechanism to provide the sequential position of the movie. Then, we use the time interval between movies to represent the time information, and model the same genre of time interval of the user-movie interaction sequence to predict the next movie. In order to improve the model's fitting ability and highlight the importance of local preferences, we add CNN to improve the model's prediction ability. Finally, our contributions are summarized as follows:

- We model the same genre of time interval information in the user-movie interaction sequence.
- We use the multi-head self-attention mechanism to train the same genre of time interval by adding the absolute position information of the movie.
- We add the CNN to improve the stability and generalization ability of the model structure and capture the local information of user-movie interaction sequences.
- We carry out extensive experiments on MovieLens and Amazon datasets, which shows that our algorithm outperforms the state-of-the-art algorithms.

2 Related Work

During the past decades, extensive algorithms based on sequential recommendation have been proposed in the recommendation systems area. In general, existing methods can be categorized into three groups: general sequential recommendation method, deep learning-based sequential recommendation method, and self-attention-based sequential recommendation method.

General sequential recommendation algorithms include sequential pattern mining and Markov chain models. Yap et al. [23] proposed a recommendation framework based on personalized sequential pattern mining, which effectively learned important knowledge of user sequences. The FPMC model [15] proposed by Rendle et al. combines Matrix Factorization with the Markov Chain model, which incorporates both the common Markov chain and the normal matrix factorization model. It introduces modifications to the Bayesian personalized ranking framework recommended for the sequential basket. However, the Markov chain model can only capture the local information of the sequence, ignoring the global information related to the sequence.

In deep learning-based sequential recommendation method, RNN and CNN are most commonly used algorithms. RNN is inherently capable of processing sequence data. In order to solve the long-term dependency problem in RNN, two variants of RNN are generated, namely Long Short-Term Memory Neural Network (LSTM) [12, 17] and Gated Recurrent Unit (GRU) [5]. Duan et al. [6] proposed a new architecture based on LSTM for RNN ignoring collective dependencies due to the monotonous temporal relationship between items. The model adds the "Q-K-V" triplet to the recurrent unit to enhance the memory ability of LSTM, and proposes a "recovery gate" to solve the memory loss problem caused by the "forget gate". However, RNN is only suitable for long-term sequences. CNN can treat sequence as one-dimensional space and extract features from local sequence convolution. Tang et al. [18] proposed a convolutional sequence embedding model (Caser) to embeds recent sequence items into the "image" of time and latent space, which can use a convolution filter to turn the sequence into a local feature of the image. However, CNN is only good at capturing short-term sequences, which is not suitable for long-term sequences.

In recent years, self-attention mechanism has attracted great attention in the fields of natural language processing and computer vision. Chiang et al. [4] proposed a stacked attention network model, which stacks contextual item attention modules with multi-head attention modules and improves recommendation performance by using additional time information to model contextual items. Kang et al. [13] proposed a sequence model based on self-attention (SAS-Rec), which can be used to balance its sparse and dense data sets. Li et al. [14] proposed time interval-aware self-attention sequential recommendation (TiSAS-Rec), which models time intervals in user interaction sequences and uses the time interval information to predict the next item.

Although all the above methods can use timestamps to model time series, they seldom use the time information of timestamp itself, and do not take into account the time interval characteristics of the same genre. However, the genres of movies watched by users are different, and the time interval between them can better reflect the interests of users. In this paper, we will model the same genre time interval information of movies as the relationship between movies, and add absolute position information to the multi-head self-attention mechanism.

3 Problem Description of Movie Genre Time Interval

Since our recommendation algorithm is based on the movie genre time interval, we give the definition of movie genre time interval firstly in this section, followed by the problem description.

Movie Genre Time Interval (MGTI) refers to the time length between movies with the same category in the user-movie interaction sequence. The time interval between the same type of movies in the user-movie interaction sequence can reflect the user's recent preference for this type of movie. The smaller the time interval between two movies of the same type indicates that the user likes this type of movies more recently.

The MGTI can be modeled as following: Let $U = \{u_i|1 \leq i \leq N\}$, $V = \left\{v_j|1 \leq j \leq M\right\}$ and $G = \{g_k|1 \leq k \leq H\}$ represent the user set, the movie set and the genre set, respectively. Each movie in the movie set has a corresponding timestamp, which can be represented by $T = \left\{t_q|1 \leq q \leq M\right\}$. For a user u_i, the user-movie interactive sequence can be denoted by $S_i = \left(s_{i1}^{g_1}, s_{i2}^{g_2}, \ldots, s_{ij}^{g_k}, \ldots, s_{iM}^{g_H}\right)$, $i \in [1, N], j \in [1, M], k \in [1, H]$. MGTI can be denoted by $r_{cd}^{u_i} = s_{id}^{g_a} - s_{ic}^{g_b}$, where g_a and g_b represent type set and $g_a \cap g_b \neq \emptyset$. The absolute position sequence refers to the position of the movie in the user-movie interaction sequence, defined as $P = (1, 2, \ldots, M)$. At the time t, the model predicts the next movie based on the previous $t - 1$ movies and $r_{cd}^{u_i}$. The input of our model is a user-movie interaction sequence (S_i), an absolute position of the movie in the user-movie interaction sequence (P) and a genre time interval matrix of user-movie interaction sequence (R^i). The genre time interval matrix of user-movie interaction sequence can be denoted as below.

$$R^i = \begin{bmatrix} r_{11}^i \; r_{12}^i \cdots r_{1n-1}^i \; r_{1n}^i \\ r_{21}^i \; r_{22}^i \cdots r_{2n-1}^i \; r_{2n}^i \\ r_{31}^i \; r_{32}^i \cdots r_{3n-1}^i \; r_{3n}^i \\ \cdots\cdots\cdots \;\; \cdots \;\; \cdots \\ r_{n1}^i \; r_{n2}^i \cdots r_{nn-1}^i \; r_{nn}^i \end{bmatrix} \tag{1}$$

4 Multi-head Self-attention Mechanism Based on Movie Genre Time Interval

The overall framework of the model is shown in Fig. 1. This model includes three parts: 1) Embedding Layer: This layer vectorizes the sequence (S_i) and genre time interval matrix (R^i), and the absolute position of the movie in the user-movie interaction sequence (P), and embeds them in a low-dimensional space. 2) Multi-Head Self-Attention Mechanism Layer: This layer focuses on more relevant movies (i.e., genre time interval is shorter) and gives more weight to these movies. By assigning different weights to the movies, the recommendation results can be more personalized. 3) Convolutional Neural Network

Layer: This layer can convert the model from linear to non-linear and capture local information of user-movie interaction sequences. So it can improve the fitting ability of the model and the stability and generalization ability of the model structure.

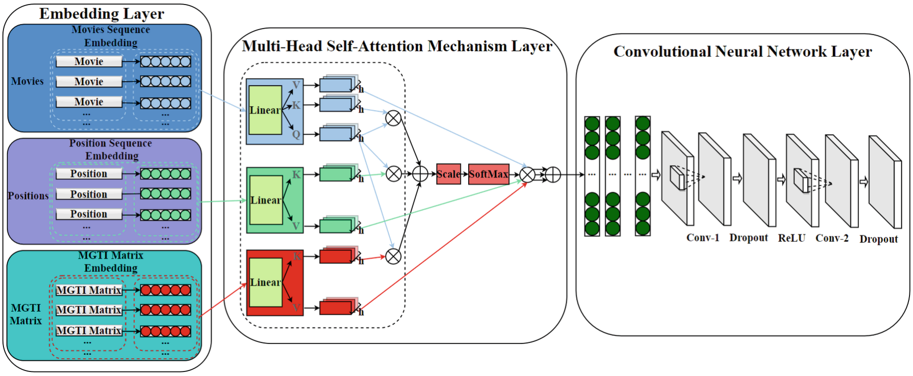


Fig. 1. Overall framework of Multi-Head Self-Attention Mechanism based on Movie Genre Time Interval.

4.1 Embedding Layer

We use the embedding layer to map the user-movie interaction sequence (S_i) to a lower dimensional space. The embedding layer uses the user interaction movie *ID* (the serial number of the movie in the dataset) as a numerical index to create an embedding matrix $E_S \in \mathbf{R}^{\mathbf{c}\times\mathbf{d}}$, where c is the dictionary size and d is the potential dimension. It maps user interaction movie *ID* to fixed-size vectors by embedding matrix (E_S), where S represents user-movie interaction sequence. So we will get the mapped low-dimensional matrix $O_S \in \mathbf{R}^{\mathbf{n}\times\mathbf{d}}$, where n is the maximum length of the sequence.

Similar to the user-movie interaction sequence, we use an embedding layer to map absolute position (P) to a lower dimensional space. The difference is that we will use two different embedding matrices to generate the keys and values in the multi-head self-attention mechanism without requiring additional linear transformations. Because the absolute position is a number, we use it as the numerical index to create embedding matrix $E_P^K \in \mathbf{R}^{\mathbf{c}\times\mathbf{d}}$ and $E_P^V \in \mathbf{R}^{\mathbf{c}\times\mathbf{d}}$ and map absolute position to fixed-size vectors by embedding matrix $\left(E_P^K, E_P^V\right)$, where P, K and V represent absolute position, keys and values of multi-head self-attention mechanism. Therefore, we can get the mapped low-dimensional matrices $O_P^K \in \mathbf{R}^{\mathbf{n}\times\mathbf{d}}$ and $O_P^V \in \mathbf{R}^{\mathbf{n}\times\mathbf{d}}$.

Likewise, we use genre time interval matrix (R^i) as numerical indexes to create embedding matrix $E_R^K \in \mathbf{R}^{\mathbf{c}\times\mathbf{d}}$ and $E_R^V \in \mathbf{R}^{\mathbf{c}\times\mathbf{d}}$ and map genre time interval matrix to fixed-size vectors by embedding matrix $\left(E_R^K, E_R^V\right)$, where R represents genre time interval matrix. Therefore, we can get the mapped low-dimensional matrices $O_R^K \in \mathbf{R}^{\mathbf{n}\times\mathbf{d}}$ and $O_R^V \in \mathbf{R}^{\mathbf{n}\times\mathbf{d}}$.

4.2 Multi-head Self-attention Mechanism Layer

The multi-head self-attention mechanism can give different weights to the movies according to the importance information in the time sequence, which works as following. Firstly, we calculate the attention weight α_{ij} by the following softmax function:

$$\alpha_{ij} = \frac{e^{v^{ij}}}{\sum_{k=1}^{n} e^{v^{ik}}}, \tag{2}$$

where v^{ij} is calculated using low-dimensional matrices $\left(O_S, O_R^K \text{ and } O_P^K\right)$.

$$v^{ij} = \frac{O_S W^Q \left(O_S W^K + O_R^K + O_P^K\right)}{\sqrt{d}}, \tag{3}$$

where $W^Q \in \mathbf{R}^{\mathrm{d}\times\mathrm{d}}$ and $W^K \in \mathbf{R}^{\mathbf{d}\times\mathbf{d}}$ are calculated by a fully connected layer, and W^Q and W^K are the coefficients of query and key. d is the dimension of the hidden layer. $\sqrt{d}$ is used to avoid large values of softmax.

Secondly, according to the attention weight α_{ij}, we calculate the final result of the multi-head self-attention mechanism model, that is, the weighted sum of value:

$$Z_i = \sum_{j=1}^{n} \alpha_{ij}\left(O_S W^V + O_R^V + O_P^V\right), \tag{4}$$

where $W^V \in \mathbf{R}^{\mathbf{d}\times\mathbf{d}}$ is calculated by a fully connected layer and W^V is the value of coefficient.

4.3 Convolutional Neural Network Layer

In order to improve the model's fitting ability and highlight the importance of local preferences, we add CNN to improve the model's prediction ability.

Firstly, we use a layer of 1D convolutional neural network for feature extraction:

$$F_i^1 = W^1 Z_i + b^1 \tag{5}$$

Secondly, The *ReLU* activation function is used after the first layer of convolutional neural network as follows:

$$F_i^2 = ReLU\left(F_i^1\right) \tag{6}$$

Finally, we use a layer of 1D convolutional neural network:

$$F_i^3 = F_i^2 W^2 + b^2 \tag{7}$$

where $W^1 \in \mathbf{R}^{\mathbf{d}\times\mathbf{d}}$ and $W^2 \in \mathbf{R}^{\mathbf{d}\times\mathbf{d}}$ are the parameter matrices of the first and the second convolutional neural network layers, respectively. $b^1 \in \mathbf{R}^{\mathbf{d}}$ and $b^2 \in \mathbf{R}^{\mathbf{d}}$ are the bias terms. F^1, F^2 and F^3 are the output of each layer.

5 Model Prediction

5.1 Prediction Layer

In the multi-head self-attention mechanism layer and the convolutional neural network layer, the increase in the number of model layers will lead to problems of overfitting, gradient disappearing and long training time. So we use layer normalization and *Dropout* regularization techniques to solve these problems:

$$g(x) = x + Dropout(g(LayerNorm(x))) \tag{8}$$

where $g(x)$ is multi-head self-attention mechanism layer or the convolutional neural network layer, and x is the input. We use the layer normalization technique on the input (x). Then we use the *Dropout* technique on the output of the multi-head self-attention mechanism layer or the convolutional neural network layer $(g(x))$. At last, we incorporate the input (x) into this result.

5.2 Loss Function

The binary cross-entropy loss function is commonly used in recommendation systems, which measures the predictive accuracy of the model by calculating the difference between the real label and the predicted label. It allows the model to converge fast and can be updated in real time without retraining the entire model. Therefore, we adopt the binary cross-entropy loss function as following:

$$-\sum_{S^u \in S} \sum_{t \in [1,2,\ldots,n]} \left[\log\left(\sigma\left(r_{o_t,t}\right)\right) + \log\left(1 - \sigma\left(r_{o'_t,t}\right)\right) \right] + \lambda \|\Theta\|_F^2 \tag{9}$$

where r_{o_t} represents positive output, $r_{o'_t}$ represents negative sampling, $\Theta = \left\{O_S, O_P^K, O_P^V, O_R^K, O_R^V\right\}$ is a low-dimensional matrix set of mapping, $\|\cdot\|_F$ represents *Frobenius* norm, and λ represents regularization coefficient.

6 Experimental Evaluation

In this section, we evaluate SSR-MGTI through extensive simulations. Firstly, we present the experimental setup, datasets and evaluation metrics in Sect. 6.1. Then, the results of SSR-MGTI and 7 recommendation baselines (GRU4Rec+ [11], NCF [10], Caser [18], SASRec [13], TiSASRec [14], LSPM [3], SSE-PT [20]) on Movielens and Amazon datasets are presented in Sect. 6.2. Finally, we also show the results of comparison under 3 different hyperparameters settings of SASRec, TiSASRec, and SSE-PT.

6.1 Experimental Configuration

Experimental Setup. All the following experiments were performed on NVIDIA RTX3090Ti GPU, and the code was implemented based on the Pytorch. The dropout rate of the Movielens dataset is 0.2 and the dropout rate of the Amazon dataset is 0.8.

Datasets. We evaluate our method on two datasets from two platforms. Dataset statistics of two datasets after preprocessing are shown in Table 1. MovieLens is the dense dataset which has more average actions with fewer users and movies. Amazon is the sparse dataset which has the fewer actions per user and movie.

- **MovieLens**: This dataset is often used in the recommendation system competition. We will use a version with 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000. (MovieLens-1M).
- **Amazon**: This dataset records users' comments on Amazon website. It is the classic dataset of the recommendation system and Amazon has been updating this dataset. We will use the Video_Games dataset from the 2014 release. (Amazon Video_Games)

Table 1. Dataset statistics (after preprocessing)

Dataset	#Users	#movies	avg.actions/user	avg.actions/movie
Movielens	6040	3416	163.50	289.09
Amazon	31013	23715	7.26	9.50

Evaluation Metrics. We use two common Top-N metrics to evaluate the performance of our method: Hit Rate@10 and NDCG@10 [8, 10]. Hit Rate@10 is mainly concerned about whether the movie that users like is recommended, which emphasizes the "accuracy" of prediction. NDCG@10 is more concerned about the "order", which emphasizes whether the recommended movie appears in a higher position in the recommended sequence.

6.2 Results and Analysis

Results on Different Recommendation Methods. We study the performance of our proposed model SSR-MGTI with all baselines on two real-world datasets. Table 2 shows the experimental results of all the methods. It can be observed that:

(1) SSR-MGTI can always achieve the best performance regardless of datasets and evaluation metrics, which can gain 20.13% Hit Rate and 41.06% NDCG improvements on average compared with other methods.
(2) SSR-MGTI has a significant improvement in the NDCG metric on both sparse and dense datasets, which can achieve performance up to 25.65% on dense dataset and up to 56.46% on sparse dataset.

Ablation Study. We conduct experiments by removing position, CNN, dropout and layernorm separately to demonstrate the role of each component of our model. Table 3 shows the performance of the two datasets with the best set of hyperparameters.

For the Movielens dataset, removing the added components clearly shows that the recommendation performance has declined, especially for the dropout component. The

Table 2. Performance of different recommendation methods. The best performance in each row is boldfaced (higher is better), and the second best method in each row is underlined. Improvements are shown in the last column.

Dataset	Metric	GRU4Rec+	NCF	Caser	SASRec	TiSASRec	LSPM	SSE-PT	SSR-MGTI	Improvement
Movielens	Hit Rate@10	0.6522	0.6954	0.7517	0.8174	0.8311	0.8303	0.8371	**0.8760**	**13.24%**
	NDCG@10	0.4334	0.5193	0.5011	0.5786	0.6108	0.6240	0.6160	**0.6970**	**25.65%**
Amazon	Hit Rate@10	0.3971	0.6642	0.4474	0.7551	0.7327	0.6157	0.7466	**0.7909**	**27.01%**
	NDCG@10	0.2321	0.4632	0.2661	0.5425	0.5256	0.4210	0.5448	**0.6695**	**56.46%**

Table 3. Ablation analysis (NDCG@10) on two datasets. Performance better than the default version is boldfaced. '↓' indicates performance drop.

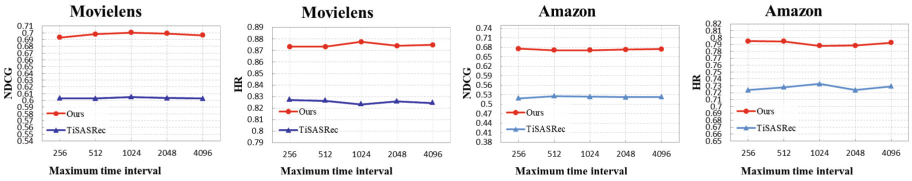
Dataset	Default	Remove Position	Remove CNN	Remove Dropout	Remove Layernorm
Movielens	0.6970	0.6846 ↓	0.6865 ↓	0.6521 ↓	0.6890 ↓
Amazon	0.6695	**0.6795**	**0.6714**	0.5530 ↓ (overfitting)	0.6410 ↓

position, CNN and layernorm components can modify the model to some extent. Adding the absolute position of the user-movie interaction sequence can be combined with the relative position of the type time interval to better capture the connection between movies. Adding the CNN component can not only convert the linear model into a non-linear model, but also extract short-term preferences. Removing the layernorm component will reduce the generalization ability of the model and may also cause gradient disappearance and gradient explosion problems, so it is lower than the default model metric. Removing the dropout component will cause the recommendation metric of the model to drop sharply, which shows that the dropout component greatly affects the recommendation performance of the model. It can also be seen from the output results of the model that the evaluation metric fluctuates around 0.6500, which indicates that the model overfitting problem is not obvious on the dense dataset.

For the Amazon dataset, removing the dropout and layernorm components will cause a severe performance drop, especially removing dropout will cause overfitting problems (evaluation metric gradually decrease). Removing the layernorm component will seriously hurts model performance for sparse datasets. Removing the position and CNN components is more suitable for sparse datasets. Since the sparse data interaction sequence is less, the absolute position plays a little role and there is little difference between the short-term feature and the long-term feature. Adding position and CNN may easily increase the noise of the data, so the recommendation performance of the model will be improved.

Comparison of 3 Different Hyperparameters Settings. We compared 3 hyperparameters (i.e., maximum genre time interval, maximum sequence length n and number of heads of attention) based on the SASRec, TiSASRec, SSE-PT and SSR-MGTI models, and the maximum movie genre time interval was only compared with TiSASRec.

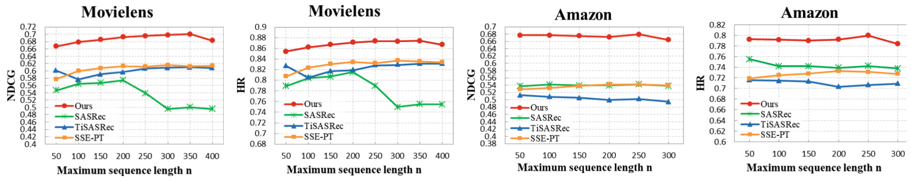
(1) Influence of the maximum genre time interval. The maximum genre time interval refers to the maximum value of the set genre time interval. Since maximum genre time interval may replace the computed movie genre time interval for training, it has a great impact on the recommendation performance. Figure 2 shows the effect of maximum time interval between TiSASRec and SSR-MGTI on the two datasets. From Fig. 2, we can see that SSR-MGTI can always achieve better performance than TiSASRec under different datasets, which can gain 7.41% Hit Rate and 22.16% NDCG improvements on average. This is because our method adds movie genre features to the time interval, which enables the model to accurately capture temporal information between movie genres in user-movie interaction sequences.



(a) (b) (c) (d)

Fig. 2. Effect of maximum genre time interval on ranking performance

(2) Influence of maximum sequence length *n***.** The maximum sequence length *n* refers to the length of the user-movie interaction sequence, which determines the number of data that can be added to the model for training. From the results, we can see that SSR-MGTI gains 7% Hit Rate and 18.99% NDCG improvements on average in the Movielens dataset. As shown in Fig. 3 (a) and (b), the recommendation performance of the TiSASRec, SSE-PT and SSR-MGTI models increases with the increase of *n*. But the SASRec rises firstly and then declines. It may because SASRec uses fewer features. So a large number of 0 are filled with *n* increases, resulting in a decline in recommendation performance. For the Amazon dataset, SSR-MGTI gains 8.95% Hit Rate and 28.04% NDCG improvements on average. As shown in Fig. 3 (c) and (d), the recommendation performance of the four models decreases slightly with the increase of *n*.

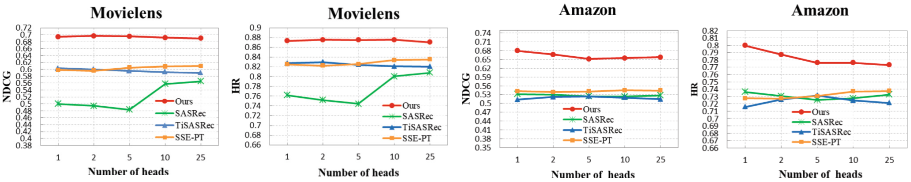


(a) (b) (c) (d)

Fig. 3. Effect of maximum sequence length *n* on ranking performance

(3) Influence of the number of heads of multi-head self-attention. Since number of heads of multi-head self-attention can enable the network to capture the user's interests from multiple aspects, it has a great impact on the recommendation performance. As shown in Fig. 4 (a) and (b), the recommendation performance of the SASRec and SSE-PT model increases significantly. But the recommendation performance of the TiSASRec and SSR-MGTI models firstly increases then decreases. This is because when the number of heads is too large, a large number of parameters will be generated to cause overfitting problems. For the Movielens dataset, SSR-MGTI gains 8.21% Hit Rate and 21.74% NDCG improvements on average. For the Amazon dataset (Fig. 4 (c) and (d)), SSR-MGTI gains 7.37% Hit Rate and 25.12% NDCG improvements on average.

It can be seen that the performance of SSE-PT is slightly improved. But the performance of SSR-MGTI, TiSASRec and SASRec decreases as the number of heads increases. This is because the dataset is relatively sparse, the hidden information of the data is relatively fewer.



(a) (b) (c) (d)

Fig. 4. Effect of the number of heads of multi-head self-attention on ranking performance

7 Conclusion

In this paper, we proposed a Self-Attention Sequential Recommendation Algorithm based on Movie Genre Time Interval (SSR-MGTI). Firstly, we give the definition of Movie Genre Time Interval (MGTI), based on which a multi-head self-attention mechanism is modeled. Then, we add the absolute position of the movie in the user-movie interaction sequence to make up for the deficiency of the multi-head self-attention mechanism. In addition, we use a convolutional neural network to convert the model to non-linear and extract local information of user-movie interaction sequences. The experiment results show that our proposed recommendation scheme can achieve 20.13% and 41.06% improvement in HR@10 and NDCG@10 respectively over other state-of-the-art schemes in terms of dense (MovieLens) and sparse (Amazon) datasets.

References

1. Ashish, V.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30, p. I (2017)
2. Chang, J., et al.: Sequential recommendation with graph neural networks. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 378–387 (2021)
3. Chen, J., Jiang, L., Sun, H., Ma, C., Liu, Z., Zhao, D.: LSPM: joint deep modeling of long-term preference and short-term preference for recommendation. In: Gedeon, T., Wong, K., Lee, M. (eds.) Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, 12–15 December 2019, Proceedings, Part IV. CCIS, vol. 1142, pp. 237–246. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36808-1_26
4. Chiang, J.H., Ma, C.Y., Wang, C.S., Hao, P.Y.: An adaptive, context-aware, and stacked attention network-based recommendation system to capture users' temporal preference. IEEE Trans. Knowl. Data Eng. **35**(4), 3404–3418 (2022)
5. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
6. Duan, J., Zhang, P.F., Qiu, R., Huang, Z.: Long short-term enhanced memory for sequential recommendation. World Wide Web **26**(2), 561–583 (2023)
7. Fkih, F.: Similarity measures for collaborative filtering-based recommender systems: review and experimental comparison. J. King Saud Univ.-Comput. Inf. Sci. **34**(9), 7645–7669 (2022)
8. He, R., Kang, W.C., McAuley, J.: Translation-based recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, pp. 161–169 (2017)
9. He, R., McAuley, J.: Fusing similarity models with Markov chains for sparse sequential recommendation. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 191–200. IEEE (2016)
10. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
11. Hidasi, B., Karatzoglou, A.: Recurrent neural networks with top-k gains for session-based recommendations. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 843–852 (2018)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
13. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 197–206. IEEE (2018)
14. Li, J., Wang, Y., McAuley, J.: Time interval aware self-attention for sequential recommendation. In: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 322–330 (2020)
15. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 811–820 (2010)
16. Shen, J., Zhou, T., Chen, L.: Collaborative filtering-based recommendation system for big data. Int. J. Comput. Sci. Eng. **21**(2), 219–225 (2020)
17. Sundermeyer, M., Schlüter, R., Ney, H.: LSTM neural networks for language modeling. In: Thirteenth Annual Conference of the International Speech Communication Association (2012)
18. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 565–573 (2018)

19. Wang, D., Xu, D., Yu, D., Xu, G.: Time-aware sequence model for next-item recommendation. Appl. Intell. **51**, 906–920 (2021)
20. Wu, L., Li, S., Hsieh, C.J., Sharpnack, J.: SSE-PT: sequential recommendation via personalized transformer. In: Proceedings of the 14th ACM Conference on Recommender Systems, pp. 328–337 (2020)
21. Xu, C., et al.: Long-and short-term self-attention network for sequential recommendation. Neurocomputing **423**, 580–589 (2021)
22. Yan, C., Wang, Y., Zhang, Y., Wang, Z., Wang, P.: Modeling long-and short-term user behaviors for sequential recommendation with deep neural networks. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2021)
23. Yap, G.E., Li, X.L., Yu, P.S.: Effective next-items recommendation via personalized sequential pattern mining. In: Lee, S.G., Peng, Z., Zhou, X., Moon, Y.S., Unland, R., Yoo, J. (eds.) Database Systems for Advanced Applications: 17th International Conference, DASFAA 2012, Busan, South Korea, 15–19 April 2012, Proceedings, Part II 17, vol. 7239, pp. 48–64. Springer, Cham (2012). https://doi.org/10.1007/978-3-642-29035-0_4

Fine Time Granularity Allocation Optimization of Multiple Networks Industrial Chains in Task Processing Systems

Pan Li[1], Kai Di[2](✉), Xinlei Bai[1], Yuanshuang Jiang[2], and Fulin Chen[1]

[1] School of Cyber Science and Engineering, Southeast University, Nanjing 211189, Jiangsu, China

[2] School of Computer Science and Engineering, Southeast University, Nanjing 211189, Jiangsu, China

dikai@seu.edu.cn

Abstract. As the industrial division of labor becomes increasingly specialized, various collaborative relationships between industrial chains develop, forming complex multi-networks. In the task processing system of multiple network industrial chains, there are dynamic online tasks. The arrival and deadline of these tasks cannot be accurately predicted. Therefore, it is necessary to divide the scheduling into finer time granularity to improve the response speed, efficiency, and timeliness, while ensuring the task completion rate and minimizing the task cost. In this paper, we study the characteristics of online tasks in multiple networks industrial chains and design a corresponding online scheduling framework. We analyze the arrival of online tasks in real-world scenarios and propose a passive scheduling algorithm based on the characteristics of different scenarios. The algorithm is tested on several sets of simulated data. Compared with previous heuristic algorithms, our algorithm can achieve better results in terms of task completion time, energy cost, and task completion rate in scenarios with fine time granularity.

Keywords: Multiple networks industrial chains · Task allocation · Fine time granularity

1 Introduction

With the development of industrial intelligence and information technology, the division of labor in the industrial chain is becoming finer. There are various collaborative relationships among product agents, which have the characteristics of multiple networks. In actual industrial scenes, sudden tasks often occur in the upstream and downstream

This work was supported by the National Key Research and Development Program of China (No. 2022YFB3304400), the National Natural Science Foundation of China (Nos. 62076060, 61932007), the Key Research and Development Program of Jiangsu Province of China (No. BE2022157), and the Defense Industrial Technology Development Program (JCKY2021214B002).

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 167–179, 2024.
https://doi.org/10.1007/978-981-99-8211-0_16

networks of product agents, such as urgent orders or product repairs, which need to be responded to in a short period of time [1]. Therefore, it is necessary to adopt a fine time granularity task assignment method that decomposes the task assignment calculation into short time slots. Fine time granularity task allocation optimization can effectively address the uncertainty of dynamic task arrival and optimally meet the requirements of online task allocation and efficiency [2].

This paper proposes an optimal strategy for task allocation at the fine time granularity level in multiple networks industrial chains, and presents a corresponding allocation strategy for online task scenarios. By adopting a passive allocation strategy, the existing tasks can be executed without interference while optimizing the task allocation, thus reducing the impact of online tasks on the system allocation results, and minimizing the system task waiting time. The application of this strategy can improve task completion rates and optimize electricity costs.

The main contributions of this paper are summarized as follows:

1) This paper establishes a systemic model for the online allocation optimization problem of multi networks industrial chains task processing systems at the fine time granularity time granularity and proposes an online allocation framework based on multiple networks industrial chain task processing systems. The framework consists of three components: the task preprocessing module, task layer calculation and allocation, and task execution and product agent information feedback;
2) This study conducts a systematic analysis of real-world scenarios. In this paper, we propose a passive online scheduling algorithm based on the task's latest allowable start time and offset error to minimize the impact of online tasks on system scheduling;
3) In this paper, the performance of the proposed algorithm has been tested and compared with other scheduling algorithms. The experimental results show that the algorithm proposed in this paper performs better in this scenario.

2 Related Work

This paper studies the problem of task allocation optimization in multiple networks industrial chains task processing systems under dynamic costs. According to the research perspective, related work can be divided into the following aspects: optimization of energy cost and research related to task and resource allocation.

Energy Cost Optimization: Currently, the world is facing an energy resource crisis, with increasing energy costs. Improving energy efficiency to reduce energy utilization costs has become a popular research area [3–5]. In multi-departmental collaborative industrial production, how to balance production efficiency and energy costs is the mainstream research topic [6–8].

Traditionally, research has mainly focused on task allocation optimization in a single system or network structure [9–11]. However, this paper focuses on an online task processing system in a multiple networks industrial chain [12], which is composed of multiple interlaced, related, or overlapping industrial chains. There are complex relationships [13] and interactions between these industrial chains, including mutual influence characteristics in data and energy.

Task Resource Scheduling: The problem of task and resource allocation has a wide range of applications in real life, such as product agent allocation planning [14, 15], project schedule allocation and arrangement [16–18], and cloud computing workflow allocation and distribution [19, 20].

In summary, traditional research in the field of task and resource allocation has considered optimizing the task execution time under various constraints, but it has paid less attention to modelling and analyzing the power cost changes of task allocation and fine time granularity task execution across systems and network layers, which is not entirely suitable for the multi-networked environment with dynamic cost and time-of-use electricity pricing studied in this paper. It cannot fully reflect the impact that multiple networks have on task allocation.

3 Problem Formulation and Analyses

3.1 Fine Time Granularity Online Task Scheduling Model

In a multiple networks industrial chain task processing system, there are many fine time granularity real-time tasks that arrive online, and the system will not know the characteristics of the tasks, such as the task arrival time, deadline, and task execution structure, before they arrive. When a task arrives, its relevant characteristics are made available to the current system. Different tasks have different time characteristics, so fine time granularity assignment is needed to deal with the dynamics of online tasks.

For the task processing system of multiple networks industrial chains, the fine time granularity dynamic online task set can be expressed as $W = \{w_1, w_2, \ldots, w_n\}$, and $w_j \in W$ represents the j-th online task in this online task set, which can be described as $w_j = \{a_j, d_j, G_j\}$ by a triple, where a_j, d_j, G_j are the arrival time, deadline and network structure of the online task w_j respectively. An online task is successfully executed only if all its subtasks finish before their deadline.

For the task execution network structure, it is further modelled as a directed acyclic graph (DAG), such as $G_j = \{T_j, E_j\}$. Where $T_j = \left\{t_{j1}, t_{j2}, \ldots, t_{j|T_j|}\right\}$ represents the set of subtasks of online task w_j, $t_{jk} = \{u_{jk}, ptime_{jk}\}$ represents the k-th subtask of online task w_j that needs to be executed on industry chain u with execution time $ptime_{jk}$, and $E_j \subseteq T_j \times T_j$ represents the set of directed edges of online task j. $e^j_{pk} \in E_j$ means that there is a task dependency between two subtasks (t_{jp}, t_{jk}) in the online task w_j, where subtask t_{jp} is called the direct predecessor of subtask t_{jk}, and subtask t_{jk} is called the direct successor of subtask t_{jp}.

3.2 Task Execution Product Agents' Model

In the current task processing system, there are usually multiple sub-task processing systems with different functions [21]. These task processing systems with different functions are coupled and associated to form a task processing system under multiple industrial chains. For any sub-task processing system, it contains a certain number of isomorphic product agents used to perform tasks, and these product agents together constitute an industrial chain of task processing. s_i^k represents the product agents with label k of industrial chain i, and the parameter m is used to represent the number of industrial chains in the multiple networks industrial chain task processing system, and the parameter $i \in \{1, 2, \ldots, m\}$ represents the index label of the industrial chain. Machines of product agents in different industrial chains usually have different functions and different powers, that is, the product agent's machines in different industrial chains have different energy consumption per unit time when performing tasks. Parameter $l \in \{1, 2, \ldots, mQ_i\}$ represents the number of product agents in industry chain i, where mQ_i denotes the number of product agents in this industry chain i.

3.3 Problem Formulation

The purpose of scheduling optimization is to allocate tasks to specific product agents in the corresponding industrial chain according to demand so that tasks can be completed in order and within the deadline while reducing the power cost required for task execution. $\Pi_j = \left\{\pi_1, \pi_2, \ldots, \pi_{|T_j|}\right\}$ is used to represent the scheduling result of online task w_j, where $\pi_k = \langle t_{jk}, s_i^l, ST_{jk}, ET_{jk}\rangle$ represents the mapping between the subtask t_{jk} of online task w_j and the specific product agents executing the task, meaning that the subtask t_{jk} of online task w_j is executed on the l-th product agents in industry chain i, and the start execution time is ST_{jk} and the end time is ET_{jk}.

The fine time granularity online scheduling optimization problem of multiple industrial chain task processing system (FTGOSO) can be defined as follows: given a multiple industrial chain task processing system G in a fine time granularity environment, tasks arrive online dynamically, and a real-time task scheduling policy Π is designed to minimize the power cost of task execution and improve the success rate of task execution while satisfying constraints. The problem is formally defined as follows:

$$Minimize \sum_{j=1}^{n} EC_j \qquad (FTGOSO) \tag{1}$$

s.t.

$$x_{jk}^{il} = \begin{cases} 1, \textit{Task } t_{jk} \textit{ is executed on the } l - th\textit{firm} \\ \textit{of industry chain } i \\ 0, \textit{ others} \end{cases} \tag{2}$$

$$\sum_{i=1}^{m} \sum_{l=1}^{mQ_i} x_{jk}^{il} = 1, \forall t_{jk} \in w_j \tag{3}$$

$$EC_{jk} = \int_{ST_{jk}}^{ET_{jk}} e(t) \cdot P_i \tag{4}$$

$$EC_j = \sum_{k=1}^{|T_j|} EC_{jk} \tag{5}$$

$$ST_{jk} = \max\left\{MT_i^l, ET_{jp}\right\}, \forall e_{pk}^j \in E_j \tag{6}$$

$$ET_{jk} = ST_{jk} + ptime_{jk} \tag{7}$$

$$ET_j \leq d_j, \forall w_j \in W \tag{8}$$

$$ST_j \geq a_j, \forall w_j \in W \tag{9}$$

Among them, (FTGOSO) represents the optimization objective, which is to minimize the execution energy cost of the task. Constraint (2) is used to represent the mapping relationship between the task and the executing product agents. A value of 1 indicates that the task is executed on the product agents. Constraint (3) indicates that only one product agents can be selected to execute the subtasks in a single industrial chain. Constraint (4) shows the cost calculation of a single task, where $e(t)$ represents the electricity price in the current period, which shows periodic fluctuations over time, and P_i represents the power of the product agents in the industrial chain i when performing the task. Constraint (5) indicates the calculation of the total cost of the entire online task; Constraint (6) indicates that the start time of the online task needs to meet the product agents availability time constraint and task arrival time constraint, task t_{jk} is executed on the l-th product agents in the industrial chain i, MT_i^l represents the earliest idle time of the product agents, ET_{jp} denotes the end time of the precursor task t_{jp} of task t_{jk}. Constraint (7) represents the actual end time constraint of the task; Constraint (8) and (9) indicate that the start time and end time of a task should satisfy its arrival time and deadline constraints.

4 Algorithm Design

4.1 Online Scheduling Framework

The fine time granularity online scheduling system cannot accurately know the arrival time and execution structure of the task set, so it cannot allocate the task set in advance. It needs to dynamically allocate the task set in real time according to the work situation of the task executing product agents. In this section, referring to previous research on the online scheduling framework [22], a fine time granularity online scheduling framework under multiple networks industrial chains is defined, which is used to execute fine time granularity real-time online tasks in the task processing system of multiple networks industrial chains. The online scheduling framework is shown in Fig. 1, which can be divided into three modules: Task preprocessing module, task calculation and scheduling module, task processing and product agents' information feedback module.

Definition 1 Ready Task. : *An online task consists of multiple subtasks, and a subtask is ready if it does not have any predecessor task, that is, $pre\left(t_{ij}\right) = \emptyset$, or its predecessor task has completed execution. A task is marked as ready means that the task can start execution at any time.*

The operation process of the online scheduling framework under the multiple networks industrial chains is as follows:

1) Online tasks arrive;
2) calculate the scheduling interval of each sub-task of the task, and allocate the sub-tasks to each industrial chain;
3) Subtasks are added to the task pool of each industry chain and sorted and waited according to the scheduling rules;
4) When the product agents are idle, select tasks to execute according to the rules;
5) Feedback the task execution time and update the adjustable interval of the next subtask;
6) The task is completed, the cost is calculated and the result is output, and the end is reached.

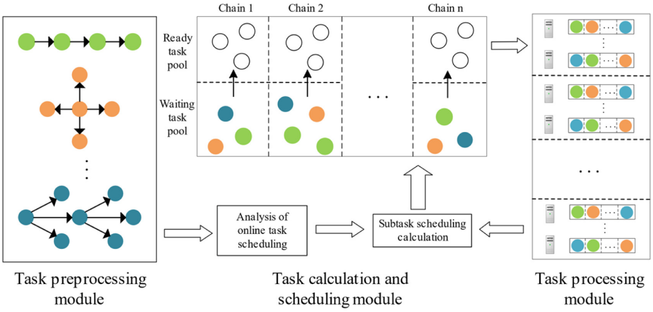

Chain 1
Chain 2
Chain n
Ready task pool
Waiting task pool
Analysis of online task scheduling
Subtask scheduling calculation
Task preprocessing module
Task calculation and scheduling module
Task processing module

Fig. 1. Online scheduling framework.

4.2 Passive Scheduling Strategy

In view of the online arrival of fine time granularity tasks with low priority, it is necessary to design relevant algorithms to determine the deadline of each sub-task for the task and arrange the execution product agents according to the industry chain required for the execution of the sub-task. In order to reduce the impact of dynamic online task arrival on the existing scheduling, based on the online scheduling framework in Sect. 4.1, this section proposes a passive online scheduling algorithm based on the latest start time [23] (AS-Late-As-Possible, ALAP) and offset error. The algorithm is mainly divided into multiple networks industrial chain online task scheduling analysis module and hierarchical sub-task execution scheduling module.

4.3 Online Task Allocation Analysis Module

ALAP represents the latest start execution time of a subtask that can be delayed without affecting the critical path [23]. It can be used to measure the adjustable time range of a task. The $alap_{jk}$ of subtask t_{jk} of task T_j can be calculated as follows:

$$alap_{jk} = \begin{cases} d_j - ptime_{jk}, & suc\left(t_{jk}\right) = \emptyset \\ \min\limits_{t_{js} \in suc\left(t_{jk}\right)} alap_{js} - ptime_{jk}, & suc\left(t_{jk}\right) \neq \emptyset \end{cases} \tag{10}$$

Definition 2 Critical Path: *The critical path in the online optimization problem of task scheduling in the task processing system of multiple networks industrial chains (Multi-Chains-Critical-Path, MCCP) is defined as the longest path from the start task to the end task of online task T_i.*

Let bl_{jk} denote the longest path from task t_{jk} to task t_{jexit}, then:

$$bl_{jk} = \begin{cases} ptime_{jk}, & suc\left(t_{jk}\right) = \emptyset \\ \max\limits_{t_{js} \in suc\left(t_{jp}\right)} bl_{js} + ptime_{jk}, & suc\left(t_{jk}\right) \neq \emptyset \end{cases} \tag{11}$$

Therefore, the critical path of T_j in the task online scheduling optimization problem of task processing system with multiple networks industrial chains is as follows.

$$MCCP_j = bl_{j1} \tag{12}$$

By using the critical path, the deadline of the subtask can be assigned according to the length of the execution time of the subtask to the critical path. The deadline d_{jk} of the subtask t_{jk} of the task T_j is:

$$d_{jk} = d_j \times \frac{alap_{jk} + ptime_{jk}}{MCCP_j} \tag{13}$$

This method will result in the subtasks at the top of the sequence having more adjustable time. To solve this problem, this paper introduces an offset error Δ.

Definition 3 Offset Error $\boldsymbol{\Delta}$: *The offset error in the task online scheduling optimization problem of multiple networks industrial chain task processing system is defined as the difference between the online task deadline and the critical path time, and is calculated as follows:*

$$\Delta_j = d_j - a_j - MCCP_j \tag{14}$$

The sub offset error Δ_{jk} of the subtask t_{jk} of task T_j is as follows.

$$\Delta_{jk} = \Delta_j \cdot \frac{ptime_{jk}}{MCCP_j} \tag{15}$$

Through the offset error of the subtask, we can obtain the $\Delta{-}alap_{jk}$ value of the subtask t_{jk}, which is used to represent the latest start time of the subtask t_{jk} to perform the scheduling optimization. It is calculated as follows:

$$\Delta{-}alap_{jk} = \begin{cases} d_j - ptime_{jk} - \Delta_{jk}, & suc(t_{jk}) = \emptyset \\ \min\limits_{t_{js} \in suc(t_{jk})} alap_{js} - ptime_{jk} - \Delta_{jk}, & suc(t_{jk}) \neq \emptyset \end{cases} \tag{16}$$

It can be concluded that the left boundary lr_{jk} and right boundary rr_{jl} of the adjustable interval of subtask t_{jk} are:

$$\begin{cases} lr_{jk} = \Delta{-}alap_{jk} \\ rr_{jk} = \Delta{-}alap_{j(k+1)} - ptime_{jk} \end{cases} \tag{17}$$

The pseudocode of the online task scheduling algorithm is shown in Algorithm 1.

Algorithm 1: Online task scheduling analysis

```
The online task $T_j$ arrives
for $k = |t_j|$ to 1 do
   if $suc(t_{jk})$ is $\emptyset$ then
      $bl_{jk} \leftarrow ptime_{jk}$
   else:
      for $t_{jp} \in suc(t_{jk})$
          $bl_{jk} \leftarrow max\, bl_{jp} + ptime_{jp}$
      end for
    end if
 end for
 $MNCP_j \leftarrow bl_{j1}$
Calculate the $alap$ value of all subtasks according to formula (10)
for $k$ in 1 $to$ $|T_j|$ do
   $\Delta_{jk} \leftarrow \Delta_j \times ptime_{jk}/MCCP_j$
        Calculate $\Delta$-$alap_{jk}$ according to formula (16)
   if $k \neq 1$ then
        $rr_{j(k-1)} \leftarrow \Delta\text{-}alap_{jk} - ptime_{j(k-1)}$
   $lr_{jk} \leftarrow \Delta\text{-}alap_{jk}$
   end if
end for
return $< lr_{jk}, rr_{jk} >$
```

The pseudo-code of the hierarchical task execution scheduling algorithm is shown in Algorithm 2.

Algorithm 2: Multi-chain subtask scheduling

```
$t_{jk} \leftarrow ready\ task$
$EC_{jk} \leftarrow +\infty, startTime_{jk} \leftarrow +\infty, machine \leftarrow -1$
Sort the executive product agents in the industry chain $i$ according to the earliest available time, and get the sequence $set$
for $machine_l$ in $set$ do
    $sr \leftarrow avlTime \cap [lr_{jk}, rr_{jk}]$   //$sr$ indicates schedulable time range
    for $time$ in $sr$ do
        Calculate the cost according to Equation (4)
        if $cost < EC_{jk}$ then
            $EC_{jk} \leftarrow cost;$
            $startTime_{jk} \leftarrow time$
            $machine \leftarrow l$
        end if
    end for
end for
Update corporate availability time
return $< startTime_{jk}, machine, EC_{jk} >$
```

5 Experiments

5.1 Experimental Settings

The experiments were conducted on a single PC equipped with a 3.6 GHz CPU processor, 8 GB of RAM, and Windows 10 operating system. The test code was implemented using Java11, and the data was obtained from the PSPLIB (PROJECT SCHEDULING PROBLEM LIBRARY) dataset [24], which can generate task structures with different dependencies.

During the experiment, task dependencies were generated using the complexity parameter of its industrial chain [24]. As stipulated in this paper, the average online task is comprised of 8 sub-tasks with dependencies in multiple networks industrial chains.

5.2 Benchmark Algorithms

To confirm the efficacy of the proposed algorithm, it is necessary to compare and analyse its experimental results with those of a suitable comparison algorithm. This paper employs the following comparison algorithm:

Greedy algorithm based on the latest start time (ALAP) [23], which is introduced in Sect. 4.1. Subtasks are sorted based on their latest start time, and the subtask with the smallest latest start time is scheduled to execute first;

Improved fast Non-dominated Sorting Genetic Algorithm-II (NSGAII): NSGAII algorithm is widely used in multi-objective optimization solutions. In this paper, referring to previous cloud computing deployment, the algorithm is improved and applied to multiple networks industrial chains scenarios, the solution set size is set to 50, and the number of iterations is set to 200 [25];

Heuristic rule prioritization algorithm: Referring to different heuristic rules defined in previous works of literature, the scheduling priority of tasks is calculated according to the defined rules, and the scheduling requirements of high-priority tasks are met first. In this paper, the Arrival Time First (AT) strategy, the Emergency Task First (ET) Algorithm, and the Deadline Greedy Algorithm (DGA) [26] are adopted.

5.3 Optimization Objective

The experimental performance of the algorithm is evaluated from the following four aspects:

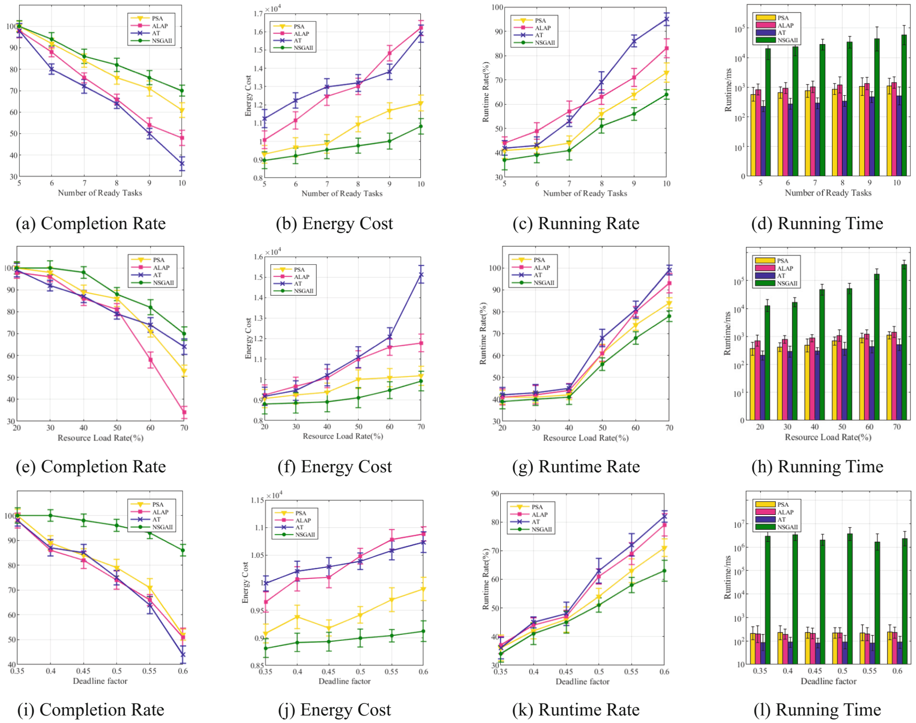
Runtime: The time it takes for the algorithm to run; Task Completion Rate: represents the proportion of successful online task execution times to the total number of tasks, which is used to measure the usability of the algorithm; Task execution Energy Cost: the energy cost required for online task execution. Different power costs can be obtained when tasks are executed in different periods, which is used to measure the effectiveness of the algorithm in executing task scheduling optimization; Runtime Rate: the ratio of the time required by an online task to the difference between its deadline and arrival time. A lower ratio means that the task has a shorter waiting time and can be executed quickly after its arrival.

5.4 Experimental Results of Passive Scheduling

Different deployment outcomes will arise for online tasks under different deadline coefficients. This section will provide experimental verification of the specific impact relationship between the ready task pool, resource load, deadline coefficient, and respective task experimentally verified.

Parameter Settings: In the experiment settings of Fig. 2 (a)–(d), the initial load of the system (the proportion of non-idle product agents in the total number) was set to 40%, and the average deadline ratio of the tasks (the minimum time required to complete the task accounted for the total time ratio) was 0.4. In the experiment settings of Fig. 2 (d)–(h), the system's average number of tasks in the ready queue was set to 4, and the average deadline ratio of the tasks was 0.4. In the experiment settings of Fig. 2 (i)–(l), the initial load of the system was set to 60%, and the average number of tasks in the ready queue was 6.

Phenomena and Reasons: As the number of executable tasks increases, the waiting time of the task queue also increases, while the scheduling time of online tasks shortens. The success rate of the task execution time selection algorithm decreases, resulting in an increase in task running rate. PSA algorithm can allocate deadlines according to task paths, so its success rate, operational rate and cost are closest to offline optimization, which has good effects in this scenario. With the increase of system resource load, the execution time of the algorithm is affected and ultimately leads to the increase of the running rate. PSA algorithm calculates the task execution time based on the available time of the product agents. When the resource load increases, the complexity of available time remains unchanged and has no direct relationship with the resource load. PSA algorithm



(a) Completion Rate (b) Energy Cost (c) Running Rate (d) Running Time

(e) Completion Rate (f) Energy Cost (g) Runtime Rate (h) Running Time

(i) Completion Rate (j) Energy Cost (k) Runtime Rate (l) Running Time

Fig. 2. Online scheduling framework.

optimizes energy cost based on task adjustable time. However, with the increase of adjustably schedulable tasks, the adjustably schedulable time of online tasks gradually decreases, which may reduce the possibility of executing tasks at the lowest electricity prices, ultimately leading to an increase in cost. Reducing the task adjustable time may also reduce the success rate of the task execution time selection algorithm, leading to further delays in task execution and increase in task running rate. In summary, PSA algorithm has good effects in this scenario and has advantages such as success rate, operational rate, and closest operational cost to offline optimization.

6 Conclusions

This paper explores strategies for optimizing tasks that dynamically arrive and are not available in advance in fine time granularity scenarios, within multiple networks chain task processing systems. We propose a general framework for online dynamic assignment, based on which we propose a passive assignment algorithm. This algorithm utilizes the critical path and offset errors of tasks and reduces the impact of dynamic allocation on production scheduling tasks. The algorithm is designed to balance allocation effectiveness and efficiency and is particularly suitable for scenarios involving fine-grained time tasks arriving online. Through simulation experiments and theoretical analysis, we demonstrate that our proposed algorithm better optimizes task completion time and execution cost compared to other comparative algorithms.

References

1. Cai, Z., Li, X., Ruiz, R., et al.: A delay-based dynamic scheduling algorithm for bag-of-task workflows with stochastic task execution times in clouds. Futur. Gener. Comput. Syst. **71**, 57–72 (2017)
2. Sha, L., Abdelzaher, T., Årzén, K.E., et al.: Real time scheduling theory: a historical perspective. Real-Time Syst. **28**, 101–155 (2004)
3. Zhou, B., Li, W., Chan, K.W., et al.: Smart home energy management systems: concept, configurations, and scheduling strategies. Renew. Sustain. Energy Rev. **61**, 30–40 (2016)
4. Rocha, H.R.O., Honorato, I.H., Fiorotti, R., et al.: An artificial intelligence based scheduling algorithm for demand-side energy management in smart homes. Appl. Energy **282**, 116145 (2021)
5. Aman, S., Simmhan, Y., Prasanna, V.K.: Energy management systems: state of the art and emerging trends. IEEE Commun. Mag. **51**(1), 114–119 (2013)
6. Zhou, B., Zou, J., Chung, C.Y., et al.: Multi-microgrid energy management systems: architecture, communication, and scheduling strategies. J. Mod. Power Syst. Clean Energy **9**(3), 463–476 (2021)
7. Schulze, M., Nehler, H., Ottosson, M., et al.: Energy management in industry–a systematic review of previous findings and an integrative conceptual framework. J. Clean. Prod. **112**, 3692–3708 (2016)
8. Ullah, I., Hussain, I., Singh, M.: Exploiting grasshopper and cuckoo search bioinspired optimization algorithms for industrial energy management system: smart industries. Electronics **9**(1), 105 (2020)
9. Jiang, Y., Zhou, Y., Li, Y.: Network layer-oriented task allocation for multiagent systems in undependable multiplex networks. In: 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, pp. 640–647, November 2013
10. Jiang, Y., Zhou, Y., Li, Y.: Reliable task allocation with load balancing in multiplex networks. ACM Trans. Auton. Adapt. Syst. (TAAS) **10**(1), 1–32 (2015)
11. Zhao, Z., Zhou, M., Liu, S.: Iterated greedy algorithms for flow-shop scheduling problems: a tutorial. IEEE Trans. Autom. Sci. Eng. (2021)
12. Li, Z., Yan, F., Jiang, Y.: Cross-layers cascade in multiplex networks. Auton. Agent. Multi-Agent Syst. **29**, 1186–1215 (2015)
13. Li, K., Wu, S., Wen, Y., et al.: Task allocation of multiagent groups in social networked systems. IEEE Internet Things J. **9**(14), 12194–12208 (2021)
14. Graves, S.C.: A review of production scheduling. Oper. Res. **29**(4), 646–675 (1981)
15. Shao, W., Shao, Z., Pi, D., et al.: Modeling and multi-neighborhood iterated greedy algorithm for distributed hybrid flow shop scheduling problem. Knowl.-Based Syst. **194**, 105527 (2020)
16. Hartmann, S., Briskorn, D.: An updated survey of variants and extensions of the resource-constrained project scheduling problem. Eur. J. Oper. Res. **297**(1), 1–14 (2022)
17. Pellerin, R., Perrier, N., Berthaut, F.: A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. Eur. J. Oper. Res. **280**(2), 395–416 (2020)
18. Tirkolaee, E.B., Goli, A., Hematian, M., et al.: Multi-objective multi-mode resource constrained project scheduling problem using Pareto-based algorithms. Computing **101**, 547–570 (2019)
19. Jayadivya, S.K., Bhanu, S.M.S.: QoS based scheduling of workflows in cloud computing. In: ICCC-2012, vol. 1, p. 47 (2012)

20. Belgacem, A., BeghdadBey, K.: Multi-objective workflow scheduling in cloud computing: trade-off between makespan and cost. Clust. Comput. **25**(1), 579–595 (2022)
21. Xin, C., Addy, M.M., et al.: Waste-to-biofuel integrated system and its comprehensive techno-economic assessment in wastewater treatment plants. Biores. Technol. **250**, 523–531 (2018)
22. Chen, H., Zhu, X., Liu, G., et al.: Uncertainty-aware online scheduling for real-time workflows in cloud service environment. IEEE Trans. Serv. Comput. **14**(4), 1167–1178 (2018)
23. Wang, Z., Lu, Z., Pan, J., et al.: Workflow scheduling strategy for deadline constrained and cost optimization in cloud. Comput. Sci. **49**(11A), 210800154–210800156 (2022)
24. Kolisch, R., Sprecher, A.: PSPLIB-a project scheduling problem library: OR software-ORSEP operations research software exchange program. Eur. J. Oper. Res. **96**(1), 205–216 (1997)
25. Li, H., Wang, B., Yuan, Y., et al.: Scoring and dynamic hierarchy-based NSGA-II for multi-objective workflow scheduling in the cloud. IEEE Trans. Autom. Sci. Eng. **19**(2), 982–993 (2021)
26. Branke, J., Nguyen, S., Pickardt, C.W., et al.: Automated design of production scheduling heuristics: a review. IEEE Trans. Evol. Comput. **20**(1), 110–124 (2015)

ε-Maximum Critic Deep Deterministic Policy Gradient for Multi-agent Reinforcement Learning

Yuanshuang Jiang[1], Kai Di[1], Zhongjian Hu[1], Fulin Chen[2], Pan Li[2],
and Yichuan Jiang[1](✉)

[1] School of Computer Science and Engineering, Southeast University, Nanjing 211189, Jiangsu, China
yjiang@seu.edu.cn

[2] School of Cyber Science and Engineering, Southeast University, Nanjing 211189, Jiangsu, China

Abstract. In Multi-Agent Reinforcement Learning, the agents are vulnerable to the other agents and the training environment, which can lead to agents' policy achieving a local optima easily and poor convergence efficiency. To tackle the above challenges, we propose a novel algorithm, *g*-Maximum Critic Multi-Agent Deep Deterministic Policy Gradient algorithm (*g*-M2DDPG), which leverages a new critic technique called *g*-Maximum Critic to balance the exploitation and exploration in updating Q-value function. We empirically evaluate our algorithms in three kinds of mixed cooperative and communication environments. These experimental results demonstrate that our algorithms significantly accelerates the learning process and outperform existing baseline algorithm MADDPG.

Keywords: Multi-agent · Cooperation · Reinforcement Learning

1 Introduction

With the help of deep learning, deep reinforcement learning (DRL) has emerged as a powerful approach for sequential decision-making problems, achieving a number of impressive results in many real-world applications such as playing go chess [1], Atari games [2], manipulating robots [3–5] and so on. Various successful algorithms have been proposed, like deep Q-network (DQN) [2], asynchronous advantage actor-critic (A3C) [6], Trust-Region Policy Optimization (TRPO) [3], Proximal Policy Optimization (PPO) [7] and deep deterministic policy gradient (DDPG) [4].

Recent advances in single agent scenarios of DRL have made it possible to transfer to multi-agent RL (MARL) problems [8], which has been used to model a lot of important

This work was supported by the Key Research and Development Program of Jiangsu Province of China (No. BE2022157), the National Natural Science Foundation of China (Nos. 62076060, 61932007), and the Defense Industrial Technology Development Program (JCKY2021214B002).

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 180–189, 2024.
https://doi.org/10.1007/978-981-99-8211-0_17

situations involving more than one single agent, for example, multi-scenario recommendation [9, 10], multiplayer games [11–13], traffic control [14, 15], autonomous driving [16] and the analysis of social dilemmas [17]. However, different from the single agent RL problems, MARL addresses the decision-making problem of multiple agents that operate in a shared environment, and each agent aims to optimize its own long-term return by interacting with the environment and other agents [18, 19]. The main challenge in MARL is training instability [19–21], if each agent just improve its policy according to environment feedback but not the other agents' actions, the changes caused by the other agents in the whole system are inexplicable. For example, policy gradient algorithms would suffer from exponentially large variance as the number of the agents increases. To tackle these challenges, many algorithms have been developed under different settings. Matignon, Laurent, and Le FortPiat [22] proposed the Hyper-Q Learning algorithm for multi-agent systems by inputting other agent's policy parameters to the Q function. Foerster et al. [23] leveraged a centralised critic to estimate the Q-function and designed Counterfactual Multi-Agent (COMA) Policy Gradients algorithm to solve multi-agent credit assignment in cooperative settings. Jiang and Lu [24] proposed an attentional communication model called ATOC inspired by recurrent models of visual attention for large-scale multi-agent environments.

Recently, Lowe et al. [20] proposed a Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm based on the actor-critic learning framework by utilizing a centralized critic for mixed cooperative-competitive environments. Li et al. [21] extended the algorithm by the minimax idea in game theory and proposed MiniMax Multi-agent Deep Deterministic Policy Gradient (M3DDPG) for competitive environments. In spite of the introduction of the centralized critic strategy, the learned policies can still get stuck in a poor local optima easily and converge in a slow rate. In fact, their critic value is updated based on the global view information, which consists of the observation and actions of all agents. However, this might not always be the best choice to obtain a good policy. For example, in an multiple landmarks environment, the speaker sends out a message, and the listener understands the speaker's information and then moves to the landmark. The reward in this setting is defined by the distance between the landmark and the listener. Sometimes the multiple landmarks are gathered together, and even become a landmark due to clustering. When we just consider to make use of the global information, the listener would spend a lot of time to understand this information. As a result, the agent's policy will achieve a local optima at the end of the episode. Furthermore, such an problem also can lead to the poor convergence rate.

To deal with the above difficulties of MADDPG algorithm, in this paper, we focus on the MARL cooperative environment setting, in which all agents collaborate with each other to achieve some shared goal. We propose a novel algorithm *ϵ-**M**aximum Critic **M**ulti-**A**gent **D**eep **D**eterministic **P**olicy **G**radient algorithm* (called ϵ-M2DDPG for short). To the best of our knowledge, this is the first work that introducing this novel ϵ-**M**aximum Critic strategy to make a trade-off between exploitation and exploration, and then such a technique can improve the convergence efficiency and overcome the local optima dilemma. Specifically, our major contributions can be summarized in the following three aspects:

- **ϵ-Maximum Critic:** To address the drawbacks of MADDPG in updating Q value given all the other agents' information, which can cause that the agent's policy gets stuck into the local optima, we introduce an ϵ-maximum critic technique during the updating Q-value process to balance the exploration and exploitation and accelerate the learning process.
- **Algorithms:** Based on thisϵ-maximum critic technique, we propose the ϵ- M2DDPG algorithm, which is an extension of MADDPG, and ϵ-M2DDPG becomes MADDPG algorithm when $\epsilon = 0$.
- **Effectiveness:** We empirically evaluate our proposed algorithm in three mixed cooperative and communicate multi-agent environments and these experimental results prove the effectiveness of our methods. The agents utilized our method significantly outperforms existing baselines.

2 Analysis of MADDPG Algorithms

Multi-agent Deep Deterministic Policy Gradient (MADDPG) [20] utilizes a centralized critic to overcome the non-stationary training problem of MARL.

Specifically, consider a game with n agents with policies that are parameterized by $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_n]$, and let $\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_n\right]$ be joint-policy of the agents. Then the gradient of the expected reward of agent i with policy $\boldsymbol{\mu}_\mathrm{i}$ is

$$\begin{aligned} &\nabla_{\theta_i} J(\theta_i) = \\ &E_{\mathbf{x},a\sim\mathcal{D}}\Big[\nabla_{\theta_i}\boldsymbol{\mu}_i(o_i)\nabla_{a_i}Q_i^{\mu}(\mathrm{x}, a_1, \ldots, a_n)\big|_{a_i=\boldsymbol{\mu}_i(o_i)}\Big], \end{aligned} \tag{1}$$

where $Q_i^{\mu}(\mathbf{x}, a_1, \ldots, a_n)$ is a centralized action-value function which takes as input the actions (*i.e.* $a = (a_1,\ldots, a_n)$) and the observations of all agents (*i.e.* $\mathbf{x} = \{o_1, , o_n\}$). Let $\mathbf{x}^{'}$ denote the next state from $\mathbf{x}$ after taking actions $a_1,\ldots, a_\mathrm{n}$, the experience replay buffer consists of the tuples ($\mathbf{x}$, $\mathbf{x}^{'}$, $a_1,\ldots, a_n$, $r_1,\ldots, r_n$). Then Q_i^{μ} is updated according to the gradient of

$$\mathcal{L}(\theta_i) = E_{\mathrm{x},a,r,\mathbf{x}'}\Big[\big(Q_i^{\boldsymbol{\mu}}(\mathbf{x}, a_1, \ldots, a_n) - y\big)^2\Big] \tag{2}$$

where $y = r_i + \gamma Q_i^{\boldsymbol{\mu}'}\left(\mathbf{x}', a_1', \ldots, a_n'\right)\Big|_{a_i'=\boldsymbol{\mu}_i'(o_i)}$ and $\boldsymbol{\mu}' = \left[\boldsymbol{\mu}_{\theta_1'}, \ldots, \boldsymbol{\mu}_{\theta_N'}\right]$ is the set of target policies with delayed parameters $\theta_i^{'}$. Note that the centralized Q function is only used during training. During decentralized execution, each policy $\boldsymbol{\mu}_{\theta}i$ only takes local information o_i to produce an action.

In the MADDPG algorithm, the critic value is updated based on the global information, which consists of the observation and actions of all agents. However, this might not always be the best choice to obtain a good policy. For example, in a multiple landmarks environment, the speaker sends out a message, and the listener understands the speaker's information and then moves to the landmark.

The reward in this scenario is defined by the distance between the landmark and the listener. Sometimes the multiple landmarks are gathered together, and even become one a landmark due to clustering, when we just consider to make use of the global information, the listener would spend a lot of time to understand this information. As a result, the agent's policy will achieve a local optima at the end of the episode. Furthermore, such an problem also can lead to the poor convergence rate. In the remainder of this paper, we present a novel algorithm ϵ-M2DDPG to deal with this challenge.

3 Algorithms

In this section, we propose the ϵ-Maximum Critic Multi-Agent Deep Deterministic Policy Gradient algorithm (ϵ-M2DDPG) to overcome the drawbacks of MAD-DPG mentioned in Sect. 2.

In the training process, the goal of each agent i is to maximize its accumulative return $J(\theta_i) = E_{s\sim p^{\mu}}[R_i]$. Then we have

$$\max J(\theta_i) = \max E_{s\sim p^{\mu}}[R_i] \tag{3}$$

$$= \max E_{s\sim p^{\mu}}\left[\sum_{\mathrm{t=0}}^{\mathrm{T}} \gamma^t r_i\left(s^t, a_1^t, \ldots, a_n^t\right)\Big|_{a_i^t=\boldsymbol{\mu}_i(o_i^t)}\right] \tag{4}$$

$$= E_{s^0\sim p^{\mu}}\left[\max Q_i^{\mu}\left(s^0, a_1^0, \ldots, a_n^0\right)\Big|_{a_i^0=\boldsymbol{\mu}_i^0(o_i^0)}\right] \tag{5}$$

In Eq. 4, state s^t at time t depends on this distribution $p^{\boldsymbol{\mu}}$ and the action $\boldsymbol{\mu}_i\left(o_i^t\right)$. To get a bigger expectation of reward, we use the current information to maximize $E_{s\sim p^{\mu}}[R_i]$. In Eq. 5, we derive the modified Q function $\max Q_i^{\mu}(s, a_1, \ldots, a_n)$.

Here we consider updating the centralized action-value function Q_i^{μ}. It is naturally centralized and can be rewritten in this form:

$$\begin{aligned} Q_i^{\mu}(s, a_1, \ldots, a_n) &= r_i(s, a_1, \ldots, a_n) + \\ &\gamma E_{s'}\left[Q_{\mathrm{i}}^{\mu}\left(s', a_1', \ldots, a_n'\right)\big|_{\mathrm{a}_i'=\boldsymbol{\mu}_i(o_i')}\right] \end{aligned} \tag{6}$$

Critically, $Q_i^{\mu}\left(s', a_1', \ldots, a_n'\right)$ conditions on the current state s as well as the current actions a_1,..., a_{n}, and represents the current reward plus the discounted future return starting from the next state, s'. We can naturally apply off-policy Temporal Difference learning to update Q_i^{μ}.

So we can directly leverage the deterministic policy gradient theorem to compute $\nabla_{\theta_i} J(\theta_i)$ and use off-policy Temporal Difference method to update the Q function. Thanks to the centralized Q function in MADDPG (i.e., Eq. 1), which takes in the actions from all the agents, our derivation naturally can be suited and is perfectly aligned with the MADDPG formulation (as shown in Eq. 1):

$$\begin{aligned} &\nabla_{\theta_i} J(\theta_i) = \\ &E_{\mathbf{x}\sim\mathcal{D}}\left[\nabla_{\theta_i}\boldsymbol{\mu}_i(o_i)\nabla_{a_i}\max Q_i^{\boldsymbol{\mu}}(\mathbf{x}, a_1, \ldots, a_n)\big|_{a_i=\boldsymbol{\mu}_i(o_i)}\right] \end{aligned} \tag{7}$$

Correspondingly, we can obtain the new Q function update rule by combining Eq. 5, Eq. 6 and Eq. 2:

$$\mathcal{L}(\theta_i) = E_{\mathbf{x},a,r,\mathbf{x}'\sim\mathcal{D}}\Big[\big(\max Q_i^{\mu}(\mathbf{x}, a_1, \ldots, a_n) - y\big)^2\Big] \tag{8}$$

where $y = r_i + \gamma Q_i^{\boldsymbol{\mu}'}\big(\mathbf{x}', a_1', \ldots, a_n'\big)$, $\boldsymbol{\mu}'$ denotes the target policy of agent i with delayed parameters θ_i', and $Q_i^{\boldsymbol{\mu}}$ denotes the target Q network for agent i. In this way, we increase the value of Q by seeking a maximum Q, which in turn improves the exploration.

To get this max Q_i^{μ}, we try to look at this problem from multiple views (multi-view refers to individual view information (current agent's information), global view information (all n agents' information), combined view information (the combined view also contains multiple views, which are the combination of the current agent's information and other agents' information)). So there are $\sum_{\mathrm{k}=0}^{n-1} C_{n-1}^{\mathrm{k}}$ kinds of information, we use this multi-view information as the input of the network to generate Q, and then get a maximum Q value. This can be formalized as follows:

$$\begin{aligned}\max Q_i^{\mu}(\mathbf{x}, a_1, \ldots, a_n) = \max\Big[\, & Q_i^{\boldsymbol{\mu}}(O_1, A_1)\big|_{a_i=\boldsymbol{\mu}_i(o_i)}, \\ & \cdots, Q_i^{\boldsymbol{\mu}}\Big(O_{\sum_{k=0}^{n-1} C_{n-1}^k}, A_{\sum_{k=0}^{n-1} C_{n-1}^k}\Big)\Big|_{a_i=\boldsymbol{\mu}_i(o_i)}\Big]\end{aligned} \tag{9}$$

where O_i and A_i is the i-th combination of observation and action information, in this way, the algorithm will be more exploratory and it can remove some negative information. Because in this case the maximum Q that we choose is not necessarily generated by the global view information.

However, an increase in exploration may lead to a decrease in exploitation. Therefore, how to balance the exploration and exploitation becomes a very important problem. We try to use a new perspective to look at the exploitation and exploration in reinforcement learning. We use the ϵ-greedy method to update the critic instead of the way in MADDPG. In the beginning of training, for each agent, we generate a probability. Then we divide the agents according to these probabilities ϵ_{i} generated for each agent: agents with a probability greater than ϵ always perform Q update as in MADDPG algorithm, and agents with a probability less than ϵ always perform maximum Q update, which advantageously solves the problem of robustness of the algorithm when the experimental environment changes. Our algorithm can still guarantee its exploratory and convergence properties. The experimental results also prove that when we do not use this, we cannot obtain better rewards in some environments, and the convergence speed is not fast enough. Finally, combining Eq. 7, Eq. 8 and ϵ-update method, we can get our ϵ-Maximum Critic method. This can be formalized as follows

$$\max Q_i^{\mu} = \begin{cases} \max Q_{\mathrm{i}}^{\mu} & \mathit{if}\ \varepsilon_i \leq \varepsilon \\ Q_i^{\mu} & \mathit{otherwise}. \end{cases} \tag{10}$$

Final, our proposed ϵ-Maximum Critic Multi-Agent Deep Deterministic Policy Gradient algorithm as shown in Algorithm 1.

Algorithm 1: ϵ-Maximum Critic Multi-Agent Deep Deterministic Policy Gradient (ϵ-M2DDPG)

```
for agent $i$ = 1 to $n$ do
    Initialize a random probability $\epsilon_i$
for episode = 1 to $M$ do
    Initialize a random process $\mathcal{N}$ for action exploration
    Receive initial state $\mathbf{x}$
    for $t$ = 1 to max-episode-length do
        for each agent i, select action $a_i = \boldsymbol{\mu}_{\theta_i}(o_i) + \mathcal{N}_t$ w.r.t. the current
          policy and exploration
        Execute actions $a = (a_1, ..., a_n)$ and observe reward $r$ and new state $\mathbf{x}'$;
        Store $(\mathbf{x}, a, r, \mathbf{x}')$ in replay buffer $\mathcal{D}$, and set $\mathbf{x} \leftarrow \mathbf{x}'$
        for agent $i$ = 1 to $n$ do
            Sample a random minibatch of $\mathbf{x}$ samples $(\mathbf{x}^j, a^j, r^j, \mathbf{x}'^j)$ from $\mathcal{D}$ :
            $y^j = r_i^j + \gamma Q_i^{\boldsymbol{\mu}'}(\mathbf{x}'^j, a_1', \ldots, a_n')|_{a_i'=\boldsymbol{\mu}_i'(o_i)}$ if $\epsilon_i < \epsilon$ then
                $Q_i^{\boldsymbol{\mu}}(\mathbf{x}^j, a_1^j, \ldots, a_n^j) = \max Q_i^{\boldsymbol{\mu}}(\mathbf{x}^j, a_1^j, \ldots, a_n^j)|_{a_i=\boldsymbol{\mu}_i(o_i)}$
            Update critic by minimizing the loss:
              $\mathcal{L}(\theta_i) = \frac{1}{S}\sum_j \left(Q_i^{\boldsymbol{\mu}}(\mathbf{x}^j, a_1^j, \ldots, a_n^j) - y^j\right)^2$ Update actor using the
              sampled policy gradient: $\nabla_{\theta_i} J \approx$
              $\frac{1}{S}\sum_j \nabla_{\theta_i}\boldsymbol{\mu}_i(o_i^j)\nabla_{a_i} Q_i^{\boldsymbol{\mu}}(\mathbf{x}^j, a_1^j, \ldots, a_n^j)|_{a_i=\boldsymbol{\mu}_i(o_i)}$
        Update target network parameters for each agent $i$:
          $\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i'$
```

4 Experiments

In this section, the ϵ-M2DDPG algorithm are applied in mixed cooperative and communication scenario, compared with MADDPG as a baseline. We use one scenario from MADDPG, cooperative navigation, with the same configuration as MADDPG [20]. For our proposed algorithms, ϵ is selected from 0.7, 0.8 and 0.9.

4.1 Environments

In this subsection we mainly explain the environments we used in this paper. The environment consists of N agents and L landmarks in a two-dimensional world.

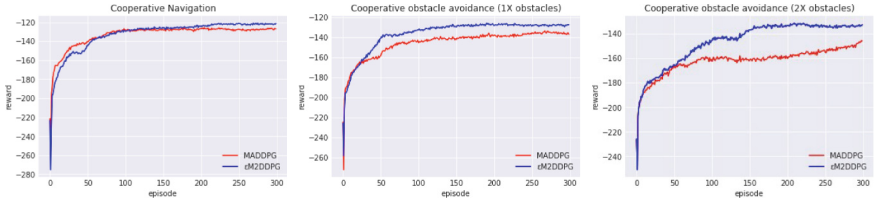
Cooperative Navigation. In this scenario, the number of agents is as same as the number of landmarks, and each agent is required to move to one landmark. The agent can observe the relative positions of other agents and the landmark, and its rewards are calculated based on the shortest distance between all agents and each landmark. Thus the agent must learn to "cover" all landmarks. Meanwhile, since agents occupy physical spaces, they will be punished if they collide with each other. So they need to not only infer which landmark one must cover but also move there while avoiding the others.

Cooperation Obstacle Avoidance. In this scenario, the cooperative navigation environment has been upgraded by adding obstacles, meanwhile agents, landmarks and obstacles are of the same size. The agent must learn to reach the target point while avoiding obstacles and other agents.

4.2 Results Analysis

We compare the MADDPG method with our algorithms, the experiment proves that our methods have good performance in the cooperative scenario. The following is the results analysis. The results are shown in Fig. 1. We note that in environment, ϵ-M2DDPG is superior to the MADDPG training agent, and the rewards obtained by ϵ-M2DDPG can reach a position that cannot be reached by the MADDPG algorithm, which implies that the strategy quality of ϵ-M2DDPG training is better than MADDPG. To better compare our method to the original method, in the cooperative navigation experiments, we use four indicators to measure whether the agent can arrive at all the landmarks by using a certain strategy. We train our models until convergence, and then it is evaluated by the following indicators: Collisions are the average agent collisions of 300,000 iterations. Average dist is the average distance between the agent and the landmark in 300,000 iterations. Full occupied is the percentage of at least one landmark taken up in 300,000 iterations. Occupied is the percentage of at least one landmark taken up in 300,000 iterations. Time is the earliest moment when the agent reaches the maximum reward of the MADDPG algorithm in 300,000 iterations. The results in Table 1 further verify the effectiveness of our proposed algorithm.

In the cooperative navigation environment, the size of the agent is three times that of the landmarks. This means that the agent can more easily reach the landmarks as it may only need to place part of the agent on the landmark to achieve the goal. Experimental simulation results show that the agent exhibits a certain level of laziness under the MADDPG algorithm, where it tends to complete the task by only partially placing the agent on the landmark instead of fully placing it. The ϵ-M2DDPG algorithm overcomes this issue. The ϵ-M2DDPG algorithm aligns the center of the agent with the center of the landmark. Therefore, in the experimental results shown in Fig. 1(a), the final reward of ϵ-M2DDPG is significantly higher than that of the MADDPG algorithm. This also indicates that the proposed ϵ-M2DDPG algorithm is more exploratory than the MADDPG approach. Moreover, according to the time metrics in Table 1, it can be observed that the ϵ-M2DDPG algorithm indeed converges faster.



(a) (b) (c)

Fig. 1. Experimental Results. The x-axis represents the number of training episodes (in thousands) and the y-axis represents the reward for averaging a single agent.

To further investigate the exploratory nature of the algorithm, this study makes the environment more complex and practical. The study also considers whether the limited improvement in experimental results is due to the large size of the agent or the simplicity

Table 1. Cooperative Navigation

	collisions	Average dist	full occupied	occupied	time
MADDPG	1.025	0664	33.6%	71.6%	205
g-M2DDPG	1.014	0.623	33.7%	74.6%	139

of the environment. Therefore, in the cooperative navigation environment, the size of the agent is reduced and obstacles of different multiples relative to the number of agents are introduced. In two different difficulty experiments of cooperative obstacle avoidance, the performance metrics of ϵ-M2DDPG are better than those of the MADDPG algorithm (as shown in Fig. 1(b),1(c) and Table 2).

Table 2. Cooperative obstacle avoidance (1X obstacles)

	collisions	Average dist	full occupied	occupied	time
MADDPG	1.059	0762	15.2%	67.4%	268
g-M2DDPG	1.058	0.643	42.8%	74.9%	95

Table 3. Cooperative obstacle avoidance (2X obstacles)

	collisions	Average dist	full occupied	occupied	time
MADDPG	1.098	0.856	4.5%	56.2%	299
g-M2DDPG	1.102	0.676	28.2%	72.2%	98

Based on the experimental results with 1X obstacles (as shown in Fig. 1(b)), it is found that the ϵ-M2DDPG algorithm is more exploratory. For example, when multiple agents pass between two obstacles, from a global perspective, the agents may not directly pass through the gap between the obstacles (which reduces the chance of collision). Instead, when using the multi-view evaluation method, the agents pass directly between the two agents (using individual views without considering the information of other agents, thus passing directly). This is also the reason why the ϵ-M2DDPG approach has a higher collision rate (as shown in Table 3), and even when obstacles are added, the experimental results remain the same (as shown in Table 3). The reason for this may be that the ϵ-M2DDPG algorithm encourages exploration by adopting the multi-view approach.

5 Conclusions

In this paper, we propose a novel algorithm, ϵ-Maximum Critic Multi-Agent Deep Deterministic Policy Gradient (ϵ-M2DDPG) by introducing a ϵ-Maximum Critic technique to MADDPG, which can make a trade-off between exploitation and exploration during the updating Q-value process. Specifically, we randomly selecting several agents to update with a global view with a probability of $1-\epsilon$, instead of all agents using the maximum Q value to update. The experimental results show that our new proposed algorithms outperform MADDPG algorithm in five kinds of mixed cooperative and communication environments in terms of the final agent average reward and convergence efficiency.

For the future work, there are many important and interesting directions: (1) the computation efficiency in the calculation of ϵ-maximum critic when the number of agents grows requires further investigation; (2) theoretical analysis of ϵ-M2DDPG is also important problems in need of discussion; (3) generalizing the ϵ-Maximum Critic into the cooperative and competitive mixed environments is another intriguing problem.

References

1. Silver, D., et al.: Mastering the game of go with deep neural networks and tree search. Nature **529**(7587), 484–489 (2016)
2. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
3. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization. In: International Conference on Machine Learning, pp. 1889–1897 (2015)
4. Lillicrap, T.P., et al.: Continuous control with deep reinforcement learning. In: ICLR (Poster) (2016)
5. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. J. Mach. Learn. Res. **17**(1), 1334–1373 (2016)
6. Mnih, V., et al.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937 (2016)
7. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
8. Hu, J., Wellman, M.P., et al.: Multiagent reinforcement learning: theoretical framework and an algorithm. In: ICML, vol. 98, pp. 242–250. Citeseer (1998)
9. Zhao, X., Xia, L., Yin, D., Tang, J.: Model-based reinforcement learning for wholechain recommendations. arXiv preprint arXiv:1902.03987 (2019)
10. Gui, T., et al.: Mention recommendation in twitter with cooperative multi-agent reinforcement learning. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 535–544 (2019)
11. Peng, P., et al.: Multiagent bidirectionally-coordinated nets: emergence of human-level coordination in learning to play starcraft combat games. arXiv preprint arXiv:1703.10069 (2017)
12. Brown, N., Sandholm, T., Machine, S.: Libratus: the superhuman AI for no-limit poker. In: IJCAI, pp. 5226–5228 (2017)
13. Brown, N., Sandholm, T.: Superhuman AI for multiplayer poker. Science **365**(6456), 885–890 (2019)
14. Bazzan, A.L.: Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. Auton. Agent. Multi-Agent Syst. **18**(3), 342 (2009)
15. Ma, J., Wu, F.: Feudal multi-agent deep reinforcement learning for traffic signal control. In: Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2020, Auckland, New Zealand, 9–13 May 2020, pp. 816–824 (2020)
16. Shalev-Shwartz, S., Shammah, S., Shashua, A.: Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295 (2016)

17. Leibo, J.Z., Zambaldi, V., Lanctot, M., Marecki, J., Graepel, T.: Multi-agent reinforcement learning in sequential social dilemmas. arXiv preprint arXiv:1702.03037 (2017)
18. Busoniu, L., Babuska, R., De Schutter, B.: A comprehensive survey of multiagent reinforcement learning. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **38**(2), 156–172 (2008)
19. Zhang, K., Yang, Z., Başar, T.: Multi-agent reinforcement learning: a selective overview of theories and algorithms. arXiv preprint arXiv:1911.10635 (2019)
20. Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Abbeel, O.P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in Neural Information Processing Systems, pp. 6379–6390 (2017)
21. Li, S., Wu, Y., Cui, X., Dong, H., Fang, F., Russell, S.: Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4213–4220 (2019)
22. Matignon, L., Laurent, G.J., Le Fort-Piat, N.: Independent reinforcement learners in cooperative Markov games: a survey regarding coordination problems (2012)
23. Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., Whiteson, S.: Counterfactual multi-agent policy gradients. arXiv preprint arXiv:1705.08926 (2017)
24. Jiang, J., Lu, Z.: Learning attentional communication for multi-agent cooperation. In: Advances in Neural Information Processing Systems, pp. 7254–7264 (2018)

Effective Density-Based Concept Drift Detection for Evolving Data Streams

Zelin Cui[1], Hui Tian[2], and Hong Shen[3,4](✉)

[1] Institute of Information Security Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Information and Communication Technology, Griffith University, Brisbane, Australia
[3] Faculty of Applied Sciences, Macao Polytechnic University, Macao, China
hong.shen@adelaide.edu.au
[4] School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China

Abstract. Concept drift is a common phenomenon appearing in evolving data streams of a wide range of applications including credit card fraud protection, weather forecast, network monitoring, etc. For online data streams it is difficult to determine a proper size of the sliding window for detection of concept drift, making the existing dataset-distance based algorithms not effective in application. In this paper, we propose a novel framework of Density-based Concept Drift Detection (DCDD) for detecting concept drifts in data streams using density-based clustering on a variable-size sliding window through dynamically adjusting the size of the sliding window. Our DCDD uses XGBoost (eXtreme Gradient Boosting) to predict the amount of data in the same concept and adjusts the size of the sliding window dynamically based on the collected information about concept drifting. To detect concept drift between two datasets, DCDD calculates the distance between the datasets using a new detection formula that considers the attribute of time as the weight for old data and calculates the distance between the data in the current sliding window and all data in the current concept rather than between two adjacent windows as used in the exiting work DCDA [2]. This yields an observable improvement on the detection accuracy and a significant improvement on the detection efficiency. Experimental results have shown that our framework detects the concept drift more accurately and efficiently than the existing work.

Keywords: Data Mining · Machine Learning · Data-Stream Clustering · Concept-Drift Detection

1 Introduction

Data stream clustering has been successfully applied for detection of concept drift [11] which is an important problem arising in a wide range of applications including credit card fraud protection, the weather forecast, network intrusion detection, etc.. The concept of interest may depend on some hidden context, not given explicitly in the form of predictive features [3]. In other words, the concepts drift with time from what we analyze from current data. For example, the buying preferences of customers may change with

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 190–201, 2024.
https://doi.org/10.1007/978-981-99-8211-0_18

time, depending on the day of the week, availability of alternatives, discount rate, etc. [3]. If we do not detect the concept drift in time, we may end up with taking the wrong concept, which not only decreases the quality of clusters but also can lead to unexpected clustering results. Hence, dealing with concept drift is crucial in many applications.

Typically, concept drift detection is done by calculating the distance of two datasets between adjacent sliding windows of fixed size using the rough-set theory. The detection effectiveness depends heavily on the size of sliding windows. Both too small and too large windows are undesirable, because the former may be unable to capture a single concept and the latter may contain multiple concepts. However, because of the fast evolution property of online data streams, it is difficult to determine a proper size of the sliding window for effective detection of concept drift. This makes the existing algorithms based on this approach difficult to be used in real application.

In this paper, we present a new framework for concept drift detection, named density-based concept drift detection (DCDD). It is based on density-based clustering [15] with a variable-size sliding window which is formed by dynamically adjusting the size of the sliding window based on the prediction model trained by XGBoost(eXtreme Gradient Boosting) [4] to adapt to the changes of the data stream. We extend the formula used in the existing drift detection algorithm DCDA [2] by incorporating the time attribute in calculating the distance of two datasets to find the concept drift.

2 Related Work

Concept drift, which was first introduced by Schlimmer and Granger in [11], refers to the phenomenon that data points are subject to different distribution models in different time periods. There is a rich literature on concept drift detection in which many algorithms are based on classification relying on error rate of classification prediction, such as [17]. While the classification-based algorithms are simple and efficient, they need data with class labels as training data set which is difficult to obtain for time-evolving data streams.

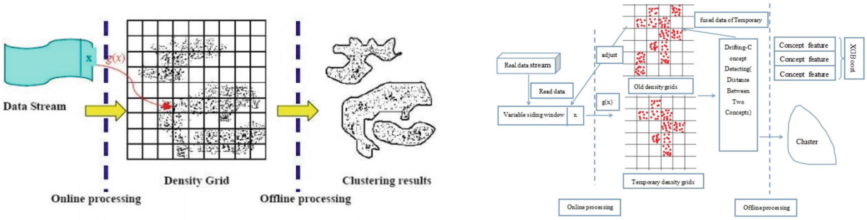
To address this issue, concept drift detection based on clustering was proposed. Stream-detect [10], detects concept drift through analyzing the clustering results to identify changes in data streams by measuring deviation of clustering result online. Chen et al. [3] proposed a framework to perform clustering on the categorical time-evolving data by comparing the distribution of the clusters and the outliers from the last and the current sliding windows. Because the window size is fixed, it does not adapt to the change of data streams. [2], proposed a concept drift detection algorithm (DCDA) based on the rough-set theory [13] and sliding window technique to improve the efficiency, which calculates the distance between the last and the current windows to detect concept drift before starting the clustering process.

To find arbitrarily shaped clusters and handle noises efficiently, numerous density-based clustering algorithms have been developed, such as D-Stream I [5], DD-Stream [12], D-Stream II [18], GDC-Stream [7] and PKS-Stream [14], and Relative Density-Based [9], These algorithms process the raw data only once and do not need to set the number of clusters. Recently, concept drift detection has been applied to high-speed streams [16] and also finds application for multi-label classification of IoT data steams [19]. They suffer from the difficulty of effectively adapting to dynamic changes of online data streams.

3 Preliminaries

3.1 Density-Based Clustering

From the Fig. 1(a), we can see the framework of density-based clustering. It uses a two-phase scheme [1], which consists of an online component and an offline component. In the online component, density-based clustering maps each input data record into a corresponding grid and updates the density of grid which is the sum of all data points in the gird. In the offline component, it uses an incomplete partitioning strategy to cluster the density grids. We take advantage of this process and design our framework for concept drift detection that is shown in Fig. 1(b).



(a) Density-based clustering [5]

(b) Density-based concept drift detection

Fig. 1. Density-based Clustering and Concept Drift Detection

3.2 Definitions

In this section, we introduce the relevant concepts used in our framework. We assume that the input data stream has d dimensions and define the data space $S = S_1 \times S_2 \times \cdots \times S_d$, where S_{i} is the definition space for the i^{th} dimension.

Definition 3.2.1 (Grid Cell). For data space $S = S_1 \times S_2 \times \cdots \times S_d$, each $S_i (1 \leq i \leq d)$ is divided into p_i parts evenly, we define the intersection of $S_i (1 \leq i \leq d)$ as the grid cell g. That is, $g_{j_1 j_2 .. j_d} = S_{1,j_1} \cap S_{2,j_2} \cap \ldots \bigcap S_{d,j_d}$, $1 \leq j_t \leq p_t$, $1 \leq t \leq d$.

When a data record $X = (x_1, x_2, \ldots, x_d)$ arrives, it can be mapped to a density grid $g(x)$ as follows: $g(x) = (j_1, j_2, \cdots, j_d)$, where $x_{\mathrm{i}} \in S_{ij_i}$, $1 \leq i \leq d$, $1 \leq j_t \leq pt$, $1 \leq t \leq d$;

The grid density of a grid cell is the sum of all the data points in the grid cell. That is, the density of grid cell g at t is:

$$D(g, t) = \sum_{x \in g} D(x, t)$$

At time t, the average density of the non-empty grid cells is $Den_{\text{avg}} = \frac{\sum_{i=1}^{K} D(g,t)}{K}$, where the $D(g, t)$ is the grid density of the non-empty grid cell g and K is the number of non-empty grid cells.

Definition 3.2.2 (Dense Grid and Sparse Grid). Grid cell *g* is a dense grid if $D(g, t) \geq \alpha Den_{avg}$, and a sparse grid if $D(g, t) < \alpha Den_{avg}$, where α is a parameter controlling the threshold.

In paper [5], the boundary between dense grid and sparse grid is defined as a fixed value that is not flexible and hard to set. In comparison, our above definition on the boundary can effectively adapt to the unknown data stream.

Definition 3.2.3 (Grid Characteristic Vector). The characteristic vector of grid cell *g* is defined as (*D, label, status, t*), where *D* is the last updated density of *g*, *label* is the class of *g*, *t* is the time that the last data came in, and *status* (either SPORADIC or DENSE) is used to mark the status of *g*.

In our framework, in order to get the distance between two data sets, we use two density grids: temporary density grids and old density girds. The temporary density grids store the grid characteristic vectors of the data in the current sliding window. The old density girds store the grid characteristic vectors of all data in the same concept.

In order to predict the next concept and the amount of data stream in the next concept, we need to collect and store the feature vector of the concept that we call concept-feature when the concept drift is detected. We use the XGBoost (eXtreme Gradient Boosting) [4] to train the concept-features and the trained model to predict. How to extract the attributes of the concept-feature will be explained in Sect. 4.4.

3.3 Concept Drifting Detection

Concept drift detection algorithm for data streams (DCDA) was proposed in [2] that works by calculating the distance between the current sliding window and the last sliding window based on the rough membership function and the sliding-window technique [1, 6, 8, 10].

The distance between two datasets is measured as follows:

For the current subset S^{T_i} and the last subset S^{T_j}, the distance between S^{T_i} and S^{T_j} is defined as

$$d_A\left(S^{T_i}, S^{T_j}\right) = \frac{1}{|A|}\sum_{a\in A} d_{\{a\}}\left(S^{T_i}, S^{T_j}\right) = \frac{\sum_{a\in A}\sum_{x\in S^{[T_i,T_J]}}\left|\mu^{\{a\}}_{S^{T_i}}(x) - \mu^{\{a\}}_{S^{T_j}}(x)\right|}{\left|S^{[T_i,T_j]}\right||A|},$$

where *A* is a non-empty set of attributes, and $\mu^{\{a\}}_{S^{T_i}}(x)$ is a rough membership function.

If the distance between two datasets is larger than the threshold, the data in the current sliding window will perform re-clustering to capture the emerging new concept. In contrast, if the concept is steady, each object of the current window will be allocated into the corresponding cluster according to distance comparison [2].

4 The Proposed Algorithm

4.1 Overall Framework

Our density-based concept drift detection (DCDD) framework follows the density-based clustering framework [15] composed of an online component and an offline component as illustrated in Fig. 1(b). In the online component, we use a variable sliding window to read new data records. When the variable sliding window is full, the data stream is mapped into the temporary density grids and the characteristic vector of the corresponding grid cells is updated. Then we calculate the distance between the old density grids and the temporary density grids to detected concept drift (initially the old density grids was empty). If the distance is smaller than a certain threshold, no concept drift is detected, and the temporary density grids are merged into the old density grids and cleared. Then the variable sliding window is adjusted by our strategy described in Sect. 4.3. Otherwise, if the distance is greater than the threshold, concept drift is detected, the old density grids are copied and clustered in the offline component and cleared. The temporary density grids are copied to the old density grids and cleared. In the offline component, our DCDD forms clusters based on the copy of the old density grids. Besides, it extracts the concept-feature of this concept and adds it into the concept-list. In addition, when the size of concept-list is enough large, it uses the XGBoost (eXtreme Gradient Boosting) [4] to train the dataset of concept-list and then uses the trained model to predict the message of the next concept to adjust the variable sliding window.

4.2 Time-Weighted Concept Drift Detection

For the online component, to detect concept drift in a data stream, we apply an extended concept drift detection model of DCDA [2] to calculate the distance between the old density grid and the temporary density grid as follows, observing the deficiencies of DCDA:

We assign all grids in the old density grids a weight $\frac{1}{\delta^{t_{now}-t-1}}$, where t_{now} is present time, t is in the Grid Characteristic Vector and $\delta \in (0, 1)$ is a constant called the weight factor.

For dense grids in the temporary density grids T and dense grids in the old density grids O, the distance between T and O with respect to S is defined as

$$d_A(T, O) = \frac{1}{|S|} \sum_{s \in S} d_{\{s\}}(T, O) = \frac{\sum_{s \in S} \sum_{g \in T \cup O} \left| \mu_T^{\{s\}}(g) - \frac{1}{\delta^{t_{now}-t-1}} \mu_O^{\{s\}}(g) \right|}{|T \cup O||S|}, \quad (1)$$

where $\delta \in (0, 1)$ is a constant, t_{now} is the present time, t is in the Grid Characteristic Vector of g, $\frac{1}{\delta^{t_{now}-t-1}}$ is the weight for grid in the old density grids, S is the dimension of the defined data space, $\mu_G^{\{s\}}(g)$ is a rough membership function.

4.3 Sliding Window Size Calculation

To improve efficiency, our detection scheme uses a variable-size sliding window whose size is dynamically adjusted according to the framework in Fig. 2. Firstly, we initialize a sliding window size $N = N_{init}$ and set a maximum size N_{MAX} based on the memory capacity. We detect the concept drift between the temporary density grids and the old density grids based on the detection formula of Eq. (1). If the old and temporary density grids present the same concept, we determine whether the temporary density grids have new dense grids that are not in the old density grids, and calculate the density of all these dense grids M. We then adjust the sliding window size to $N = N + M$ ($N = N_{MAX}$ if $N + M$ exceeds N_{MAX}); If the old and temporary density grids present concept drift, we revert the size of sliding window to the initial value in the next step ($N = N_{init}$). If our prediction model that is described in Sect. 4.4 has created, we set N based on the predicted value. Using the above strategy, our algorithm is described in algorithm 1.

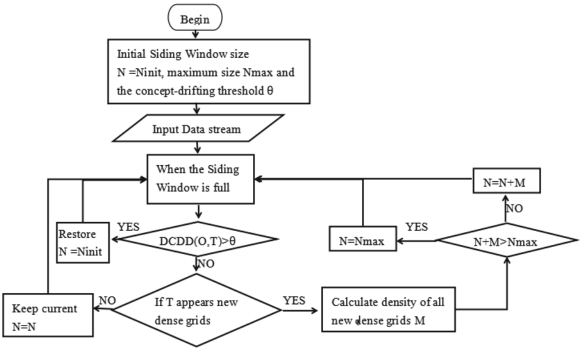


Fig. 2. Dynamic Adjustment of Sliding Window Size

4.4 Prediction on Concept-Feature Classification

When we collect certain amount of concept-feature, we use XGBoost [4] to train concept-feature to obtain a prediction model. Before running XGBoost, three parameters are set for xGboost: general parameters, booster parameters and task parameters. General parameters control the booster which are either tree model (tree) or linear model (linear) commonly.

We set 5 types of attributes for concept-feature, our prediction model with 5 trees is defined below:

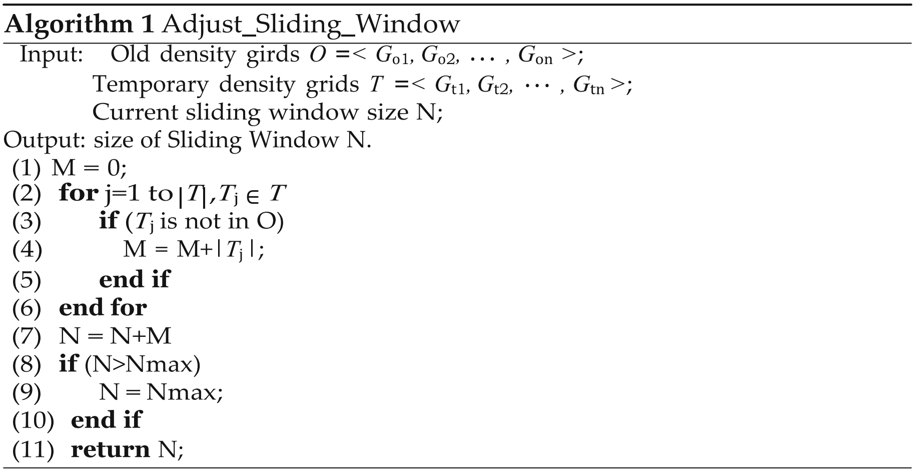
$$model\ : \hat{y}_i = \sum_{k=1}^{5} f_k(x_i), f_k \in F$$

The objective is defined as:

$$Obj = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{5} \Omega(f_k),$$

where $\sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right)$ is the training loss.

This prediction model is based on concept-feature of the current concept drift to predict the amount of data stream in the next concept. When the current concept is over, we get the concept-feature and add concept-feature into concept-list. When the size of concept-list reaches β, the concept-list is trained by XGBoost. Using the model, we obtain the predicted value PN. So at the fastest detection speed, PN is divided into two parts, which effectively sets the size of variable sliding window to $\frac{PN}{2}$.



Algorithm 1 Adjust_Sliding_Window

```
Input:  Old density girds $O$ =< $G_{o1}$, $G_{o2}$, ... , $G_{on}$ >;
        Temporary density grids $T$ =< $G_{t1}$, $G_{t2}$, ··· , $G_{tn}$ >;
        Current sliding window size N;
Output: size of Sliding Window N.
M = 0;
for j=1 to |$T$|,$T_j \in T$
    if ($T_j$ is not in O)
        M = M+| $T_j$ |;
    end if
end for
N = N+M
if (N>Nmax)
    N = Nmax;
end if
return N;
```

4.5 Algorithm Description

The whole algorithm of our density-based concept drift detection (DCDD) is presented in Algorithm 3. The algorithm for computing grid density distance is presented in Algorithm 2. Our DCDD algorithm can be simply summarized below: we use the detection formula of Eq. (1) to detect concept drift. If concept drift occurs, we use the prediction model to adjust the size of the sliding window. Otherwise, we use the strategy in Sect. 4.3 to adjust the size of the sliding window.

The time complexity of our detection algorithm is $O(|T \cup O|\,|S|)$, where T is the number of dense grids in the temporary density grids, O is the number of dense grids in the old density grids and S is the dimensions of the defined data space. Compared with DCDA [2], the time complexity of our detection is greatly reduced. It is easy to see that the time complexity of our detection algorithm is linear with respect to the number of dense girds.

Algorithm 2 Compute_Grid_Density_Distance

```
Input:  Old density girds $O$ =< $G_{o1}, G_{o2}, \cdots, G_{on}$ >;
        Temporary density grids $T$ =< $G_{t1}, G_{t2}, \cdots, G_{tn}$ >;
Output: Distance between $T$ and $O$.
$G = O \cup T$;
distance = 0;
    for s=1 to |S| do
        $G/IND(S_s) = \{g_1, g_2, \cdots, g_m\}_p, S_s \epsilon S$;
        for j=1 to $m_p$ do
            distance = distance + $||g_j \cap O| - \frac{1}{\delta^{tnow-t_1}} \times |g_j \cap T||$;
        end for
    end for
return distance;
```

5 Experimental Results

We evaluate the precision, recall and efficiency of our DCDD and compare it with DCDA [2]. We use the synthetic data that contains 15% noisy data and a real data set KDD CUP-99 that is network intrusion detection data set and has been cited by many articles of data stream clustering. It collected 9 weeks of TCPdump (*) network connection and system audit data by the MIT Lincoln laboratory which contains the simulation of various types of users, a variety of network traffic and attack means, and it like a real network environment and the network intrusion detection data stream. It contains a total of 41 dimensional properties, of which 34 are continuous attributes. Each data stream of network connection is marked as normal or abnormal, and the abnormal type is subdivided into 4 main categories that are DOS, R2L, U2R and PROBING. We test the DCDD on KDD CUP-99 data set one hundred times persistently and test the accuracy of clustering results.

5.1 Performance Evaluation

In order to compare our proposed DCDD with the existing DCDA, we use the popular evaluation metrics of precision and recall. If a is the number of drifting concepts in the data set, b is the number of drifting concepts that we detect and c is the number of drifting concepts that are correctly detected. The precision and recall of the detection are defined as $Precision = \frac{c}{b}$ and $Recall = \frac{c}{a}$, respectively.

Firstly, we test DCDD under our detection formula Eq. (1) and DCDA with different initial sizes of the sliding window on the same synthetic dataset, and we set parameters $\theta = 0.3$, $\alpha = 0.5$ and set $\delta = 0.5$. From the result of the 3, although the recall of DCDD is slightly worse than the recall of DCDA, DCDD gives a much better precision than the DCDA (Fig. 3).

Algorithm 3 Density-based Concept Drift Detection (DCDD)

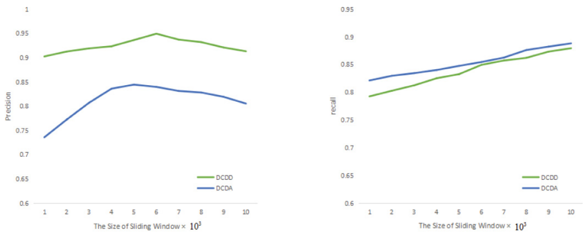
```
Input:  Data Stream $S = < X_1, X_2, \cdots, X_d >$;
Output: Cluster Result.
variable-size sliding window size N = Ninit;
initialize an empty concept_list;
old density girds O;
temporary density grids T;
while data stream is active do
    add data stream $X = (X_1, X_2, \cdots, X_d)$ into sliding window;
    if (variable-size sliding window is FULL)
        map the data into T;
        if DCDD (O,T) $\geq \theta$
            Cluster(O);
            Cluster_Feature(O) add into Cluster_List;
            if No Train(Cluster_List) and size of Cluster_List $\geq \beta$
                obj = Train(Cluster_List);
            end if;
            clean O;
            O=T;
            if size of Cluster_List $\geq \beta$
                N = obj(Cluster_Feature(O))/2;
            end if
            else
                N = Ninit;
        end if
        else
            N = Adjust_Sliding_Window(O,T);
            clean O;
        end else
    end if
end while
```

Then we test our DCDD using the real data set KDD CUP-99, we set the parameters of XGBoost: booster is gbtree and others are default and set $\beta = 10000$, $\delta = 0.5$. In our experiment, we use the strategy in Sect. 4.3 to adjust the size of the variable sliding window in the first ten thousand concept drifts that are used to train and we use the prediction model to adjust it afterwards. From the result in Fig. 4(a), it is obvious that the number of detected drifting concepts by our DCDD decreases with the increase of α. The precision and recall of DCDD are presented in Fig. 4(b). In these experiment, the threshold value θ is set to 0.1 and the size of the sliding window is initialized to 100. The parameter α is set from 0.2 to 1 with a step length of 0.2.

With the increase in the number of detected concept drifts, the precision and recall of DCDD with model 1 are shown in Fig. 4(c), where the threshold value θ is set to 0.1.

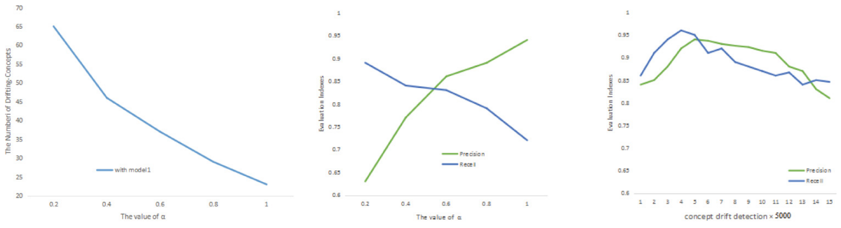
5.2 Result Comparison

We compare the experimental results of our DCDD with DCDA on F1-measure, where $F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$, and time consumption wrt the number of concept drifts. Firstly, the threshold value θ is set to 0.1 and α is set to 0.5. The results are shown in Fig. 5. We



(a) Precision (b) Recall

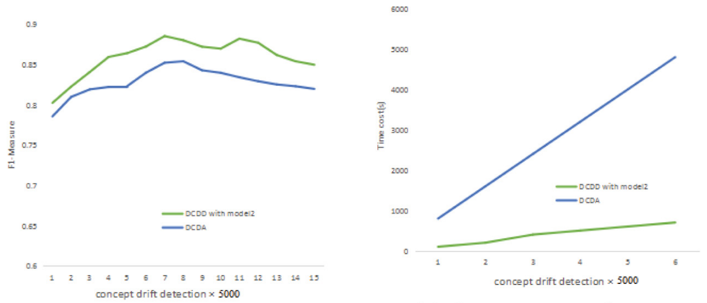
Fig. 3. Precision and recall of our DCDD and DCDA on synthetic dataset.



(a) (b) (c)

Fig. 4. Number of drifting concepts, precision and recall wrt to α and drift direction

can see from Fig. 5(a) that F1-measure of DCDD is slightly better than DCDA at the first ten thousand concept drift, and then it grows to a more significant level as the number of concept drifts increases. For comparison of time consumption shown in Fig. 5(b), it is clear that our DCDD has a much lower time cost than DCDA and runs about 8 times faster than DCDA.

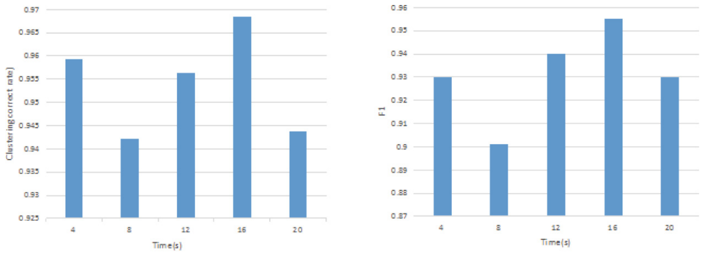


(a) Comparison of F1-measure

(b) Comparison of time cost for Concept Drift Detection

Fig. 5. Comparison between DCDD with DCDA on KDD-CUP dataset.

The accuracy of clustering results of our DCDD on KDD-CUP dataset is shown in Fig. 6 (a) and the F1-measure of clustering results in Fig. 6 (b). From the clustering results



(a) Clustering accuracy t (b) F1 measure on

Fig. 6. Performance of DCDD on KDD-CUP dataset

at different times, DCDD achieves not only good accuracy but also good F1-measure at different times that are better than the results of DCDA [2].

6 Conclusion

In this paper, we proposed DCDD, a new framework for concept drift detection based on density-based clustering [15]. It improves the DCDA algorithm [2] both in terms of the F1-measure and computational cost (quite significantly). Our algorithm depends only on the number of grids rather than the number of data points in the grids, which makes it much more efficient. In addition, we proposed a strategy and prediction model to adjust the variable-size sliding window to adapt to the changes of data streams and to further improve the efficiency. In the future we will address the periodicity of concepts in data streams to gain improvement in detection accuracy.

Acknowledgement. This work is supported by Macao Polytechnic University Research Grant RP/FCA- 13/2022. The corresponding author is Hong Shen.

References

1. Aggarwal, C.C., Yu, P.S., Han, J., Wang, J.: A framework for clustering evolving data streams. In: International Conference on Very Large Data Bases, pp. 81–92 (2003)
2. Cao, F., Liang, J., Bai, L., Zhao, X., Dang, C.: A framework for clustering categorical time-evolving data. IEEE Trans. Fuzzy Syst. **18**(5), 872–882 (2010)
3. Chen, H.L., Chen, M.S., Lin, S.C.: Catching the trend: a framework for clustering concept-drifting categorical data. IEEE Trans. Knowl. Data Eng. **21**(5), 652–665 (2009)
4. Chen, T., He, T., Benesty, M.: Xgboost: extreme gradient boosting (2015)
5. Chen, Y., Tu, L.: Density-based clustering for real-time stream data. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 133–142 (2007)
6. Chi, Y., Song, X., Zhou, D., Hino, K., Tseng, B.L.: Evolutionary spectral clustering by incorporating temporal smoothness. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, pp. 153–162 (2007)

7. Cai, B., Hu, C., Ren, J.: Clustering over an evolving data stream based on grid density and correlation. ICIC Exp. Lett. **4**5(A), 1603–1609 (2010)
8. Corne, D., Handl, J., Knowles, J.: Evolutionary clustering. In: Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, pp. 332–337 (2006)
9. Cui, Z., Shen, H.: The framework of relative density-based clustering. In: Chen, G., Shen, H., Chen, M. (eds.) PAAP 2017. CCIS, vol. 729, pp. 343–352. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-6442-5_31
10. Gaber, M.M., Yu, P.S.: Detection and classification of changes in evolving data streams. Int. J. Inf. Technol. Decis. Mak. **05**(5), 659–670 (2006)
11. Granger, R.H., Schlimmer, J.C.: Beyond incremental processing: tracking concept drift. In: Proceeding of the Twenty-Second International Conference on Very Large Databases, pp. 502–507 (1986)
12. Jia, C., Tan, C.Y., Yong, A.: A grid and density-based clustering algorithm for processing data stream. In: International Conference on Genetic and Evolutionary Computing, pp. 517–521 (2008)
13. Pawlak, Z.: Rough sets. Int. J. Comput. Inform. Sci. **11**(5), 341–356 (1982)
14. Ren, J., Cai, B., Hu, C.: Clustering over data streams based on grid density and index tree. J. Converg. Inf. Technol. **6**(1), 83–93 (2011)
15. Sander, J., Ester, M., Kriegel, H.P., Xu, X.: Density-based clustering in spatial databases: the algorithm gdbscan and its applications. Data Min. Knowl. Disc. **2**(2), 169–194 (1998)
16. Souza, V.M.A., Chowdhury, F.A., Mueen, A.: Unsupervised drift detection on high-speed data streams. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 102–111 (2020)
17. Tsymbal, A., Pechenizkiy, M., Cunningham, X.: Dynamic integration of classifiers for handling concept drift. Information Fusion **9**(1), 56–68 (2008)
18. Tu, L., Chen, Y.: Stream data clustering based on grid density and attraction. ACM Trans. Knowl. Discov. Data **3**(3), 167–176 (2009)
19. Wang, P., Jin, N., Fehringer, G.: Concept drift detection with false positive rate for multi-label classification in iot data stream. In: 2020 International Conference on UK-China Emerging Technologies (UCET), pp. 1–4 (2020)

An End-to-End Multiple Hyper-parameters Prediction Method for Distributed Constraint Optimization Problem

Chun Chen[1](✉), Yong Zhang[2], Li Ning[3], and Shengzhong Feng[4]

[1] Shenzhen Institute of Information Technology, Shenzhen, China
chun.chen@siat.ac.cn
[2] Shenzhen Institute of Advanced Technology, Shenzhen, China
[3] University of Electronic Science and Technology of China, Chengdu, China
[4] National Supercomputing Center in Shenzhen, Shenzhen, China

Abstract. Distributed Constraint Optimization Problem (DCOP) is an important model for multi-agents, has been widely used in various fields. When a large scale of DCOP implement on the supercomputer, various parameters need to choose, and the complement time vary widely for different combinations of parameters. Automatically provided accurate operating parameters for DCOP can improve the operation speed and enables the rational use of computational resources. However, the number of hyper-parameters of DCOP is huge, and correlation exists between hyper-parameters, thus make the prediction of multiply hyper-parameters difficult. In this paper we propose a new framework combine graph neural network and recurrent neural network. The performance shows that our framework can outperform the SODA method.

Keywords: multiply hyper-parameter · DCOP · Graph neural network · recurrent neural network

1 Introduction

The rapid development of artificial intelligence has attracted researchers' attention on multi-agent systems. The Distributed Constraint Optimization Problem (DCOP), as an important research direction on multi-agent, has been widely used in various fields in recent years. With the exponential growth of scale for DCOP, supercomputers have become the primary choice to cope with large-scale DCOP due to the storage and computing capacity of traditional computers. Nowadays, the common way to solve DCOP is to calculate the operating parameters by users who masters the domain knowledge and provide them to the supercomputing platform, which time-consuming and labor-intensive. So automatically provided accurate operating parameters for DCOPs can improve the operation speed and enables the rational use of computational resources.

The prediction of DCOP hyper-parameters is difficulty. First, DCOP involves many hyper-parameters, such as the DCOP algorithm and the corresponding parameters, the

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 202–214, 2024.
https://doi.org/10.1007/978-981-99-8211-0_19

graph partitioning algorithm. Second, correlation exists between hyper-parameters of DCOP. The performance under a single optimal parameter do not guarantee the overall optimal.

The multiply hyper-parameter prediction of DCOP can be simply considered as a multi-label recognition problem [3, 10]. However, the data of both image [4–9] and text problem are [11–14] regular, while the graph representation of DCOP is irregular data, so it is not possible to directly apply the present multi-label classification methods to the prediction for the hyper-parameter set of DCOP.

This paper addresses the difficulties of DCOP multiply hyper-parameter prediction and proposes a multiply hyper-parameter prediction framework combining graph neural network and recurrent neural network, whose contributions include the following:

(1) As there is no research on multi-parameter prediction for DCOPs, this paper gives the basic definition of the optimal parameter set and turns the DCOP multiply hyper-parameter prediction problem into a multi-label classification problem.

(2) For the multiply hyper-parameter prediction problem, this paper proposes a GRNN (Graph Recurrent Neural Networks) frameworks combining graph neural networks and recurrent neural networks, which considering the correlation of each parameter. The framework learned the features of the DCOP constraint graph by graph neural networks and handled the higher order parameter correlations by recurrent neural network.

(3) The extraction accuracy of graph feature vectors can affect the prediction accuracy of multiply hyper-parameter. This paper explores the influence on the number of layers of the graph neural network.

This paper is organized as follows: Sect. 2 introduces the basic theory of DCOP multiply hyper-parameter prediction and transforms the DCOP multiply hyper-parameter prediction problem into a multi-label classification problem, Sect. 3 introduces the multiply hyper-parameter prediction framework--- GRNN in detail, Sect. 4 analyzes the experimental results and discusses the experimental results and summarizes in Sect. 5.

2 Background

The performance of DCOP on supercomputing platforms are often associated with multiply hyper-parameter, such as graph partitioning algorithm [15], the DCOP algorithm, and the parameters corresponding to that algorithm. Before to predict the optimal hyper-parameter set, the definition of the optimal set of parameters OPT_{para} is given.

2.1 Definition of Optimal Set of Parameters

Given an DCOP instant and the overall sets of parameters for the instance $P_{para} = \{Para_1, Para_2, \ldots, Para_N\}$. For each set of parameters $Para_i$, which includes the execution method E_m, the algorithm A and the parameters corresponding to that algorithm $P_A = \{P_{A_1}, P_{A_2}, \ldots, P_{A_f}\}$, the graph partitioning algorithm G_p and the number of cores k, where $Para_i = [E_m, \left\{P_{A_1}, P_{A_2}, \ldots, P_{A_f}\right\}, G_p, k]$. The goal for this paper is to search the optimal set of parameters $Para_{opt}$ with the minimization completion time.

Firstly, we give a definition for the completion time of any instance under the set of parameters $Para_i$. If the $Para_i$ is selected, the instance implement the graph partitioning algorithm G_p to divide the DCOP's constrained graph into k parts and call the DCOP algorithm A and the parameters under the algorithm A to run the instance (synchronously or asynchronously) on k processes for a total of n rounds. Define the effective running time of the i_th round under the parameter $Para_i$ to be t_{pi_j}, which is the time for the cost function of DCOP to reach 0. This paper assumes that each instance of DCOP is solvable, i.e., there exists an effective time for the cost function to reach 0.

As the law of large numbers (LLN) in probability theory, where the average obtained from multiple experiments should be close to the expectation when performing the same experiment with multiple times, and the average will be closer to the expectation as the number of experiments increases. So, in this chapter, the completion time of any instance under the set of parameters is defined as the average completion time.

$$t_{pi} = \frac{\sum_{j=1}^{n} t_{pi_j}}{n} \tag{1}$$

where t_{pi} is the completion time of the j_th round of DCOP under the parameter set $Para_i$ and n is the total number of rounds run.

when the completion time of any instance under the set of parameters is defined then this paper defines the optimal set of parameters as follows:

$$Para_{opt} = argmin(t_{p1}, t_{p2}, \ldots, t_{pN}) \tag{2}$$

where N is the total capacity of the parameters P_{para}.

2.2 Comparison with Different Sets of Parameters

This paper introduces a small example, graph coloring problem, to compute $Para_{opt}$, and gives a representation of the completion time under different parameters sets. The example divides the constrained graph of DCOP into 1–4 subgraphs by the Giran-Newman algorithm or the METIS algorithm for graph partitioning. Each subgraph is then placed on the corresponding core to implement using the DCOP algorithm (DSA) either synchronously or asynchronously.

The result under some sets of parameters is showed in Fig. 1, which the example contains a total of 6 cases, and 10 rounds are executed for each set of parameters. The completion time is calculated by Eq. (2) and the optimal parameter for this example is obtained from Eq. (2) which is $P_{para} = \{\text{sy, Giran} - \text{Newman, DSA, p} = 0.7, 3\}$.

As shown in Fig. 1, the results under different sets of parameters are different and irregular, thus it hard to find the optimal parameters according to the traditional statistical methods. With the rapid development of neural networks, the multi-label classification solution problem has matured. In this paper, we will transform the multiply hyper-parameter prediction problem into a multi-label classification problem which using the optimal parameters as labels.

This paper gives the definition of multiply hyper-parameter prediction. For Each DCOP instance $G_i \in R_m$, which owns L subsets y in the parameter label space Y. The multi-parameter prediction task is to learn a function $h: $h : R_m \rightarrow 2Y^D = \{(x_i, y_i)|1 \leq$

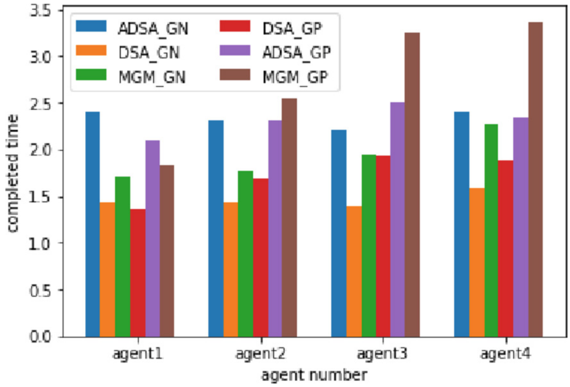


Fig. 1. The completion time under sets of parameters for graph coloring problem

i ≤ N}, where N is the total amount of training data, x_i is the vector in the input feature R_m of the i_th instance and $y_i \subset Y$ is a subset of the label space Y. Unlike the multi-classification problem where each instance is assigned only one label, the generalization of the multilabel problem provides multiple label assignments for each instance at the same time.

3 Multiply Hyper-parameter Prediction Model

In this section, a neural network framework---GRNN is proposed to predict the multiply hyper-parameters set, as shown in Fig. 2. The framework consists of three modules, the preprocessing module, the graph representations feature extraction module, and the multilabel prediction module. The preprocessing module converts the DCOP into a graph representation and extract the fixed-length feature vectors by the graph representations

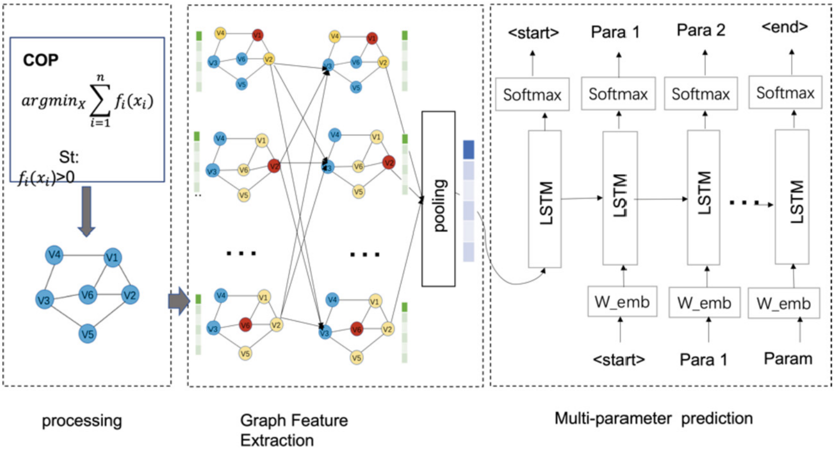


Fig. 2. Multiply hyper-parameter Prediction Framework Diagram

feature extraction module. Then, according to these feature vectors, the higher-order correlations between parameters are modeled in the multi-parameter prediction module.

3.1 Preprocessing Module

The DCOP cannot be solved directly in a graph neural network, [1, 2, 17] it needs to convert the DCOP into a graph representation. Since this paper may involve multiple graph representations, where different algorithms correspond to different graph representations. To ensure consistency, this paper uniformly converts the DCOP into a constraint graph.

3.2 Graph Feature Extraction Module

For the feature vector extraction of the DCOP, the GraphSNN is chosen in this paper. This network maps the local structure into the aggregation, considering not only the features of the neighbors but also the overlapping subgraphs. The feature extraction of DCOP includes node feature extraction as well as graph feature extraction.

3.2.1 Node Feature Extraction

Regarding node feature extraction, to better describe the neighborhood relationship between vertex v and its neighbors u, GraphSNN defines structural coefficients $\omega(S_v, S_{uv})$ for each vertex v, $\omega : S \times S^* \to R$.

$$\omega(S_v, S_{uv}) = \frac{|E_{vu}|}{(|V_{vu}||V_{vu} - 1|)|V_{vu}|^{\lambda}} \tag{3}$$

where $\omega(S_v, S_{uv})$ is the structure coefficient of vertex v and its neighbors. S_v is the neighborhood subgraph of vertex v and S_{uv} is the set of overlapping subgraphs of vertex v with $\lambda > 0$. $\omega(S_v, S_{uv})$ satisfies the properties of local compactness, local denseness, and isomorphism invariance. Let its adjacency matrix be $A = (A_{uv})_{uv \in V}$, where $A_{uv} = \omega(S_v, S_{uv})$.

GraphSNN also defines a weighted adjacency matrix $\mathrm{A} = (\overline{A_{uv}})_{uv \in V}$, where $\overline{A_{uv}}$ is the normalized value of A_{uv}, $\overline{A_{uv}} = \frac{A_{uv}}{\sum_{u \in N(v)} A_{uv}}$. So, the node eigenvectors of v are updated as

$$\begin{aligned} m_a^t &= \mathrm{Aggregate}^N(\{A_{vu}, h_u^t)|u \in N(v) \\ m_v^t &= \mathrm{Aggregate}^I(A_{vu}|u \in N(v))h_v^t \\ h_{t+1}^v &= Combine(m_v^t, m_a^t) \end{aligned} \tag{4}$$

$\mathrm{Aggregate}^N(*)$ and $\mathrm{Aggregate}^I(*)$ are two different parameterized cumulative functions. Where m_a^t is the information aggregated from the neighbors v and their structural coefficients, m_v^t after performing the multiplication between the cumulative function

Aggregate$^{I}(*)$ and the multiplication between the eigenvectors, the “adjusted” message from v to account for the structural effects of its neighbors.

Specifically, the update function of GraphSNN for each vertex $v \in V$, whose node feature vector at $t+1$ layer is

$$h_v^{t+1} = MLP_\theta \gamma^t (\sum\nolimits_{v \in N(u)} A_{uv} + 1) h_v^t \sum\nolimits_{u \in N(v)} A_{uv} + 1) h_u^t) \tag{6}$$

where γ^t is a scalar parameter that can be learned. $N(v)$ refers to the one-hop neighbors v, and multiple layers can be stacked to handle more than one-hop neighbors. Note that to ensure the Monolicity of feature aggregation in the presence of structural coefficients, add 1 to the first and second terms of Eq. (6).

3.2.2 Graph Feature Extraction

For the graph classification problem, all node features in the graph need to be transformed into graph features, and the whole graph is represented as h_G.

$$h_G = Readout(h^k | v \in G) \tag{7}$$

where h_G is the graph G denotes the vector and Readout denotes the substitution invariant function, which can also be a graph-level pooling function.

The Readout function of the GraphSNN framework is single-shot. To consider all the structural information, the GraphSNN framework utilizes the information from all iterations of the model and uses an architecture similar to Jumping Knowledge Networks. The graphs represent connections in all iterations/layers and the *Readout* function sums all node features from the same iteration.

$$h_G = Concat(\sum\nolimits_{u \in N(v)} h_v^k | k = 0, 1, \ldots, K) \tag{8}$$

3.3 Hyper-parameter Prediction Module

After graphSNN obtains the representation graph vector of DCOP, the parameter prediction module uses the output graph vector of graphSNN as the initial state input for label prediction. Because there is some correlation before the parameters, for this reason LSTM is chosen in this paper to predict multiple parameters.

Despite the existence of several LSTM variants, this paper selects the standard LSTM, and applies an additional word embedding layer for the labels. The LSTM consists of three gates: an input gate i, an output gate o, and a forgetting gate f. The three gates work in concert to control what is read on the input, what is output, and what is forgotten, allowing some complex long-term relationships to be modeled.

$$\begin{aligned}
i &= \sigma(W^{(i)} x_t + U^{(i)} h_{t-1} + b^{(i)}) \\
o &= \sigma(W^{(o)} x_t + U^{(o)} h_{t-1} + b^{(o)}) \\
f &= \sigma(W^{(f)} x_t + U^{(f)} h_{t-1} + b^{(f)}) \\
u &= tanh(W^{(u)} x_t + U^{(u)} h_{t-1} + b^{(u)}) \\
c_t &= i \odot u + f \odot c_{t-1} \\
h_t &= o \odot \tanh(c_t)
\end{aligned} \tag{9}$$

where $\sigma(*)$ denotes element-by-element multiplication, which is a sigmoid function. $x_t \in R_d$ is the input of the lower layer at time step t. If the lower layer is a word embedding of parameters, then d can be the dimension of the labeled word vector or can be the hidden state dimension of the lower layer, if the lower layer is an LSTM. If there are q LSTM units, then for all types (i, o, f, u), $h_t \in R_q$, $W(*) \in R_q \times d$ and $b(*) \in R_q$.The memory cell c_t is the key in the LSTM, which maintains long-term dependencies while getting rid of the gradient disappearance/explosion problem. The forget gate f is used to erase some parts of the memory cell, while the input gate i and the output gate o control what is read from and written to the memory cell.

LSTM by linear transformation as Eq. (9) in each type (i, o, f, u) with additional terms$W(T)T$, where T is the output constraint graph feature from GNN with fixed dimension$t$, $W(T) \in R_q \times t$, q is the hidden dimension of LSTM, e.g. input The formula for the gate reads.

$$i = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + W(T)T) \tag{10}$$

The label sequence prediction always starts with the tag < START >. At each time step, there is a SoftMax layer on top of the LSTM top layer. The probability of each label is calculated by first applying a linear transformation to the hidden state of the top LSTM layer.

Then, the tag with the highest probability is predicted. The prediction of the tag ends with the < END > tag. Therefore, for each input DCOP, a sequence of labels of different lengths is predicted. Ideally, the label sequence for each input DCOP matches exactly with the subset of labels belonging to that input DCOP.

4 Experimental Results and Analysis

4.1 Experimental Data

In the paper, we choose the graph coloring problem, a typical model of DCOP, to generate the corresponding dataset of this experiment. The datasets consist of two main parts, one part is the description of the DCOP and the corresponding constraint graph, and the other part is the label set which correspond to the multiply hyper-parameter. In this chapter, these two parts are introduced separately.

4.1.1 DCOP Problem Description

The graph coloring problem is a typical DCOP that has been widely used in coordination algorithms for sensor networks as well as benchmark, and many DCOP algorithms have also used it for performance comparisons.

In the distributed graph coloring problem, variables are located at the nodes of the constraint graph and choose a color (i.e., $x_i \in$ (1, ..., c) to avoid conflicts (i.e., choosing the same color) with other variables(nodes) connected to themselves through edges. Thus, the cost of each variable is expressed as

$$U_m(x_m) = \gamma_m(x_m) - \sum\nolimits_{i \in \frac{N(m)}{m}} x_i \otimes x_j \tag{11}$$

where, $x_i \otimes x_j = \begin{cases} 10 \; if \; x_i == x_j \\ 0 \quad otherwise \end{cases}$, $\gamma_m(x_m) \ll 1$, reflecting the preference of the variable for any color in the absence of conflict. Consistent with the DCOP definition, the goal is to find the state of each variable that minimizes the conflict. In this experiment, this paper sets $\gamma_m(x_m) = 0$ and sets the edge conflict cost, i.e., two nodes with edges in the constraint graph choose the same color, $x_i \otimes x_j = 10$.

4.1.2 Random Graph Generation Based on Graph Coloring DCOP

In this experiment, three kinds of undirected, unweighted and connected random graphs are generated by network further four datasets are selected which cover the basic random graphs, etc.

1) dataset contains 438 random graph instances of 11 colors generated by the Erdős - Rényi model, which 316 instances are generated by the gnm function with 200 nodes and 200–400 edges, and 122 instances are generated by the gnp function with an link probability from 0.1–0.2.
2) The second dataset has 29 instances of 11-color random graph coloring, which consists of instances generated by the Small wolrd model.
3) The third dataset has 100 instances of 11-color random graph coloring. The instances are generated by the Barabasi Albert model. The random graph degree generated by this model has a power-law distribution.
4) The fourth dataset are assembled the above three datasets.

4.1.3 Hyper-parameter Set Generation and Validity

Since the goal of this paper is to find the set of optimal hyper-parameters, and the framework is a supervised learning framework, this section starts by labeling each random graph with the original label. To ensure the accuracy of the prediction, this chapter needs to ensure the validity of the labels and that the initial assignment is robust. To verify the validity of the framework, this paper selected DCOP algorithms such as DSA, MGM, etc., and the Giran- Newman algorithm as well as the METIS graph partitioning algorithm.

To ensure the validity of the labels, this paper conducts 10 trials for each set of hyper-parameters and hopes that the results of each set of hyper-parameters on experiments are stable, i.e., the variance is not large. In this paper, we analyze the results of each set of hyper-parameters as shown in Fig. 3.

From Fig. 3, we finds that the variance of the fitted curve coefficients is low, about 0.26 times the mean. The expected time can be considered as the label of the constrained graph.

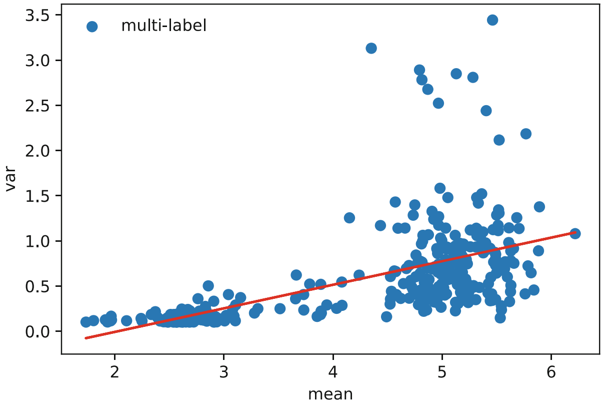


Fig. 3. The relationship between expected running time and variance

For the label distribution, we run multiple DCOP problems in this paper and find a more uniform parameter distribution, as shown in Fig. 4. The figure shows the optimal parameter distribution of the dataset $\mathrm{DCOP}_{\mathrm{BA}}$ after multiple rounds of experiments.

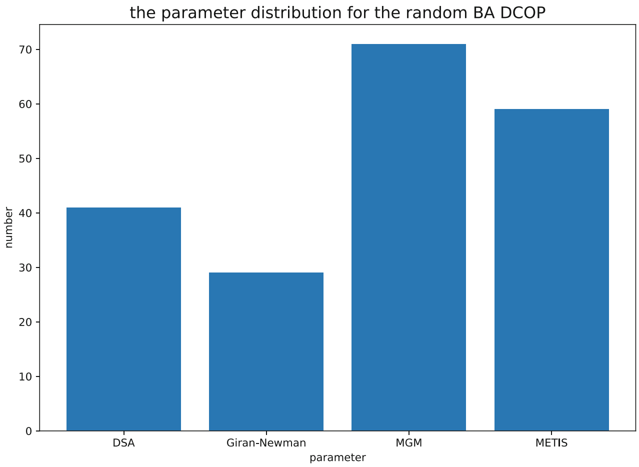


Fig. 4. The distribution for BA parameter

4.1.4 Dataset Description

For this purpose, the structural information of the four datasets trained and the labeling information are described in this paper as follows, as shown in the Table 1, Where $\mathrm{DCOP}_{\mathrm{ER}}$ is dataset 1, a random graph generated for the Erdős - Rényi model, $\mathrm{DCOP}_{\mathrm{BA}}$ is dataset 2, a random graph generated by the Barabasi Albert model, $\mathrm{DCOP}_{\mathrm{SW}}$ is dataset 3, a random graph instance generated for the SW model, $\mathrm{DCOP}_{\mathrm{ALL}}$ is data set 4, which is the merge of the above four data sets.

Table 1. Dataset description

data	number	Degree of nodes	Number of edges	Number of hyper-parameters
DCOP_{ER}	438	3.6/1.94/1/15	359.88/81.96/200/546	3
DCOP_{BA}	100	1.99/2.81/1/51	199/0/199/199/199	4
DCOP_{SW}	29	5.56/1.44/4/12	556.28/1.44/4/12	3
DCOP_{ALL}	567	3.42/2.24/1/51	341.55/110.07/199/737	4

4.2 Experimental Results and Analysis

4.2.1 Evaluation Index

To fairly compare the results of other methods, the average precision (CP) is reported in this section for performance evaluation.

$$CP = \frac{1}{c}\sum\nolimits_i \frac{N_i^c}{N_i^p} \tag{12}$$

4.2.2 Parameter Setting and Running Platform

All experiments were performed on a server with an Intel Xeon CPU 4110 equipped with 20 2.20 GHz cores. The system was Linux 3.10.0 and all DCOPs were implemented in the PyDCOP library. All multiclassification graph neural networks were implemented in pytorch.

This paper uses the Adam optimizer [16] with $\lambda = 1$. For all datasets of DCOP, the model was trained for 500 periods with a learning rate of 0.01, a loss rate of 0.5, a hidden layer of 256, and $\gamma = 0.1$. This chapter select the random division method, i.e., the graph is randomly divided into 60\%, 20\% and 20\% for training, validation, and testing.

4.2.3 Analysis of Experimental Results

Since this paper is required to calculate the optimal parameters, in order to verify the effectiveness of the algorithm, two common methods are compared: 1) ordinary dichotomous GNN, i.e., GNN is used to generate the graph features of the DCOP constraint graph, for each label, which is treated as a one-by-one dichotomous classification in this chapter. (2) Since this paper does not involve many parameters, the multi-classification method graphSNN is chosen as a comparison experiment. Since the parameters involved in this chapter are less, the method converts the multi-parameter prediction into a multi-classification method and puts it into the graphSNN network. The results are shown in Table 2.

Table 2 lists the accuracy results of multiply hyper-parameters prediction for different datasets. The results show that the accuracy of both the multiclassification algorithm---GraphSNN and the GRNN algorithm on all the datasets is higher than that of the ordinary binary classification algorithm GNN. For the GraphSNN algorithm and GRNN

Table 2. The accuracy for the prediction

dataset	Algorithm	Accuracy
DCOP_{ER}	GNN	76.53 ± 3.12
	GraphSNN	93.84 ± 2.28
	GRNN	84.74 ± 4.34
DCOP_{BA}	GNN	42.31 ± 4.58
	GraphSNN	46.0 ± 11.13
	GRNN	58.67 ± 2.66
DCOP_{SW}	GNN	70.75 ± 2.94
	GraphSNN	91.16 ± 5.49
	GRNN	86.4 ± 10.88
DCOP_{ALL}	GNN	73.17 ± 1.38
	GraphSNN	81.13 ± 2.44
	GRNN	93.52 ± 2.47

algorithm, the accuracy of the recurrent neural network does not play a larger role when there are few labels, and its accuracy is not as good as that of the GraphSNN, and its ER dataset and WS dataset both perform less well than the GraphSNN when there are only three labels. In the ER dataset, the accuracy of GraphSNN is 10.7% higher than that of GRNN method, and in the SW dataset, the accuracy is 5.8% higher. However, the accuracy of GRNN improves as the number of labels increases, and it improves by 27.54% in the BA dataset and 14% in the total dataset compared to the GraphSNN.

4.2.4 The Effect of Graph Neural Network Depth on Performance

The exaction of graph features is one of the most important factors affecting multi-parameter prediction, and different graph neural network embedding operations have an impact on the performance of experimental results. Specifically, different layers of graph neural networks have different obtained graph features. For this reason, this section explores the effect of different neural network layers on the prediction results, as shown in the Table 3.

Table 3. The accuracy on different neural network layers

Dataset	2 layers	3 layers	4 layers
DCOP_{ER}	84.74 ± 4.34	82.36 ± 3.74	81.47 ± 7.41
DCOP_{BA}	58.67 ± 2.66	57.49 ± 4.51	55.16 ± 2.73
DCOP_{SW}	86.4 ± 10.88	83.14 ± 7.29	82.46 ± 5.29
DCOP_{ALL}	93.52 ± 2.47	91.45 ± 4.28	89.54 ± 6.85

It finds that the accuracy of the prediction results to decrease to some extent when increasing the number of layers of the convolutional layers. The possible reasons for this are mainly due to the following two points. One is that the number of parameters will also become dramatically larger due to the increase in the number of layers of the convolution of the graph which will cause the overfitting phenomenon to some extent. Second, although the method in this paper greatly alleviates the over-smoothing problem, but it does not avoid the over-smoothing problem, and the deepening of the convolutional layer will add the over-smoothing problem leads to performance degradation.

5 Summary

Multiply hyper-parameter prediction of DCOP is an important subject, and its high accuracy can be an effective guarantee of DCOP. This paper first demonstrates experimentally that DCOP has large differences in its operation results under multiple sets of parameters and that traditional methods cannot effectively predict multiple parameters accurately. Then transforms the multiply hyper-parameter prediction problem of DCOP into a multi-label prediction problem and proposes a novel neural network-based multi-label classification method. Experiments demonstrate the effectiveness of the methods from both qualitative and quantitative perspectives, respectively.

However, the GRNN is a supervised prediction models, which require multiple runs of DCOP to generate the corresponding training data, and the data acquisition cost is relatively expensive. In the future, we will learn new techniques such as semi-supervised graphical neural networks to solve the problem.

References

1. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search (2019). arXiv:1806.09055
2. Schweidtmann, A.M., Rittig, J.G., König, A., et al.: Graph neural networks for prediction of fuel ignition quality. Energy Fuels **34**, 11395–11407 (2020)
3. Zhang, J., Wu, Q., Shen, C., et al.: Multilabel image classification with regional latent semantic dependencies. IEEE Trans. Multimed. **20**, 2801–2813 (2018)
4. Chen, Z.M., Wei, X.S., Wang, P., et al.: Multi-label image recognition with graph convolutional networks. IEEE/CVF Conf. Comput. Vision Pattern Recogn. (CVPR) **2019**, 5172–5181 (2019)
5. Wang, Y., Xie, Y., Liu, Y., et al.: Fast graph convolution network based multi-label image recognition via cross-modal fusion. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management (2020)
6. Chen, T., Xu, M., Hui, X., et al.: Learning semantic-specific graph representation for multi-label image recognition. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 522–531 (2019)
7. You, R., Guo, Z., Cui, L., et al.: Cross-modality attention with semantic graph embedding for multi-label classification. arXiv:abs/1912.07872 (2020)
8. Zhang, M., Shao, H.C., Song, G., et al.: Top-1 solution of multi-moments in time challenge. arXiv:2003.05837 (2019)
9. Zhao, J., Yan, K., Zhao, Y., et al.: Transformer-based dual relation graph for multi-label image recognition. IEEE/CVF Int. Conf. Comput. Vision (ICCV) **2021**, 163–172 (2021)

10. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP (2014)
11. Lai, S., Xu, L., Liu, K., et al.: Recurrent convolutional neural networks for text classification. In: AAAI (2015)
12. Chen, G., Ye, D., Xing, Z., et al.: Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. Int. J. Conf. Neural Netw. (IJCNN) **2017**, 2377–2383 (2017)
13. Yang, Z., Yang, D., Dyer, C., et al.: Hierarchical attention networks for document classification. In: NAACL (2016)
14. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune BERT for text classification? In: Sun, M., Huang, X., Ji, H., Liu, Z., Liu, Y. (eds.) Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings, pp. 194–206. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-32381-3_16
15. Pizzuti, C.: Evolutionary computation for community detection in networks: A review. IEEE Trans. Evol. Comput. **22**(3), 464–483 (2018). https://doi.org/10.1109/TEVC.2017.2737600
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR abs/1412.6980 (2015)
17. Pei, H., Wei, B., Chang, K.C.C., et al.: Geom-GCN: geometric graph convolutional networks. arXiv:2002.05287 (2020)

Dynamic Priority Coflow Scheduling in Optical Circuit Switched Networks

Hongkun Ren[1](✉), Hong Shen[2], and Xin Wang[1]

[1] School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China
renhk3@mail2.sysu.edu.cn

[2] Faculty of Applied Sciences, Macao Polytechnic University, Macao, China

Abstract. OCS (Optical Circuit Switch) is increasingly popular for accelerating data transmission of coflows due to its higher bandwidth and lower power consumption compared with EPS (Electronic Packet Switch), where a coflow is a collection of related parallel flows between two computation stages in data-intensive applications. However, the extra port constraints and reconfiguration delay of OCS obstruct the efficiency of OCS operations. This paper studies the problem of coflow scheduling in the OCS of datacenter networks to minimize the total Coflow Completion Time (CCT). We propose a Dynamic Priority Coflow Scheduling Algorithm that schedules coflows preemptively by considering coflow transmission time and OCS reconfiguration delay jointly to dynamically update each coflow's priority, which can significantly reduce the waiting time of small coflows and reduce head-of-line blocking. Extensive simulations based on Facebook data traces show that our approach outperforms the state-of-the-art scheme OMCO [19] significantly, and transmits multiple coflows 1.30× faster than OMCO.

Keywords: Optical Circuit Switch · Coflow Scheduling

1 Introduction

Coflow [3] is proposed as the collection of related parallel flows between two comptation stages to handle the communication requirements of data-parallel applications like MapReduce [7]. Recent works have shown that the intermediate data transmission in a datacenter network (DCN) accounts for more than 50% of applications' completion time [5], and scheduling flows at coflow-level can significantly reduce the completion time of the communication stages. Nowadays, there are two main types of switches in DCNs, Electronic Packet Switch (EPS) and Optical Circuit Switch (OCS). With the growing demand for data transmission in DCNs, OCS is increasingly deployed in the next generation data center due to its advantages such as higher bandwidth and lower power consumption [14].

We study the online multiple coflow scheduling problem in OCS, aiming at minimizing the total coflow completion time (CCT) which is defined as the duration from its arrival to the completion of all its flows [23]. While OCS offers much higher bandwidth than EPS for data transmitting, it suffers from the deficiency of less flexibility than EPS

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 215–226, 2024.
https://doi.org/10.1007/978-981-99-8211-0_20

for accommodating flow dependency due to the non-blocking requirement that only a single path can be established between any pair of ingress and egress ports in the OCS. Worse still, it will take a non-negligible delay to reconfigure new circuits in OCS [13]. Reducing from the open-shop problem, it has been shown that scheduling a single coflow in OCS is NP-hard [8], letting alone the multi-coflow online scheduling problem this paper studies.

Researchers have proposed many algorithms [6, 17, 18] to minimize the total CCT in EPS. For example, Varys [6] schedule coflows based on the Smallest-Effective-Bottleneck-First principle. By calculating all flows' largest ratio of size to the bandwidth of corresponding ports, which is defined as the bottleneck completion time, Varys greedily choose the coflow of the minimum bottleneck time to be scheduled first. Adopting Multi-level Feedback Queue Algorithm, Aalo [4] prioritize each queue by the volume of sent data bytes to schedule coflows without prior knowledge of their size. In contrast to the deep learning methods [17], which are quite not robust, all the above mentioned works greedily give higher priority to coflows with smaller data size. Inspired by them, researchers also employ a greedy algorithm of scheduling the coflows with the smallest volume to be sent in a specific port in OCS [19]. However, this work neglects the reconfiguration delay of coflows and thus violates the principle of shortest coflow first.

Recent work has found that preemption plays an important role in coflow scheduling in OCS [9] and proposed a preemptive multi-coflow scheduling algorithm. However, it only works when all coflows arrive in the network at the same time, which contradicts with the practical situation that requests of transmitting coflows generate from time to time. To meet the requirements of online scheduling, OMCO [19] orders all the pending coflows and transmits them one-by-one in the non-preemptive way, but it still faces the challenges of Head-of-line (HOL) blocking, which increases the waiting time of newly-arrived small coflows.

To overcome these drawbacks, we propose an online algorithm to schedule multiple coflows in OCS preemptively for minimizing the total CCT. Our design is described as follows. We first decompose the demand matrix of each coflow into a series of configuration plans by the classic Birkoff-von Neumann (BvN) decomposition algorithm [20], and then we prioritize the coflows by their transmission time (determined by coflow size) and reconfiguration delay of OCS jointly. We schedule the coflows according to their priorities following SJF (Shortest-Job-First), update each coflow's priority when new coflows arrive, and preempt the current executing coflow if a new coflow has a higher priority. We summarize our contributions below:

- We mathematically formulate the online mutiple-coflow scheduling problem.
- We propose a novel algorithm for online multi-coflow scheduling in OCS by jointly considering coflow transmission time and OCS reconfiguration delay with preemption enabled to decrease the overall blocking time.
- We conduct extensive simulation experiments to verify the performance superiority of our algorithm, i.e., transmitting multiple coflows is 1.30× faster than OMCO.

2 Motivation

Prior work [19] only prioritizes coflows according to each coflow's demand matrix diameter (maximum sum of row or column elements, reflecting the coflow transmission time), and then irrevocably one-by-one schedules these coflows following the given priority. This evidently brings two problems:

1. How to prioritize coflows a small demand matrix's diameter but a large number of OCS configuration plans?
2. How to adjust in-system coflow priorities when new coflows arrive online?

To solve these two problems, we schedule coflows preemptively by jointly considering the coflow transmission time (diameter of the demand matrix) and the reconfiguration delay of OCS to best utilize the bandwidth of OCS and decrease the total CCT. In the following, we will summarize the drawbacks of existing online coflow scheduling algorithms in OCS and detail the necessity of our designs.

2.1 Drawbacks of diameter-based scheduling

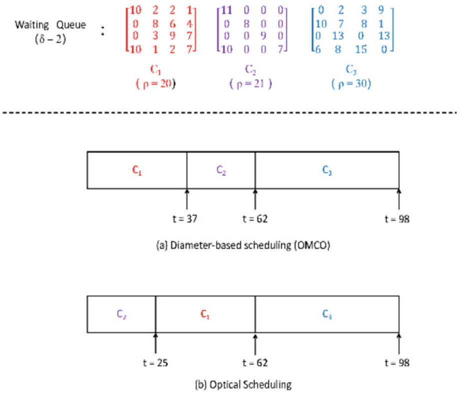
Let the diameter of the coflow's demand matrix be ρ. The existing algorithm Online Multiple Coflow Scheduling (OMCO) [19] greedily schedules the first coflow with the smallest ρ repeatedly without taking into consideration of the switch reconfiguration cost for realizing non-blocking flow-transmission of all coflows in this order. As shown in Fig. 1. A Motivating Example, there are three coflows C_1, C_2 and C_3 arrive simultaneously, of which diameters are $\rho = 20/21/30$, respectively. When we schedule coflows one by one, the OMCO schedules the coflows in the order of $< C_1, C_2, C_3 >$, and the corresponding total CCT is $37 + 62 + 98 = 197$, as shown in Fig. 1. A Motivating Example. However, if we schedule the coflows in the order of $< C_2, C_1, C_3 >$, the total CCT decreases to $25 + 62 + 98 = 185$, as shown in Fig. 1. A Motivating Example. In OCS, the coflow completion time (CCT) consists of the transmission time and the configuration delay, where the diameter ρ can reflect the transmission time. The OMCO algorithm only runs coflows of the smallest diameter first, ignoring the effect of the number of configurations on the CCT, which will violate the shortest-coflow-first policy and increase the total CCT. To mitigate this issue, we design a prioritization strategy to further decrease the total CCT by taking the impact of ρ (the diameter) and τ (the maximum number of non-zero elements in demand matrixes rows or columns) into account jointly, where τ can reflect configuration delay.

2.2 Drawbacks of Head-of-line blocking

The existing non-preemptive online coflows scheduling algorithms in OCS, such as Reco [20], OMCO [19], cannot schedule the newly-arrived small coflows in time, which may cause Head-of-line (HOL) blocking and further violates the shortest first policy in online task scheduling.

To illustrate this drawback, we assume a simple scenario, where a long coflow A and a short coflow B arrive at moment 0 and 10 respectively. If preemption is prohibited, coflow B cannot start to transmit until coflow A has been completed. Hence the coflow

B is blocked by *A*, which means the completion time of *B* has to be postponed for extra waiting time and therefore increases the total CCT. By contrast, we transmit coflow *B* at once to preempt coflow *A*'s transmission, ensuring little blocking time. The preemptive scheduling strategy is presented in Sect. 5.


Waiting Queue
(δ – 2)
:
$\begin{bmatrix} 10 & 2 & 2 & 1 \\ 0 & 8 & 6 & 4 \\ 0 & 3 & 9 & 7 \\ 10 & 1 & 2 & 7 \end{bmatrix}$
C_1
(ρ = 20)
$\begin{bmatrix} 11 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 \\ 0 & 0 & 9 & 0 \\ 10 & 0 & 0 & 7 \end{bmatrix}$
C_2
(ρ = 21)
$\begin{bmatrix} 0 & 2 & 3 & 9 \\ 10 & 7 & 8 & 1 \\ 0 & 13 & 0 & 13 \\ 6 & 8 & 15 & 0 \end{bmatrix}$
C_3
(ρ = 30)
C_1
C_2
C_3
t = 37
t = 62
t = 98
C_2
C_1
C_3
t = 25
t = 62
t = 98

(a) Diameter-based scheduling (OMCO)

(b) Optical Scheduling

Fig. 1. A Motivating Example

3 System Model

3.1 Switch model

We assume that the data center network is a non-blocking optical circuit switch with *N* ingress ports and *N* egress ports [13, 15, 20, 22], where each port is linked to a group of hosts via Top of Rack switches. The data flows are buffered at sending-hosts and wait to be transmitted by switches. Any data from input port $p_{in}(0 \leq p_{in} \leq \text{N})$ to output port $p_{out}(0 \leq p_{out} \leq \text{N})$ can be transmitted only when a circuit has been established between p_{in} and p_{out}. OCS network requires any input port can not share the same circuit with each others and the same idea applies to output port, which means we can not transmit data from one input port to two or more output ports simultaneously. When we need to change the data transmission routes (i.e. the mapping between input ports and output ports), it is expected to reconfigure the optical circuit. The cost of circuit reconfiguration is a non-negligible fixed delay denoted as δ, and during this time period of δ, the two ports involved in circuit reconfiguration cannot transmit any data. Currently, the reconfiguration mechanism of OCS is divided into two types, All-Stop and Not-All-Stop. The All-Stop model, adopted by most previous works [12, 13, 15, 16], assumes that when one circuit is reconfigured, all other circuits are affected and torn down. In contrast, Sunflow [9] proposes Not-All-Stop model, which assumes that when a circuit is reconfigured, other unchanged circuits are unaffected and can continue to transmit. In this paper, we adopt the All-Stop circuit switch model.

3.2 Coflow Model

We denote the demand matrix of coflow i by DM_i, $i \in \{1, 2, 3 \ldots m\}$, which reflects both the amount of data in each flow and its transmission route [13, 20]. The element $A^i_{p_{in},p_{out}}$ in DM_i indicates that there is a flow of size $A^i_{p_{in},p_{out}}$ to be transmitted on the circuit from port p_{in} to port p_{out} for coflow i.

We represent the set of pending coflows as $\mathcal{C}$, where each coflow $C^{T_i}_i$, $i \in \{1, 2, 3 \ldots m\}$ is to be transmitted through the OCS after it arrives at time T_i. Similar to related works [2, 11, 23], We assume all flows of the same coflow arrive at the same time and the coflows' information (i.e., the demand matrixes) is a priori knowledge. We conclude two characteristics of DM_i as follows:

1. Diameter: The diameter of the DM_i is defined as the maximum of the sum of each row/column of the matrix, denoted as ρ_i.
2. Least reconfiguration times: For coflow i, it has at least τ_i reconfiguration times, where τ_i denotes the maximum number of non-zero elements per row or column in DM_i.

4 Problem Formulation

In the OCS system, we consider the problem of online scheduling the pending coflows to minimize the total CCT. Inspired by the existing work [12, 20], we can complete a coflow's transmission in OCS by configuring a series of optical circuits to realize non-blocking data-transmission of all flows in the coflow. That is, we can transform each coflow i 's demand matrix DM_i into the sum of l_i weighted permutation matrices as follows:

$$DM_i = \sum \alpha^i_j Q^i_j, \forall i \tag{1}$$

where Q^i_j is a $N \times N$ binary matrix encoding which ports are connected to each other in the circuit switch (i.e. circuit configuration) and α^i_j denotes how long the circuit switch should remain in this configuration (i.e. configuration duration) for the j-th matrix in coflow i.

In the scheduling strategy $\mathbb{P}$, the completion time of j-th configuration plan in coflow i is defined as $T\left(P^i_j\right)$, where P^i_j includes α^i_j and Q^i_j, and hence the CCT of coflow i is calculated as:

$$CCT_i = max_{j \in [1,l_i]} T\left(P^i_j\right) - T_i, \forall i \tag{2}$$

And we define the configuration plan ahead of the j-th one as $P^{i\prime}_{j\prime}$, which is described as below:

$$i^*, j^* = argmax_{i^{\prime},j^{\prime}} T\left(P^{i^{\prime}}_{j^{\prime}}\right) x^{i,i^{\prime}}_{j,j^{\prime}} \tag{3}$$

In addition to configuration strategy $\mathbb{P}$, another main decision we need to make is the order of all these configuration plans $P^i_j \in \mathbb{P}$. We introduce binary variables $x^{i,i\prime}_{j,j\prime}$ to indicate whether $P_{j'}^{i'}$ precedes P_j^i or not.

$$x_{j,j'}^{i,i'} = \begin{cases} 1 \; if \; P_{j'}^{i'} \; precedes \; P_j^i \\ 0 \; otherwise \end{cases} \tag{4}$$

And the reconfiguration delay between P_{j*}^{i*} and P_j^i is denoted as

$$\delta_{j,j*}^{i,i*} = \begin{cases} 0 \; if \; Q_{j*}^{i*} = Q_j^i \\ \delta \; otherwise \end{cases} \tag{5}$$

which means that if the circuit configuration of P_{j*}^{i*} and P_j^i are the same, the reconfiguration overhead can be dismissed.

The completion time of j-th configuration plan in coflow i is calculated as:

$$T\left(P_j^i\right) = T\left(P_{j^*}^{i^*}\right) + \delta_{j,j^*}^{i,i^*} + \alpha_j^i \tag{6}$$

As a result, we formulate our problem as below:

$$\min_{\mathbb{P}} \sum_{i=1}^{m} CCT_i$$

s.t. (1), (2), (3), (4), (5), (6)

When we cannot preempt any configuration plan P_j^i, our problem reduces to the classic NP-hard problem of open-shop scheduling [9], which establishes our problem's NP-hardness.

5 Algorithm Design

We present in this section a heuristic algorithm composed of two stages of pre-processing coflows' demand matrix, which generates a series of configuration plans, followed by one-by-one online configuration plan ordering. Coflow pre-processing transforms each coflow's demand matrix to a series of weighted permutation matrices, where the permutation matrix represents a configuration of OCS and the weight (coefficient) means the corresponding duration. Given every OCS configuration and the corresponding duration (we consider them together as a configuration plan), we order these plans by consecutively prioritizing coflows and the configuration plans decomposed from the coflow of top priority. In online settings, our scheduling algorithm will dynamically update each coflow's priority when OCS configuration is changed and then re-schedule a configuration plan of minimum duration from the most prioritized coflow over and over again. The procedure of coflow pre-processing and scheduling coflows with dynamic priority will be detailed as follows.

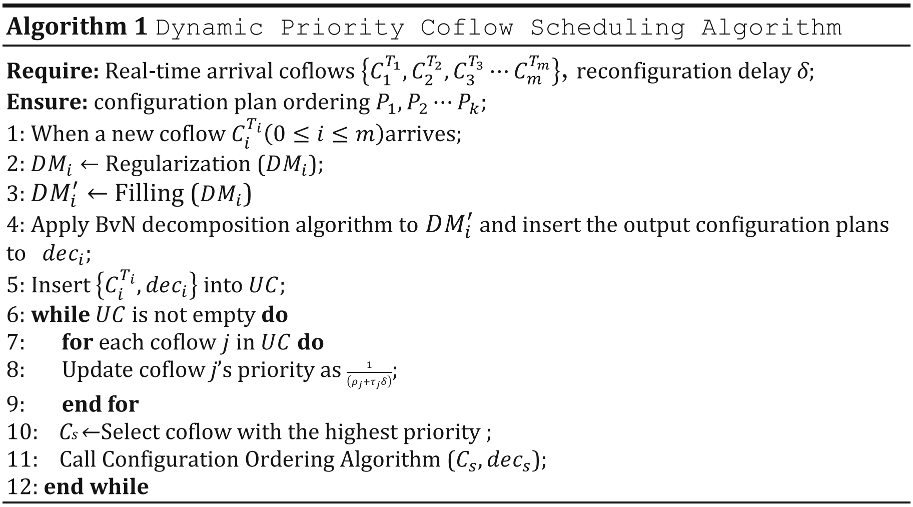
5.1 Coflow Pre-processing

In order to transmit coflows in the optical circuit switch, following the previous work [12, 19, 20], we apply the Birkoff-von Neumann decomposition (BvN) to decompose the DM_i into l_i weighted permutation matrices. Since The BvN only accepts bistochastic matrix as the input, we convert the demand matrix to a bistochastic matrix $DM_{i'}$, whose each row and column sums to the same value, in three steps: regularization, filling and Birkoff-von Neumann decomposition (BvN).

Regularization: Since optical circuit reconfiguration requires a non-negligible delay, the performance of coflow scheduling algorithm declines sharply under frequent reconfigurations. To avoid such problem, we regularize each entry $A_{p_{in},p_{out}} \in DM_i$ to $\lceil \frac{A_{p_{in},p_{out}}}{\delta} \rceil \cdot \delta$, which can greatly reduce the reconfiguration times at the cost of slightly more transmission time.

Filling: We go through each elements in regularized matrix DM_i and increase its value until all row/column sums are ρ_i, which ensures the least link usage waste in OCS. In this process, we make every effort to avoid increasing the number of zero elements, as otherwise more reconfigurations will occur. After applying the above two operations, we obtain a bistochastic matrix $DM_{i'}$ and then use BvN algorithm to transform $DM_{i'}$ to a series of configuration plans.

Birkoff-von Neumann Decomposition (BvN): Given a doubly stochastic matrix $DM_{i'}$, we decompose it into permutation matrices with specific coefficients (we consider them together as configuration plans), which satisfies the port constraints. We can map BvN decomposition of a coflow demand matrix into OCS, where each permutation matrix is the OCS circuit reconfiguration status and coefficient of permutation matrix is the circuit duration time.



Algorithm 1 Dynamic Priority Coflow Scheduling Algorithm

```
Require: Real-time arrival coflows $\{C_1^{T_1}, C_2^{T_2}, C_3^{T_3} \cdots C_m^{T_m}\}$, reconfiguration delay $\delta$;
Ensure: configuration plan ordering $P_1, P_2 \cdots P_k$;
When a new coflow $C_i^{T_i} (0 \le i \le m)$arrives;
$DM_i \leftarrow$ Regularization $(DM_i)$;
$DM_i' \leftarrow$ Filling $(DM_i)$
Apply BvN decomposition algorithm to $DM_i'$ and insert the output configuration plans to $dec_i$;
Insert $\{C_i^{T_i}, dec_i\}$ into $UC$;
while $UC$ is not empty do
    for each coflow $j$ in $UC$ do
    Update coflow $j$'s priority as $\frac{1}{(\rho_j + \tau_j \delta)}$;
    end for
    $C_s \leftarrow$Select coflow with the highest priority ;
    Call Configuration Ordering Algorithm $(C_s, dec_s)$;
end while
```

5.2 Coflow Scheduling on Dynamic Priority

Given a number of configuration plans, we now discuss how to arrange them and schedule one-by-one.

In the light of the problem we discussed in Sect. 2.1, the diameter ρ_i and the number of permutation matrices τ_i, which reflect the transmission time and reconfiguration delay of coflow $C_i^{T_i}$ respectively, are both dominating factors of high-performance scheduling. Therefore, we measure the coflow's priority by $\frac{1}{(\rho_i+\tau_i\delta)}$, where $\rho_i+\tau_i\delta$ is clearly the lower-bound of coflow completion time. The larger the $\frac{1}{(\rho_i+\tau_i\delta)}$ value is, the higher priority coflows have. We precede the configuration plans of the shortest duration from the coflow of the highest priority with others. Considering that newly arrived coflows will be blocked in online non-preemptive scheduling, we allow the small coflows preempt the large coflows from relentlessly transmitting data, and thus we need to dynamically update the coflow's priority when this configuration plan has been completed. The aforementioned procedures will iterate until all coflows complete.

According to the above idea, we propose a Dynamic Priority Coflow Scheduling Algorithm, which ensures no additional reconfiguration delays by allowing preemptions. The details are elaborated in Algorithm 1.

When a new coflow $C_i^{T_i}$ arrives, with dec_i representing the set of uncompleted configuration plans from $C_i^{T_i}$, we pre-process the coflow i's demand matrix DM_i to a bistochastic matrix $DM_{i'}$ through regularization and filling and we insert all the plans to dec_i(line 1 to line 4). Let UC be the set of uncompleted coflows in waiting queue (line 5), our algorithm repeats iteration over and over again until UC is empty (line 6). In each iteration, we name the coflow in UC with the highest priority as the candidate coflow C_s and call algorithm 2 to schedule the configuration plans in dec_s (line 11). In Configuration Ordering Algorithm, we sort the series of configuration plans $\left\{P_s^1, P_s^2, P_s^3 \cdots P_s^{l_s}\right\}$ in the ascending order of duration and determine the plan with minimum duration P_{min} is the first one to be scheduled (line 1). Note that we can select suitable flows from UC to fill the under-load circuits to improve the bandwidth utilization, which is called Back-Filling [19] in line 2. Last, we update UC and dec_s when P_{min} has completed and re-call Algorithm 1 to schedule coflows.

```
Algorithm 2 Configuration Ordering Algorithm($C_s, dec_s$)
Require: $C_s$, $dec_s$;
Ensure: A min-duration permutation plan of $C_s$;
1: $P_{min}$ ←the min-duration permutation plan in $dec_s$;
2: Back-Filling: Fill the suitable data of the waiting coflows in $UC$ to the under-load circuit $i \rightarrow j$ in $P_{min}$;
3: $C_s \leftarrow C_s - P_{min}$;
4: $dec_s \leftarrow dec_s - P_{min}$;
5: if $C_s$ is uncompleted then
6:     Put $\{C_s, dec_s\}$ back to $UC$;
7: end if
8: return $P_{min}$;
```

6 Experimental Evaluations

In this section, we use the traces of Facebook [1] to test the performance of the proposed method and provide simulation results and detailed performance analysis.

6.1 Simulation Settings

For the simulation environment, we create an online coflow scheduling simulator with Python 3.7.

Workload: Our workload is generated based on Facebook traces [1], collected from a 3000-machine, 150-rack MapReduce cluster at Facebook. The Facebook trajectories are widely used in simulation [4, 10, 21], which contains 526 coflows that are scaled down to a 150-port fabric with exact inter-arrival times. For each coflow, the Facebook trace contains sender machines, receiver machines, and transmitting bytes at the receiver level, not the flow level. Thus we partition the bytes in each receiver to each sender pseudo-uniformly to generate flows. We randomly selected P machines from the trace as servers. The arrival time of each job obeys a Poisson distribution $P(\lambda)$.

Evaluation Metric: Our metric is the Normalized CCT of a scheme compared with our method.

- The Normalized CCT of Algorithm A is defined as

$$NormalizedCCT = \frac{the\ CCT\ of\ Algorithm\ A}{the\ CCT\ of\ our\ method}$$

Intuitively, our method is faster if the *Normalized CCT* is greater than one.

Baseline solutions: We compare the performances of our method with the following baselines in minimizing the total CCT.

1) *First In First Out (FIFO):* prioritizes coflows based on their arrival time.
2) *OMCO* [19]*:* prioritizes coflows based on the *traffic threshold*, which represents the diameter of their demand matrices.

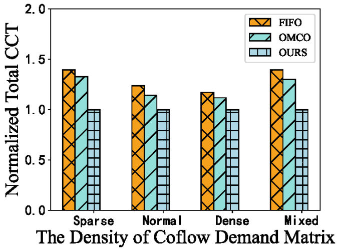


Fig. 2. Norm. CCT in Different Schemes

6.2 Simulation Results

In our simulations, we consider the OCS with 8 ports (i.e.,$P = 8$), each link in the OCS with a bandwidth of 100 Gbps. The reconfiguration delay, denoted δ, varies from 20 μs to 100 μs, and the default reconfiguration delay is 20 μs. In order to analyze the characteristics of the demand matrix, its sparsity was investigated. The sparsity of a matrix is determined by the proportion of non-zero elements in the matrix, expressed as a density value from 0 to 1. In this study, we classify coflows into three types based on the density of the demand matrix: sparse, normal and dense. A demand matrix is considered sparse when its density is less than or equal to 0.3. When a matrix has a density in the range of 0.3 to 0.6, it is classified as normal. Finally, when a matrix has a density greater than or equal to 0.6, it is labeled as dense.

Figure 2 Norm. CCT in Different Schemes illustrates the performance of our proposed method and different schedulers in terms of minimizing the total CCT at different density levels while keeping the reconfiguration time fixed at 20 μs. To establish a benchmark, we normalize the CCT of our method and compare it with the performance of various other schemes. Compared to FIFO and OMCO [19], which schedule coflows in OCS in a non-preemptive manner, our method allows for preemption and schedule coflows based on extra attributes of the demand matrices, thereby reducing the total CCT. Specifically, OMCO requires 1.33×, 1.14×, 1.12× and 1.30× more time than our method to schedule the demand matrices with sparse, normal, dense and mixed coflows, respectively. FIFO requires 1.40×, 1.24×, 1.17× and 1.38× more time than our method to schedule the demand matrices with sparse, normal, dense and mixed coflows, respectively.

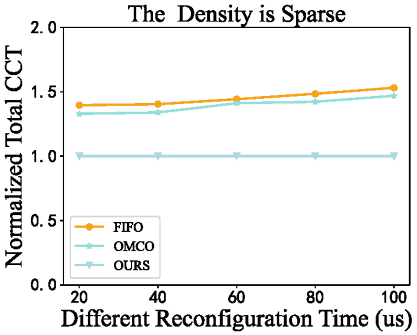


Fig. 3. Norm. CCT in Sparse Density

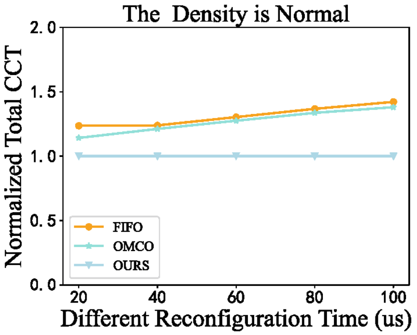


Fig. 4. Norm. CCT in Normal Density

The variation of δ, which is determined by the hardware characteristics of the OCS, plays a key role in the total CCT. Analyzing Figs. 3, 4, 5 and 6, we find that the total CCT of all schemes decreases as δ decreases, which is reasonable. Furthermore, it is noteworthy that the performance gap between these schemes widens as δ increases. The reason behind this observation is that as δ increases, reconfiguration delay becomes more dominant in determining the total CCT. In this case, the advantages of our proposed approach become more prominent. Our approach takes into account the maximum number

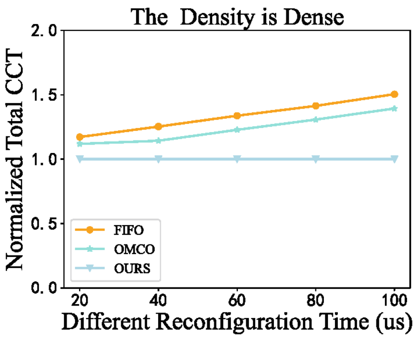


Fig. 5. Norm. CCT in Dense Density

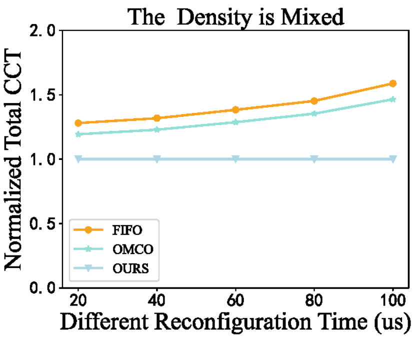


Fig. 6. Norm. CCT in Mixed Density

of non-zero elements in the rows or columns of the demand matrix when generating the coflow priorities, allowing for more efficient scheduling and reducing the overall CCT.

7 Conclusion

This paper studies the problem of scheduling online coflows in Optical Circuit Switched (OCS) datacenter networks. We propose a Dynamic-Priority Coflow Scheduling Algorithm, which prioritizes coflows by coflow transmission time and OCS configuration delay combined, and generates the configuration plans of the coflows following their priorities by employing the BvN decompostion algorithm. In the future, we will study further how to schedule coflows in OCS-EPS-Hybrid environments.

Acknowledgement. This work is supported by Macao Polytechnic University Research Grant RP/FCA-13/2022 and Key-Area Research and Development Plan of Guangdong Province #2020B010164003. The corresponding author is Hong Shen.

References

1. FaceBookTrace. https://github.com/coflow/coflow-benchmark (2019)
2. Chen, L., Cui, W., Li, B., Li, B.: Optimizing coflow completion times with utility max-min fairness. In: INFOCOM. IEEE (2016)
3. Chowdhury, M., Stoica, I.: Coflow: a networking abstraction for cluster applications. In: HotNets-XI. ACM (2012)
4. Chowdhury, M., Stoica, I.: Efficient Coflow scheduling without prior knowledge. In: SIGCOMM. ACM (2015)
5. Chowdhury, M., Zaharia, M., Ma, J., Jordan, M. I., Stoica, I.: Managing data transfers in computer clusters with orchestra. In: SIGCOMM. ACM (2011)
6. Chowdhury, M., Zhong, Y., Stoica, I.: SIGCOMM. ACM (2014)
7. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM. **51**, 107–113 (2008)
8. Gopal, I., Wong, C.: Minimizing the number of switchings in an SS/TDMA system. IEEE Trans. Commun. **33**(6), 497–501 (1985). https://doi.org/10.1109/TCOM.1985.1096336

9. Huang, X.S., Sun, X.S., Ng, T.S.E.: Sunflow: efficient optical circuit scheduling for Coflows. In: CoNEXT. ACM (2016)
10. Li, Y., et al.: Efficient online coflow routing and scheduling. In: MobiHoc. ACM (2016)
11. Li, Z., Zhang, Y., Li, D., Chen, K., Peng, Y.: OPTAS: decentralized flow monitoring and scheduling for tiny tasks. In: INFOCOM. IEEE (2016)
12. Liu, H., et al.: Scheduling techniques for hybrid circuit/packet networks. In: CoNEXT. ACM (2015)
13. Porter, G., et al.: Integrating microsecond circuit switching into the data center. In: SIGCOMM. ACM (2013)
14. Tang, Y., Yuan, T., Liu, B., Xiao, C.: Effective *-flow schedule for optical circuit switching based data center networks: a comprehensive survey. Comput. Netw. **197**, 108321 (2021). https://doi.org/10.1016/j.comnet.2021.108321
15. Wang, C.H., Javidi, T., Porter, G.: End-to-end scheduling for all-optical data centers. In: INFOCOM. IEEE (2015)
16. Wang, G., et al.: c-Through: part-time optics in data centers. In: SIGCOMM. ACM (2010)
17. Wang, S., Wang, S., Huo, R., Huang, T., Liu, J., Liu, Y.: DeepAalo: auto-adjusting demotion thresholds for information-agnostic coflow scheduling. In: INFOCOM. IEEE (2020)
18. Wang, S., Zhang, J., Huang, T., Pan, T., Liu, J., Liu, Y.: Leveraging multiple coflow attributes for information-agnostic coflow scheduling. In: ICC. IEEE (2017)
19. Xu, C., Tan, H., Hou, J., Zhang, C., Li, X.Y.: OMCO: online multiple coflow scheduling in optical circuit switch. In: ICC. IEEE (2018)
20. Zhang, C., Tan, H., Xu, C., Li, X.Y., Tang, S., Li, Y.: Reco: efficient regularization-based coflow scheduling in optical circuit switches. In: ICDCS. IEEE (2019)
21. Zhang, H., Chen, L., Yi, B., Chen, K., Chowdhury, M., Geng, Y.: CODA: toward automatically identifying and scheduling coflows in the dark. In: SIGCOMM. ACM (2016)
22. Zhang, T., Ren, F., Bao, J., Shu, R., Cheng, W.: Minimizing coflow completion time in optical circuit switched networks. IEEE Trans. Parallel Distrib. Syst. **32**(2), 457–469 (2021). https://doi.org/10.1109/TPDS.2020.3025145
23. Zhao, Y., et al.: Rapier: integrating routing and scheduling for coflow-aware data center networks. In: INFOCOM. IEEE (2015)

Deep Reinforcement Learning Based Multi-WiFi Offloading of UAV Traffic

Zhiyong Liu[1](✉) and Hong Shen[2]

[1] School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China
liuzhy88@mail2.sysu.edu.cn

[2] Faculty of Applied Sciences, Macao Polytechnic University, Macao, China

Abstract. As the growing network deployment of Unmanned Aerial Vicheles (UAVs), traffic offloading has been widely used to mitigate UAV's problem of limited bandwidth in communications due to limited battery capacity. Existing work on traffic offloading has focused on reducing the average delay of the system without considering the fairness issue, and assumed that data transmission follows line-of-sight propagation which contradicts with the realistic situations in both urban and suburban areas. Achieving both fairness and system efficiency with non-line-of-sight user-UAV communication requires to solve a complex non-convex optimization problem. This paper proposes an effective algorithm (NAPPO) for joint UAV navigation and user traffic allocation by applying deep reinforcement learning (DRL). Our NAPPO applies DRL to collect user information (position, data rate and traffic demand) and dynamically adjusts the UAV position and traffic allocation ratio to minimize the maximum delay and hence improve the fairness (i.e., variation in delay between users). We show that our proposed approach of minimizing maximum delay is more effective than minimizing average delay for achieving fairness while preserving the total delay at a reasonable level. The results of the simulation experiments show that NAPPO achieves an impressive performance on the maximum delay, i.e., 49. 82% better than the heuristic algorithm and only 0.1536s worse than the optimal solution.

Keywords: UAV · Traffic offload · Deep reinforcement learning · Cellular networks

1 Introduction

Recent years have seen an unprecedented development in the deployment of Unmanned Aerial Vehicles (UAVs) as base stations to provide wireless communications in emergency cases such as earthquakes and forest fires, and large events such as music festivals and sports events. With easy and fast communication deployment, low cost, and adaptability, UAVs can make up for the shortcomings of traditional base stations in unexpected situations and remote areas to provide additional bandwidth. To resolve traffic congestions in bandwidth-limited UAV communication, offloading of traffic in UAV base station wireless networks to WiFi networks with small coverage but high bandwidth and capacity, namely traffic offloading, becomes a preferred option to improve communication

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 227–238, 2024.
https://doi.org/10.1007/978-981-99-8211-0_21

quality. We investigate the problem of minimizing the maximum latency for multi-WiFi offloading of UAV base station user traffic over non-line-of-sight (nLos) links. This is motivated by the observation that reducing the maximum latency of the network can effectively reduce the latency difference between users and hence enhance fairness while keeping an acceptable overall network latency, and adoption of nLos links rather than the more restricted Los as assumed in the literature is closer to practical application scenarios. For nLos links, the signal-to-noise ratio (SNR) between UAV and user is not any more inversely proportional to the square of their distance as in Los, making the min-max problem under this constraint a non-convex optimization problem. To solve this problem, we propose to apply Deep Reinforcement Learning (DRL) technique that relies on dynamic information about the real environment to make real-time decisions. Our contributions are as follows:

- We define the maximum delay minimization problem as that of joint UAV navigation and user traffic allocation on nLos links and formulate the problem as a Markov Decision Process.
- We propose a joint UAV navigation and traffic allocation algorithm (NAPPO) applying Deep Reinforcement Learning with Proximal Policy Optimization (PPO [1]) for gradient updating, which provides an efficient solution to the maximum delay minimization problem.
- We conduct extensive simulation experiments and the experiment results demonstrate that NAPPO has a lower delay than the heuristic algorithm of minimizing average delay.

2 Related Work

Shanza Shakoor et al. [2] investigated the maximum user access rate for UAVs as cellular base stations. Adjustment of UAV location by the k-means clustering algorithm to maximize user access rate by cyclic search. Cheng Zhan et al. [3] studied the problem of maximizing the number of service nodes in a multi-UAV IoT edge computing scenario. The UAV trajectory, offload ratio, and resource allocation are optimized by the SCA method, respectively. The service nodes are maximized by cyclically updating the variables. Yong Zeng et al. [4] studied the navigation problem of maximizing the communication quality of UAV-connected ground networks, and proposed a DDQN-based navigation algorithm to adapt to complex channels by discretizing the action space. Xuanheng Li et al. [5] studied the UAV data acquisition and transmission problem for heterogeneous networks, and maximize the spectrum efficiency of the UAV transmission chain by controlling the UAV trajectory, data acquisition and transmission ratio, frequency band selection, and transmission power through the DQN method. Muntadher A. Ali et al. [6] investigated the problem of traffic offloading from UAVs as cellular base stations by optimizing the traffic allocation ratio, UAV location, and band allocation ratio to minimize the average delay using the block coordinate descent method. The assumption that the transmission channel is a line-of-sight link is the basis of their research. However, due to the demand for WiFi, the traffic offload scenarios are often densely built cities, and it is not reasonable to assume that the channel is a line-of-sight channel in such a complex environment. As an improvement to this, our study uses a non-line-of-sight probabilistic model to [12] describe the communication channel. In addition, since their optimization objective is the average latency, this may result in a large variation in latency between different users. And we set the optimization objective to minimize the maximum latency. If the maximum latency is small enough, the latency for each user will not be high. This also ensures that the latency difference between users is low.

3 Network model

We are given an area containing one UAV as the cellular base station, a set of stationary Access Points (APs) for WiFi networks, a set of Cellular Subscribers (CSs) accessing network through the UAV, and a set of WiFi Subscribers (WSs) accessing network through APs, as shown in Fig. 1. To minimize the communication delay of all the CS's, we need to decide (allocate) an appropriate portion of CS' traffic to go through a WiFi via a reachable AP to share WiFi bandwidth with the WS's, and that to go through the UAV according to UAV's position. For fairness, our adopt the approach of minimizing the maximum delay. For practical considerations, we assume that the links between UAV and CSs are nLos. So our problem is how to minimize the maximum delay of CS's by joint UAV navigation and CS traffic allocation.

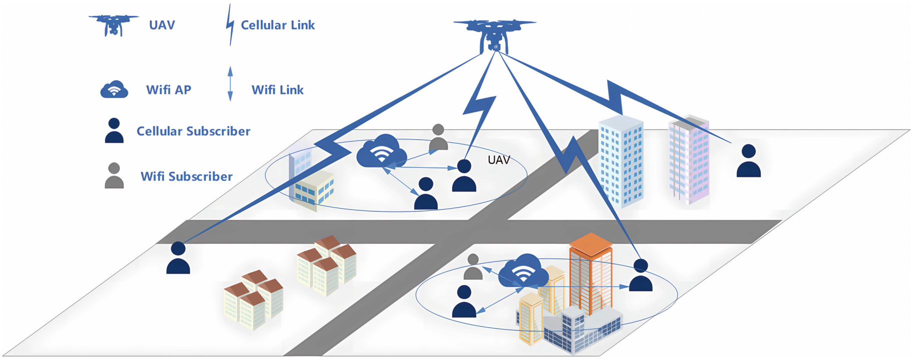


Fig. 1. WiFi AP offloads UAV traffic

More specifically, the UAV serves as the base station to provide cellular network services to N_{CS} CS uniformly distributed in a square area with the side length of $2r$, CS_i subscribers with horizontal positions of $q_i(i = \{1, 2 \ldots .N_{CS}\})$. The UAV's horizontal position is w, moving horizontally to find the best position to provide network services and maintaining height at H, avoiding collision with buildings as the UAV moves. At the same time, N_{AP} APs provide WiFi services to WSs and CS within their coverage area. The coverage radius of APs are all r_{AP}, and the coverage areas do not overlap. The number of WSs served by each AP is fixed and denoted as $N_i^{WS}(i = \{1, 2, \ldots, N_{AP}\})$. All WSs are within the coverage area of some APs and can access WiFi network only through AP, and all CS are connected to cellular network through UAV. The N CSs are within the coverage area of an AP and can access the WiFi network through the AP and offload some of its traffic demand to the WiFi network. This reduces the load on the UAV and lowers the overall system latency. The traffic demand of CS_i is λ_i^{CS}. If CS_i is within the AP coverage and its traffic offload is proportional to $\mu_i(\mu_i \in [0, 1])$, then its traffic demand transmitted through AP is $\lambda_i^{AP} = \mu_i\lambda_i^{CS}$, and its traffic demand transmitted through UAV is $\lambda_i^{UAV} = (1 - \mu_i)\lambda_i^{CS}$. If CS_i is not in the AP coverage, then its traffic demand through AP is $\lambda_i^{AP} = 0$ and its traffic demand through UAV is $\lambda_i^{UAV} = \lambda_i^{CS}$. χ denotes the mapping relationship from CS to AP. If CS_i is within the coverage of AP_j, then $\chi_{i,j} = 1$, otherwise $\chi_{i,j} = 0(i \in \{1, 2 \ldots, N\}, j \in \{1, 2 \ldots, N_{AP}\})$, and $\chi_{i,j}$ satisfy $\sum_{j=1}^{N_{AP}} \chi_{i,j} = 1\left(\chi_{i,j} \in \{0, 1\}\right)$. The number of users access to AP_j is the sum of the number of WSs and the number of CS accessed, i.e., $N_j^{AP} = N_j^{WS} + \sum_{i=1}^{N} \chi_{i,j}\epsilon(\mu_i)$. If $x \in [0, \varepsilon]$ then $\epsilon(x) = 0$, otherwise $\epsilon(x) = 1$. Here ε is a sufficiently small number and $\epsilon(x)$ indicates that if the allocation ratio μ_i is less than ε, the CS is regarded as not accessing WiFi. The total throughput of AP_j is denoted as Θ_j, so the transfer rate of CS_i within its accessed AP_j is $R_i^{AP} = \frac{\Theta_j}{N_j^{AP}}$. The authorized bandwidth of UAV is B. The spectrum is allocated to each CS according to its actual traffic demand transmitted by UAV(i.e., the spectrum allocation ratio of CS_i is $\varphi_i = \frac{\lambda_i^{UAV}}{\sum_{i=1}^{N_{CS}} \lambda_i^{UAV}}$. The channel between UAV and CS is a nLos channel [13], which results in a channel gain g_i that depends on the UAV position w and the angle between the transmission channel and the ground in a complex way, which then leads to the failure of traditional convex optimization methods. According to Shannon theory, the spectral efficiency of $CS_i is S_i = \log_2\left(1 + \frac{Pg_i}{N0}\right)$, where P is the UAV transmit power, and N_0 is the Gaussian noise power. So the rate of CS_i in transmission through the UAV is $R_i^{UAV} = \varphi_i BS_i = \frac{B\lambda_i^{UAV}\log_2\left(1+\frac{Pg_i}{N0}\right)}{\sum_{i=1}^{N_{CS}} \lambda_i^{UAV}}$. The service arrival time of each user is exponentially distributed and independent, and the service provided by UAV and AP to users obeys the M/M/1 queuing model. The delay of CS_i on the UAV is $\delta_i^{UAV} = \frac{1}{\left(R_i^{UAV} - \lambda_i^{UAV}\right)^+}$. If CS_i is within the coverage of AP_j, the delay of CS_i on AP [7, 8] is $\delta_i^{AP} = \frac{1+0.5R_j^{AP}\lambda_i^{AP}v_i}{\left(R_j^{AP} - \lambda_i^{AP}\right)^+}$. WiFi services are based on probabilistic contention, and channels are acquired by performing carrier sensing, where each CS_i has a quiescent period V_i, whose mean and variance depend on the interference intensity of other competing nodes. And $v_i = E\left[V_i^2\right]$ is the expectation of the quiescent period V_i for WiFi contention [8]. If CS_i is not in the AP coverage, then $\delta_i^{AP} = 0$. Therefore, the average delay of CS_i is $\delta_i = \frac{max\left(\lambda_i^{UAV}\delta_i^{UAV}, \lambda_i^{AP}\delta_i^{AP}\right)}{\lambda_i^{CS}}$. The maximum delay in all CSs is $\delta^{max} = max\left(\delta_1, \delta_2, \ldots, \delta_{N_{CS}}\right)$.

4 Problem Formulation

We give a formulation of the maximum delay minimization problem for multi-WiFi offloading of UAV base station user traffic over a non-line-of-sight link:

$$\min_{w,\mu_i} \delta^{max} \tag{1}$$

$$\mathrm{s.t.}\lambda_{\mathrm{i}}^{\mathrm{UAV}} \le R_{\mathrm{i}}^{\mathrm{UAV}}, \forall i \in N_{\mathrm{CS}}$$

$$\lambda_i^{AP} \le R^{AP}, \forall i \in N_c$$

Here, **constraint (a)** indicates that the transmission rate of CS on the UAV should be greater than the traffic demand transmitted by the UAV, and **constraint (b)** indicates that the transmission rate of CS on the AP should be greater than the traffic demand transmitted by the AP. The non-line-of-sight channel causes the channel gain g_i to depend on the position of the UAV wand the angle between the transmission channel and the ground in a complex way, making **Formulation (1)** non-convex which invalidates the traditional convex optimization techniques. Therefore, we propose to apply Deep Reinforcement Learning to solve the problem.

5 NAPPO Algorithm

In this section, we first build a Markov Decision Process (MDP) model of the problem to lend it for deployment of Deep Reinforcement Learning (DRL), and then propose a PPO-based DRL algorithm for solving the problem.

5.1 MDP Model Design

We model the problem as an MDP, which can be defined by defining the state, action, and reward as follows:

State: To make decisions when the channel model is nLos, define the state as:

$$s_t = \{\Delta w, R^{UAV}, R^{AP}, \lambda^{UAV}, \lambda^{AP}\}$$

where Δw is the position of the CS relative to the UAV, which is useful for determining the UAV's position. R^{UAV} denotes the transmission rate of CS to UAV, R^{AP} denotes the transimission rate of CS to WiFi. λ^{UAV} denotes the traffic demand of CS to UAV and λ^{AP} denotes the traffic demand of CS to WiFi. The real-time transmission rate and traffic demand can help the agent to decide the traffic allocation ratio correctly.

Action: The action controls the UAV movement and traffic allocation:

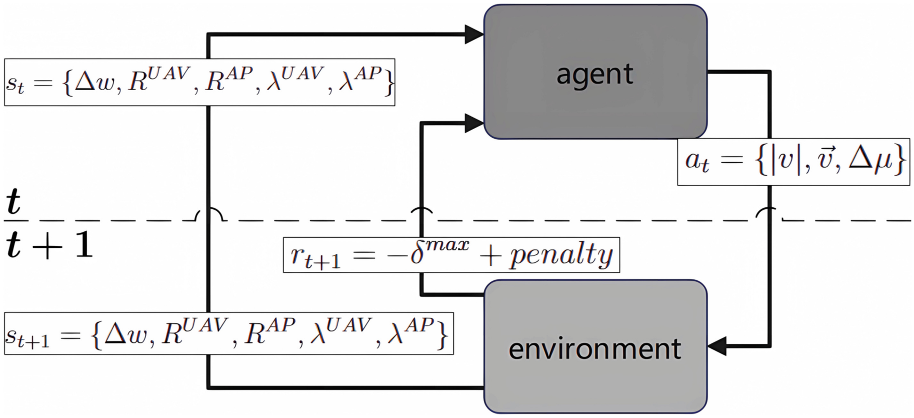
$$a_t = \{|v|, \overrightarrow{v}, \Delta\mu\}$$

where $|v|, \overrightarrow{v}$, respectively denote the velocity magnitude and direction of the UAV. The $\Delta\mu$ denotes the allocated proportional increment of CS within the AP coverage. For CS_i, the traffic allocation ratio of the next time slot is $\mu_{t+1}^i = \mu_t^i + \Delta\mu^i$.

Reward: The reward is for delay reduction and penalty for violating constraints, i.e.,

$$r_t = -\delta^{max} - \alpha\left(\sum\nolimits_i^{N_{CS}} I\left(R_i^{UAV} - \lambda_i^{UAV}\right) + \sum\nolimits_j^{N} I\left(R_j^{AP} - \lambda_j^{AP}\right)\right)$$

where I(x) denotes the number of constraint violations, if $x \ge 0$ then $I(x) = 1$, otherwise $I(x) = 0$. α is the penalty factor that is much larger than 1 and it ensures that the transfer



$s_t = \{\Delta w, R^{UAV}, R^{AP}, \lambda^{UAV}, \lambda^{AP}\}$
agent
$a_t = \{|v|, \vec{v}, \Delta\mu\}$
t
$t+1$
$r_{t+1} = -\delta^{max} + penalty$
$s_{t+1} = \{\Delta w, R^{UAV}, R^{AP}, \lambda^{UAV}, \lambda^{AP}\}$
environment

Fig. 2. MDP model

rate is higher than the traffic demand before the action is chosen. Notice that we transform (a) and (b) into penalties in the reward, which can satisfy the constraint while minimizing the maximum delay.

Our MDP model is shown in Fig. 2. In slot t, the agent first gets environment state information s_t which includes Δw, R^{UAV}, R^{AP}, λ^{UAV} and λ^{AP} in slot t. The best action a_t which includes $|v|$, $\overrightarrow{v}$ and $\Delta\mu$ is choosen based on state information s_t. Then the UAV moves to next position $w_{t+1} = w_t + |v| \times \overrightarrow{v}$, and the traffic allocation ratio of CS_i in the next slot $t+1$ changes into $\mu_{t+1}^i = \mu_t^i + \Delta\mu^i$. After executing the action a_t, the agent receives the reward reflecting the maximum delay and the number of constraint violations which will be used for updating policy, i.e.,

$$r_t = -\delta^{max} - \alpha\left(\sum_i^{N_{CS}} I\left(R_i^{UAV} - \lambda_i^{UAV}\right) + \sum_j^{N} I\left(R_j^{AP} - \lambda_j^{AP}\right)\right).$$

In the next slot $t+1$, the agent repeats the above operations. Eventually, the agent gets the best policy which can choose the best action to minimize the maximum delay while satisfying constraints.

5.2 Algorithm Design

DRL solves the MDP by finding the optimal policy π. The policy $\pi(a_t|s_t) : S \times A \rightarrow [0, 1]$ gives the probability that the agent chooses a_t in the state s_t, and the optimal policy π^* makes the value function $V_\pi(s) = E_\pi[G_t|s_t = s]$ maximum, i.e., $\pi^* = \underset{\pi}{argmax} V_\pi(s)$. In DRL, the value function-based algorithm outputs action directly, which is only suitable for discrete action space. In contrast, the policy gradient algorithm outputs the action distribution and samples the actual action in this action distribution, which is more suitable for the high-dimensional continuous action space.

To increase the probability of actions leading to greater final return and decrease that of smaller final return, the policy gradient algorithm maximizes the payoff function $J(\pi_\theta) = E_{\pi_\theta \sim \tau}[G(\tau)]$, where $G(\tau) = \sum_{t=0}^{T} \gamma r_t$ is the total return, $\gamma \in [0, 1]$ is the discount factor, and τ is the trajectory with total time step T based on the policy π_θ. The parameters θ of the neural network are optimized by gradient ascent, i.e.,

$$\theta \leftarrow \theta + \alpha \nabla J(\pi_\theta)$$

where $\nabla J(\pi_\theta) = E_{\pi \sim \tau}\left[\sum_{t=0}^{T} \nabla_\theta log\, \pi_\theta(a_t|s_t) G(\tau)\right]$.

The policy gradient algorithm is strongly influenced by the step size α when executing gradient ascents. A too-small or too-large α will respectively lead to slow convergence and excessive updates, which may ultimately make the resulting policy unsuitable. Classical algorithms such as Natural Policy Gradient (NPG) [9] and Trusted Domain Policy Optimization (TRPO) [10] restrict the update step size by the Kullback-Leibler divergence constraint. However, the high computational complexity of Kullback-Leibler divergence leads to slow convergence Proximal Policy Optimization (PPO)[1] discards the Kullback-Leibler Divergence constraint and redefines the alternative advantage to limit the update magnitude to the expected, and uses a first-order stochastic gradient ascent optimizer to improve the convergence speed. Because the action space in our problem is high-dimensional and continuous, which requires rapid decision-making based on state information, we deploy PPO for gradient updating about the proportion of UAV localization and traffic allocation.

Algorithm 1 PPO-based navigation and allocation algorithm(NAPPO)

```
Initial policy parameter $\theta_0$,discount factor $\gamma$, GAE parameter $\lambda$,clipping threshold $\epsilon$
Initial Replay buffer, orthogonal initial Actor and Critic network
for k=0,1,2… do
  while Replay buffer not full do
    Get observation $s$ from environment
    $a = \pi_k(a|s) = \pi(a|s;\theta_k)$
    Normalization $s,a,r$
    Store $< s,a,r,s' >$ in replay buffer
  end while
  Recursion compute all $A(s_t,a)$ by
            $A(s_t,a) = r + \gamma V_\pi(s_{t+1}) - V_\pi(s_t) + \gamma\lambda A(s_{t+1},a)$
  Normalization $A(s_t,a)$
  Compute policy update
            $\theta_{k+1} = argmax_{\theta \mathcal{L}_{\theta_k}^{\mathcal{CLIP}}}(\theta)$
  by taking K steps of minibatch SGD(via Adam), where
    $\mathcal{L}_{\theta_k}^{\mathcal{CLIP}}(\theta) = E_{\tau \sim \pi_k}\left[\sum_{t=0}^{T}\left[min\left(r_t(\theta)\widehat{A_t^{\pi_k}}, clip(r_t(\theta),1-\epsilon,1+\epsilon)\widehat{A_t^{\pi_k}}\right)\right]\right]$
end for
```

The PPO algorithm redefines the surrogate advantage by the *clip* function, i.e.,

$$\mathcal{L}_{\pi_\theta}^{\mathcal{CLIP}} = E_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T}\left[min\left(\rho_t\left(\pi_\theta, \pi_{\theta_k}\right)A_t^{\pi_\theta}, clip\left(\rho_t\left(\pi_\theta, \pi_{\theta_k}\right), 1-\epsilon, 1+\epsilon\right)A_t^{\pi_\theta}\right)\right]\right]$$

where the advantage function is defined as $A^{\pi_\theta}(s_t, a_t) = E(Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s))$, and $\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}$ is the importance sampling. When the importance sampling exceeds the prescribed upper or lower limit, the truncation function *clip* will return the upper bound $(1 + \epsilon)A_t^{\pi_\theta}$ or lower bound $(1 - \epsilon)A_t^{\pi_\theta}$ that does not depend on θ and so excessive update magnitude is avoided.

The detailed description of our algorithm is shown in Algorithm 1. Unlike the classical PPO algorithm, we apply the generalized advantage function estimation (GAE) [14].

$$\widehat{A_t^{GAE(\gamma,\lambda)}} = \sum\nolimits_{l=1}^{\infty} (\gamma\lambda)^l (r_t + \gamma V(s_{t+l+1}) - V(s_{t+l}))$$

GAE can improve stability and enhance discountability with its large variance. In addition, we regularize the advantage function, state function, and reward function to avoid too much variation between each sample to affect the training effect. In order to ensure that the agent can adequately explore the different UAV positions w and the allocation ratio μ, we add the policy entropy to improve the randomness of the policy. Larger policy entropy results in a more uniform distribution of action. We linearly decay the learning rate to make the later training more stable and avoid the gradient explosion during the neural network learning by gradient cropping. Finally, the gradient disappearance is prevented by the orthogonal initialization of the neural network.

6 Simulation Experiments

In this section, numerical results are presented to evaluate the performance of the proposed algorithm.

6.1 Simulation Environment Setting

As shown in Fig. 3, we consider an area of $600m \times 600m$, a standard UAV height of $H = 200m$, and a total bandwidth of $B = 100Mhz$. The total throughput of each AP is 54 Mbps and the coverage radius is $200m$, while the number of CSs within each AP is 1, 2, 3, 4 and 5, and the dormant period expectation for all CS_i is $v_i = 0.0025$. The traffic demand of each CS is set to $3Mpbs$, and the number of CSs is 20, and the total number of CSs in the coverage area of AP is 12. The UAV transmission power is $10dBm$, and the Gaussian noise power is $-100dBm$. The complexity of the channel model calculation method defined in [13] is high, so we choose the method Akram Al-Hourani et al. [11] proposed to calculate channel gain. The path loss is calculated by referring to [12]. The max step of training is set to 3×10^6, the batch size is 2048, the mini batch size is 64, both actor and critic networks have two hidden layers, each layer has 200 neurons, the learning rate is set to 10^{-4}, the discount factor is 0.99, the GAE parameter λ is 0.9, and the penalty factor α is 100.

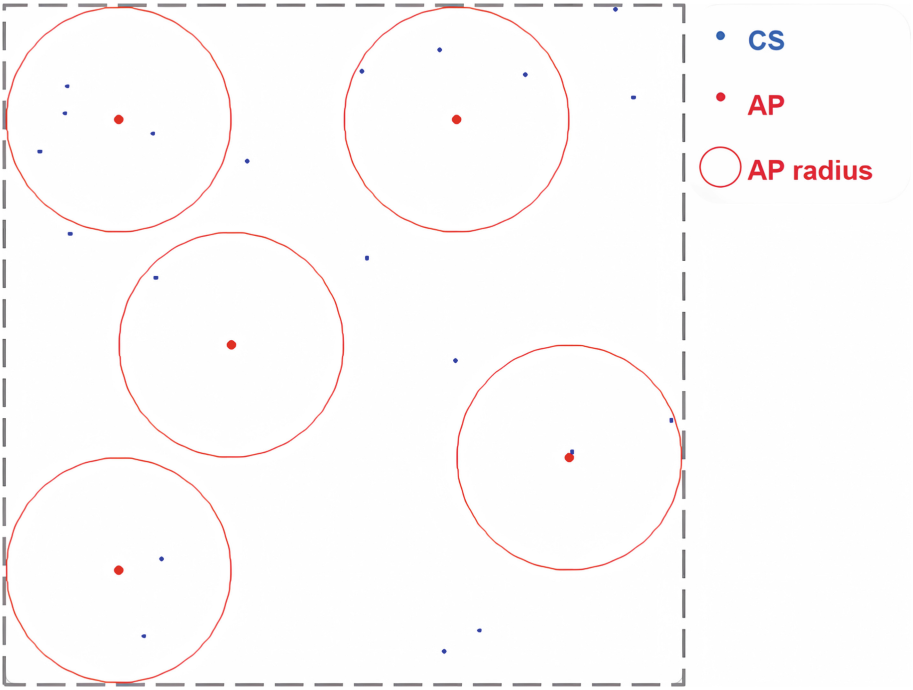


Fig. 3. The distribution of AP and CS, and the radius of AP

6.2 Performance Comparison

Since the complicated nLos channel causes the traditional convex optimization-based approach to fail, we design a plain heuristic algorithm to compare our proposed algorithm. The heuristic first initializes the UAV position and ratio, then searches for nearby positions and ratios. If a lower delay is found, the position and ratio are updated, and so on until convergence is reached. The comparison results of the heuristic algorithm and NAPPO with different position initialization are shown in Table 1. The delay of NAPPO in the table is the mean of the rewards of 500 episodes at final convergence. To avoid the excessive effect of constraint-violating reward values (reward ≤ -100) on the mean, we discard the constraint-violating values in the statistics. Position initialization heavily affects the heuristic algorithm, making it difficult to detach from the local optimum. Improper initialization (e.g., $w = (300, 300)$) even leads to constraint violations. The application of heuristics to practical communications is limited by the difficulty of correct initialization in complex environments. In contrast, NAPPO converges for every initialization and the probability of violating the constraint is only 0.657%. In addition, the convergence delay of NAPPO is lower than that of the heuristic algorithm for all given position initialization.

Figure 4 is the experimental data initialized with the CS geometric center, and Fig. 5 and Fig. 6 are the median and the harmonic mean of multiple experiments, respectively. Violation of the constraint is a low-probability event, but has a significant impact on the mean of the reward, which can be avoided by using the median and harmonic mean.

The horizontal axis is the trained episodes, the vertical axis is the negative logarithm of the reward (i.e., $\log_{10}(-reward)$), the blue line is the NAPPO delay, the yellow line is the lowest known delay obtained from 10 million random searches, and the green line is the optimal delay of the heuristic algorithm. Note that we use rewards instead of delays because the rewards include both delays and penalties for violating the constraints. The early training period has a very small reward value, on the order of -10^3, because the constraints are often violated. In the late training period, the reward value is almost equal to the actual delay, because the constraints are hardly ever violated. As shown in Fig. 4, NAPPO needs only 500 episodes of training to outperform the results of the heuristic algorithm. For Fig. 5, the mean value of the delay for the last 500 episodes is 0.6179s, with a difference of 0.1536s from the best delay. For Fig. 6, the average delay for the last 500 episodes is 0.5980s, with a difference from the best delay of 0.1337s. Thus, NAPPO can stably find a UAV navigation and traffic allocation scheme that does not violate the constraint and keep maximum delay low.

Table 1. The comparison results of the heuristic algorithm and NAPPO with different position initialization.

Initial position	Heuristic algorithm	NAPPO
CS geometry center(-18.6, 130.45)	*−0.899295968*	*−0.560426381*
AP geometry center(-80.0, 0.0)	*−0.898504235*	*−0.710564609*
Location1(-150.0, -150.0)	*−1.014797255*	*−0.893817321*
Location2(0.0, 0.0)	*−0.8711138425*	*−0.775873092*
Location3(300.0, 300.0)	*−303.49405935*	*−0.75646694*

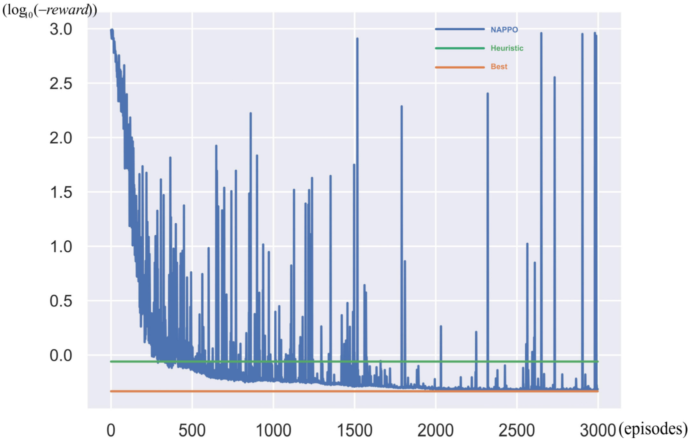


Fig. 4. The initialization position is the CS geometry center

To verify the efficiency of our maximum delay minimization algorithm, we conducted simulated experiments for minimizing the average delay and experiments for minimizing the maximum delay. The average maximum delay in the former is 0.6007*s* which lower than the latter's 1.3650*s* as we expected. In addition, we find that the average delay in the former is 0.2757s, and the average delay in the latter is 0.3269*s*, which means that minimizing the maximum delay is effective in not only reducing the maximum latency of users but also maintaining the average latency at an acceptable level.

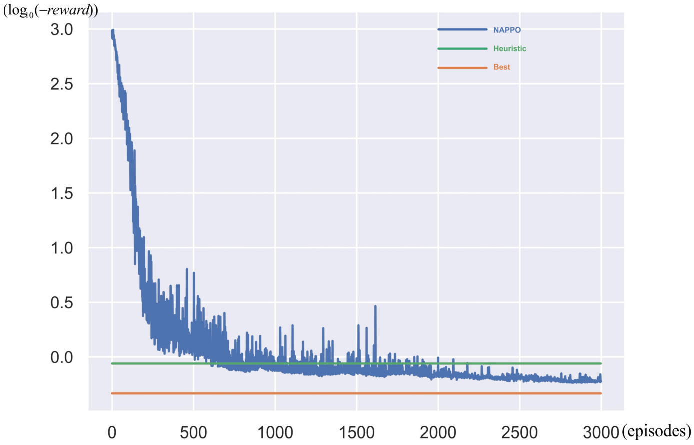


Fig. 5. Median of 5 experiments

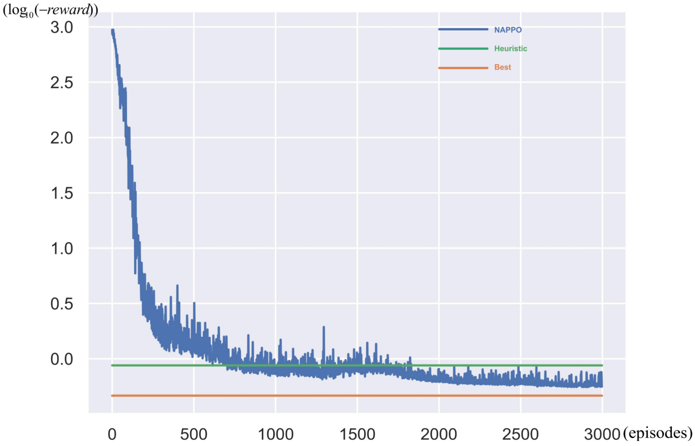


Fig. 6. Harmonic mean of 5 experiments

7 Conclusion

We proposed a deep reinforcement learning based algorithm NAPPO to solve the maximum delay minimization problem for multi-WiFi offloading of UAV base-station user traffic under non-line-of-sight links. In NAPPO, the agent obtains the transmission rate, traffic demand, and relative position information of the CSs, selects the best velocity and direction of the UAV in the current state based on this information, and properly adjusts the traffic allocation ratio of each CS. Finally, the agent determines the optimal location of the UAV and the optimal traffic ratio of the CS that will minimize the maximum average delay of the CS. The simulation results demonstrate that NAPPO produces an effective UAV positioning and traffic allocation scheme for minimizing the maximum delay.

Acknowledgment. This work is supported by Macao Polytechnic University Research Grant RP/FCA-14/2022 and Key-Area Research and Development Plan of Guangdong Province #2020B010164003. The corresponding author is Hong Shen.

References

1. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal Policy Optimization Algorithms. CoRR. abs/1707.06347 (2017)
2. Shakoor, S., Kaleem, Z., Do, D., Dobre, O., Jamalipour, A.: Joint optimization of UAV 3-D placement and path-loss factor for energy-efficient maximal coverage. IEEE Internet Things J. **8**, 9776–9786 (2021)
3. Zhan, C., Hu, H., Liu, Z., Wang, Z., Mao, S.: Multi-UAV-enabled mobile-edge computing for time-constrained IoT applications. IEEE Internet Things J. **8**, 15553–15567 (2021)
4. Zeng, Y., Xu, X., Jin, S., Zhang, R.: Simultaneous navigation and radio mapping for cellular-connected UAV with deep reinforcement learning. IEEE Trans. Wirel. Commun. **20**, 4205–4220 (2021)
5. Li, X., Cheng, S., Ding, H., Pan, M., Zhao, N.: When UAVs meet cognitive radio: offloading traffic under uncertain spectrum environment via deep reinforcement learning. IEEE Trans. Wirel. Commun. **22**, 824–838 (2023)
6. Ali, M., Zeng, Y., Jamalipour, A.: Software-defined coexisting UAV and WiFi: delay-oriented traffic offloading and UAV placement. IEEE J. Sel. Areas Commun. **38**, 988–998 (2020)
7. Tian, N., Zhang, Z.G.: Vacation Queueing Models Theory and Applications. Springer, Boston (2006)
8. Zhou, Z., Guo, D., Honig, M.: Licensed and unlicensed spectrum allocation in heterogeneous networks. IEEE Trans. Commun. **65**, 1815–1827 (2017)
9. Kakade, S.: A Natural Policy Gradient. NIPS, pp. 1531–1538 (2001)
10. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization altitude for maximum coverage. ICML **37**, 1889–1897 (2015)
11. Al-Hourani, A., Sithamparanathan, K., Lardner, S.: Optimal LAP altitude for maximum coverage. IEEE Wirel. Commun. Lett. **3**, 569–572 (2014)
12. 3GPP Technical Specification Group Radio Access Network: Study on Enhanced LTE Support for Aerial Vehicles. 3GPP (2017)
13. ITU-R Propagation Data and Prediction Methods Required for the Design of Terrestrial Broadband Radio Access Systems Operating in a Frequency Range From 3 to 60 GHz (2012)
14. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: High-dimensional continuous control using generalized advantage estimation. International Conference On Learning Representations (ICLR) (2016)

Triple-Path RNN Network: A Time-and-Frequency Joint Domain Speech Separation Model

Yu-Huan Zhai[1], Qiang Hua[1](✉), Xiao-Wen Wang[1], Chun-Ru Dong[1], Feng Zhang[1], and Da- Chuan Xu[2]

[1] College of Mathematics and Information Science, Hebei University, Baoding 071002, People's Republic of China
huaq@hbu.edu.cn

[2] Beijing Institute for Scientific and Engineering Computing, Beijing University of Technology, Beijing 100124, People's Republic of China

Abstract. Studies in speech separation have achieved significant success in recent years. To correctly separate the mixture signals, it is critical to encode the signals into an appropriate latent space. Existing speech separation methods include transforming mixed signals into frequency domain space or time domain space. The frequency domain features (spectrogram) are generated by STFT, which is closely related to speech articulation and reflects the energy of speech directly. The time domain features are learned from a latent embedding space, and the separation effect is facilitated by the end-to-end structure. However, these methods are based on the representations from only one domain, which is insufficient for providing a speech separation encoding space that is completely separable. Therefore, a Triple-Path Recurrent Neural Network (TPRNN) that fuse features from two domains is proposed. It employs a spectrogram as auxiliary information to improve the performance of speech separation. Experimental results on the Wall Street Journal (WSJ0) dataset show that this approach is beneficial to improve speech separation performance.

Keywords: Speech separation · deep learning · waveform · spectrogram

1 Introduction

Humans can easily pay attention to the speaker we want to hear in a noisy environment. However, this is a complicated problem for a computer. Monaural speech separation is split into two categories: frequency domain methods [1–8] and time domain approaches [9–19]. The former uses a short-time Fourier transform (STFT) to obtain a spectrogram and subsequently creats the origin waveforms using inverse STFT by separating the time-frequency (T-F) bins corresponding to each source (iSTFT). This way can restore the input audio nearly perfectly due to the reversibility of the STFT and iSTFT. However, this approach has several drawbacks: (1) As a general signal transformation, STFT is not

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 239–248, 2024.
https://doi.org/10.1007/978-981-99-8211-0_22

the ideal option for speech separation tasks. (2) The separation performance is upper-bounded by the oracle masks during the training process. (3) The main drawback is that phase information is not fully utilized in the separation process.

These problems are attributable to formulating the separation problem in the frequency domain. Therefore, a more straightforward way of tackling these obstacles is to separate the signal in the time domain directly. The benefit of this approach is that the separation processes, including the encoder, separator, and decoder, can be integrated into an end-to-end model. Such as Tasnet [9] modeled the mixture waveform with encoder-decoder architecture and replaced the STFT stage with a real-valued and trainable module. In addition, current time domain approaches focus on improving the performance of the separator layer. For example, Conv-TasNet [11] replaced the separator layer with temporal convolutional networks (TCNs). DPRNN [12] used a dual-path strategy to model long sequences. GALR [16] combined the advantages of the attention mechanism and recurrent neural network (RNN), which was used on low-resource devices. DPTNet [14], SepFormer [17], and Transmask [18] utilized improved transformer layers as separators.

These time domain methods have greatly improved the performance of speech separation. However, compared to approaches using spectrograms in the frequency domain, the waveform may more accurately describe the various speech realizations, since it can reflect the structure of the input signal. Both methods have advantages and disadvantages: The frequency domain method is to decompose the signals into each sub-band space, which has stable internal properties and constant frequency, and distinguish the sub-band space in the separation stage. The time domain method distinguishes the channels in the separation phase. However, for speech separation tasks, the space represented by a single latent domain only is insufficient to provide a completely separable representation. In addition, it remains a problem whether we can obtain an accurate separation by proper latent space. This motivates us to develop a model that incorporates different separation spaces. The main contributions of this study are summarized as follow:

- To obtain a better separable encoding space for speech separation, a new model (TPRNN) is proposed to allow features from two different domains to be fused.
- Frequency domain features is employed as auxiliary information to help time domain separation methods improve the separation performance.
- Model effectiveness is verified on WSJ0-2mix and WSJ0-3mix datasets.

2 Triple-path Recurrent Neural Network

TPRNN consists of three parts: a encoder, a separator, and a decoder. The overall framework is illustrated in Fig. 1. The encoder layer includes two parallel procedures, generating feature maps of both domain. The separator layer employs three RNNs, with the dual-path RNN used to obtain the information from the time domain features, and the single-path RNN employed to process the frequency domain feature. The decoder layer restores the original waveforms using masks. In the following sections, we will describe the encoder and the separator layer in detail.

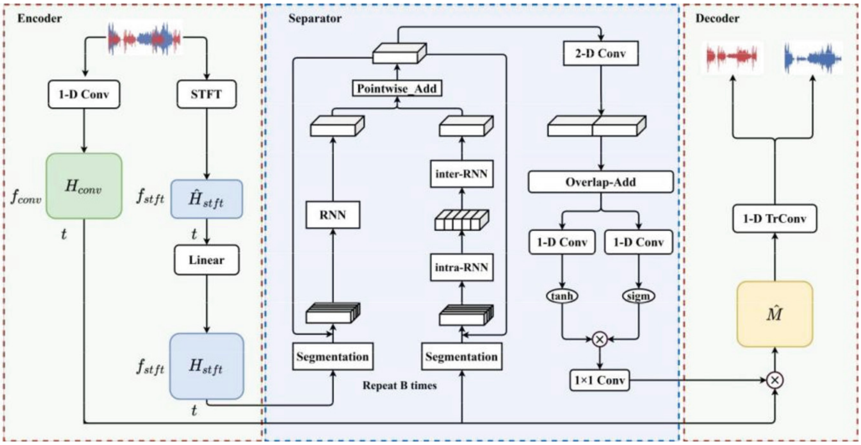


Fig. 1. Framework of speech separation with a triple-path RNN network. The encoder layer generates both time and frequency domain features. Each feature is processed in different way and fused in the separator layer. The decoder layer utilizes only the time domain feature as input.

2.1 Encoder

Let the mixture signal be $x(t) \in R^{1\times T}$, representing C sources $s_1(t), \ldots, s_C(t) \in R^{1\times T}$ captured from the same microphone. For brevity, we use x instead of $x(t)$ in the following. In most current methods [12–15, 17–19], the encoder layer employs 1-D convolution to obtain representations of embedding features. However, to fully leverage the information from frequency domain, TPRNN introduces an additional processing path for the frequency domain features. As shown in Fig. 1, in the encoder layer, the input mixture x is fed into two branches separately. The time domain representation $H_{conv} \in R^{N\times L}$ by 1-D convolution with N filters of lengh L. And the magnitude spectrogram $\hat{H}_{stft} \in R^{E\times L}$ with E frequency channels of length L is obtained by STFT. By a linear transformation, the $\hat{H}_{stft}$ expands to the N dimension layer to get the spectrogram $H_{stft} \in R^{N\times L}$.

STFT assumes that a nonstationary signal can be treated as stationary during brief time periods defined by a window. By adjusting the window along the time dimension and examining the signal segment by segment, a group, local magnitude spectrogram of the signal is obtained [20, 21]. This process is analogous to moving a 1-D convolution kernel along a single direction with a specified stride. In our experiments, we set the same window size and stride for both approaches while aligning the time frames to integrate both representations from different domains. To effectively utilize the information of the magnitude spectrogram, we align the f_{conv} and f_{stft} channels by using Eqs.(1)–(3). In this manner, both domain features will have the same shape:

$$H_{conv} = f(conv\_1D(x)). \tag{1}$$

$$\hat{H}_{stft} = \mathrm{Re}[STFT(x)]. \tag{2}$$

$$H_{stft} = W_1\left(\hat{H}_{stft}\right) + q_1. \tag{3}$$

where $f(\cdot)$ is an activation function, $conv\_1D(\cdot)$ is 1-D convolution operation, $STFT(\cdot)$ is the short-time Fourier transform, and $Re[\cdot]$ is an absolute value operation. $W_1 \in R^{N\times E}$ and $q_1 \in R^{N\times L}$ forms a linear layer.

2.2 Separator

The separator layer is composed of three stages: segmentation, estimation of the source mask, and feature fusion process.

1) Segmentation: The segmentation stage creates S equal sized chunks $D_s^{conv} \in R^{N\times K}, s = 1, \ldots, S$ and $D_s^{stft} \in R^{N\times K}, s = 1, \ldots, S$ by dividing two features into chunks of length K and hop size P. Every sample in H_{stft} and H_{conv} only appears in *K/P* chunks as a result of zero padding in the start and last chunks. These chunks are then concatenated together to obtain a 3D tensor $T_{conv} = \left[D_1^{conv}, \ldots, D_S^{conv}\right] \in R^{N\times K\times S}$ and $T_{stft} = \left[D_1^{stft}, \ldots, D_S^{stft}\right] \in R^{N\times K\times S}$.

2) Triple-path RNN block for estimation of the source mask: The TPRNN block employs three RNNs, where the information from the time domain is processed by a dual-path RNN and frequency domain features are obtained by a single-path RNN. Frequency domain procedure gets the frequency domain information of the input sequence. We use a bi-directional LSTM(Bi-LSTM) with H hidden nodes [16] to let the input segment be $T_b^{stft} \in R^{N\times K\times S}, b = 1, \ldots, B$, where B is the number of blocks. The subscript b is omitted for the convenience of expression:

$$R_{stft} = \left[W_2 RNN\left(\left(T_{stft}[:, :, s]\right)\right) + q_2, s = 1, \ldots, S\right]. \tag{4}$$

where $R_{stft} \in R^{N\times K\times S}$ is the output of the $RNN(\bullet)$ and $T_{stft}[:, :, s] \in R^{N\times K}$ is the s^{th} segment. The output R_{stft} will pass a layer normalization operation $LN(\bullet)$ with a residual connection [16] to the input T_{stft}:

$$\overline{R}_{stft} = LN\left(R_{stft}\right) + T_{stft}. \tag{5}$$

where $\overline{R}_{stft} \in R^{N\times K\times S}$ is the output of the single-path RNN. When the information comes from time domain, RNN submodule is used to learn local dependencies at a lower context level:

$$R_{intra} = \left[W_3 RNN\left(\left(T_{conv}[:, :, i]\right)\right) + q_3, i = 1, \ldots, S\right]. \tag{6}$$

$$\overline{R}_{intra} = LN\left(R_{intra}\right) + T_{conv}. \tag{7}$$

where $T_{conv}[:, :, i] \in R^{N\times K}$ is the i^{th} segment. The output $\overline{R}_{intra} \in R^{N\times K\times S}$ is normalized by a layer and connected to the input with residuals. Similarly, the output $\overline{R}_{intra}$ served as the input for the following RNN module to learn long-term global dependencies by Eqs.(8)–(9):

$$R_{inter} = \left[W_4 RNN\left(\left(\overline{R}_{intra}[:, k, :]\right)\right) + q_4, k = 1, \ldots, K\right]. \tag{8}$$

$$\overline{R}_{inter} = LN\left(R_{inter}\right) + R_{intra}. \tag{9}$$

where $\overline{R}_{intra}[:, k, :] \in R^{N\times S}$ is the k^{th} segment. $\overline{R}_{inter} \in R^{N\times K\times S}$ is the output of the dual-path RNN.

3) Feature fusion process: The outputs of both paths $\overline{R}_{stft}$ and $\overline{R}_{inter}$ are obtained separately from these two processing paths, which need to be operated by an addition operation. Then the result $T \in R^{N\times K\times S}$ is used as input for the next cycle of two parallel procedures. It is worth noting that the first input of the two processing paths is the feature map of the corresponding domain respectively. The subsequent input is the fused result based on Eq. (10):

$$T = \overline{R}_{stft} + \overline{R}_{inter}. \tag{10}$$

2.3 Signals Reconstruction

The overlap-add method was described in DPRNN [12] to transform the output of the last TPRNN block back to a sequence and obtain the mask $M_c \in R^{N\times L}$, $c = 1, \ldots, C$, for c^{th} source. The source c's mask M_c and the encoder H_{conv} output are multiplied elementwise to create the input M_c to the decoder. Therefore, the following is an expression for the decoder's transformation:

$$\hat{M}_c = H_{conv} \otimes M_c. \tag{11}$$

$$S_c = Tr\_conv1D\left(\hat{M}_c\right). \tag{12}$$

where $\otimes$ denotes the dot product operation, $Tr\_conv1D(\cdot)$ is 1-D transpose convolution and $S_c \in R^{1\times T}$ is the sequence of the c^{th} speaker.

3 Experimental Setup

3.1 Datasets

In our experiments, we use the WSJ0-2mix and WSJ0-3mix [5] datasets, derived from the Wall Street Journal (WSJ0) corpus. These datasets consist of 30 h of training data at 8 kHz, generated from *si_tr_s* set. Additionally, there are 5 h of test dataset collected from 16 different speakers in the *si_dt_*05 and *si_et_*05 directories of the WSJ0. For consistency, all data were resampled at a sampling rate of 8 kHz.

3.2 Experimental Setup and Training Details

For a fair comparison, we use the same experimental setup with DPRNN [12]. The encoder layer consists of 256 convolutional filters with a kernel size and stride setting as DPRNN. The STFT window size matches the kernel size of the 1-D convolution, and a Hann window is utilized for the STFT. The decoder layer also shares the same kernel size and stride as the encoder layer. Regarding the Separator layer, the TPRNN block size B = 6. The number of hidden nodes H is set to 128. All models are trained for 100 epochs [12]. During training, the learning rate is decayed by 0.98 every two epochs, starting from the original value of 1e-3. In case no significant improvement is observed on the validation set for ten consecutive epochs, early halting is applied. The Adam optimizer [22] is used, and gradient clipping is employed with a maximum L2-norm of 5 to stabilize the training process. All the models were trained using PyTorch [23] on a single NVIDIA GEFORCE 3090 GPU with 24G of memory devices.

4 Results And Discussions

To evaluate the separation performance of the suggested model, the SI-SNR improvement (SI-SNRi) and signal-to-distortion ratio improvement (SDRi) are used in each experiment [24], where K in the table is the block length during segmentation and win is the kernel size of 1-D convolution.

Table 1. DPRNN and TPRNN performances in WSJ0-2MIX with various configurations.

Model	win	K	Param	SI-SNRi	SDRi
DPRNN [12]	2	250	2.6M	18.8	19.0
	4	200		17.9	18.1
	8	150		17.0	17.3
	16	100		16.0	16.2
TPRNN	2	250	3.9M	18.4	18.7
	4	200		19.0(6.15%)	19.2(6.08%)
	8	150		18.4(8.24%)	18.6(7.51%)
	16	100		16.0	16.2

4.1 Results on WSJ0-2MIX

DPRNN [12] is used as a baseline method because it is a SOTA method and it is quite similar to the proposed method in this study. In our experiment, TPRNN uses the same window length and the same segment size as DPRNN. As shown in Table 1, we observe that compared with DPRNN, TPRNN achieves 1.1 dB absolute improvements in SI-SNRi and SDRi with 6.15% and 6.08% improvement when win = 4. When win = 8, it reaches 1.4 dB and 1.3 dB total improvements in SI-SNRi and SDRi with 8.24% and 7.51%. From the Table 1, we can see that the best result obtained by TPRNN with win = 4 is 0.2dB higher than that obtained by DPRNN when win = 2. However, it can be seen from Fig. 2 that when win = 2, the performance of TPRNN starts to decline, because the spectrogram generated in the frequency domain is extremely compressed. Overall, the experimented results indicate that the structure of TPRNN improves the separation performance.

Table 2 shows the results of several well-known methods and TPRNN on the WSJ0-2mix dataset. TPRNN in stride = 2 is implemented, achieving an SI-SNRi of 19.0 dB

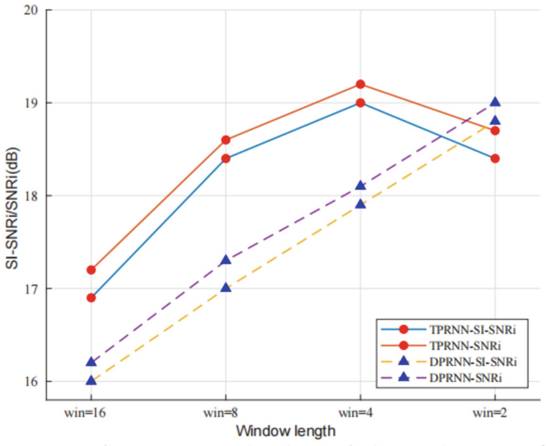


Fig. 2. Performance evaluation of the various configurations in Table 1.

and an SDRi of 19.2 dB, respectively. Sepformer in the table achieves better performance at the cost of a large model size by using a more deep and wider transformer layer. The model size of TPRNN is only 0.15 of Sepformer. With the same stride, TPRNN improves by 7.34% on SI-SNRi and 7.26% on SDRi compared with GALR, 6.15% on SI-SNRi, and 6.08% on SDRi compared with DPRNN.

Table 2. Comparison of separation performance on WSJ0-2MIX Dataset.

Method	Param	Stride	SI-SNRi	SDRi
DPCL++ [5]	13.6M	64	10.8	-
uPIT-BLSTM [6]	92.7M	128	-	10.0
BLSTM-Tasnet [9]	23.6M	20	13.2	13.6
Conv_Tasnet [11]	5.1M	10	15.3	15.6
Two-step [25]	8.6M	10	16.1	-
DeepCASA [26]	12.8M	1	17.7	18.0
SuDoRM-RF [13]	2.7M	10	17.0	-
GALR [16]	1.5M	2	17.7	17.9
DPRNN [12]	2.6M	1	18.8	19.0
DP-RCNet [19]	9.2M	8	17.7	18.0
DPTNet [14]	2.6M	1	20.2	20.6
SepFormer [17]	26M	8	22.3	22.4
TPRNN	3.9M	2	19.0	19.2

Table 3. Comparison of separation performance on WSJ0-3MIX Dataset.

Method	SI-SNRi	SDRi	Param
ConvTasnet [11]	12.7	13.1	5.1M
DPRNN[12]	14.7	-	2.6M
TPRNN	15.5	15.8	3.9M

4.2 Results on WSJ0-3MIX

The models perform well on the WSJ0-3mix dataset, as shown in Table 3. For the WSJ0-3mix dataset, we utilized the TPRNN model with the best-performing architecture from Table 1. The results in Table 3 indicate that TPRNN achieved an SI-SNRi of 15.5 dB and an SDRi of 15.8 dB, demonstrating its capability to enhance separation performance. Overall, the results presented in Table 3 demonstrate the effectiveness of TPRNN in enhancing the separation performance on the WSJ0-3mix dataset.

5 Conclusion

In this paper, we propose a triple-path recurrent neural network (TPRNN) that fuses time domain and frequency domain features to improve the performance of speech separation. Experimental results on WSJ0-2mix and WSJ0-3mix datasets indicate that the proposed TPRNN enhances separation performance by using frequency domain features as auxiliary data on time domain techniques. On the WSJ0-2mix dataset, SI-SNRi is improved by 6.15% compared to the baseline method. In the future, we would like to expand this study with information from more frequency domain feature to further improve the separation performance.

Acknowledgment. This work was supported by Beijing Natural Science Foundation Project under Grant Z200002.

References

1. Wang, D., Chen, J.: Supervised speech separation based on deep learning: an overview. IEEE/ACM Trans. Audio Speech Lang. Process. **26**(10), 1702–1726 (2018)
2. Lu, X., Tsao, Y., Matsuda, S., Hori, C.: Speech enhancement based on deep denoising autoencoder. Interspeech **2013**, 436–440 (2013)
3. Xu, Y., Du, J., Dai, L.-R., Lee, C.-H.: An experimental study on speech enhancement based on deep neural networks. IEEE Signal Process. Lett. **21**(1), 65–68 (2013)
4. Yu, D., Kolbæk, M., Tan, Z.-H., Jensen, J.: Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 241–245. IEEE (2017)
5. Hershey, J.R., Chen, Z., Le Roux, J., Watanabe, S.: Deep clustering: discriminative embeddings for segmentation and separation. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Pro-cessing (ICASSP), pp. 31–35. IEEE (2016)

6. Kolbæk, M., Yu, D., Tan, Z.-H., Jensen, J.: Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(10), 1901–1913 (2017)
7. Chen, Z., Luo, Y., Mesgarani, N.: Deep attractor network for single-microphone speaker separation. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 246–250. IEEE (2017)
8. Xu, Y., Du, J., Dai, L.-R., Lee, C.-H.: A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(1), 7–19 (2014)
9. Luo Y., Mesgarani, N.: Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 696–700. IEEE (2018)
10. Stoller, D., Ewert, S., Dixon, S.: Wave-u-net: a multi-scale neural network for end-to-end audio source separation. arXiv preprint arXiv:1806.03185 (2018)
11. Luo, Y., Mesgarani, N.: Conv-Tasnet: surpassing ideal time– frequency magnitude masking for speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. **27**(8), 1256–1266 (2019)
12. Luo, Y., Chen, Z., Yoshioka, T.: Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation. In: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 46–50. IEEE (2020)
13. Tzinis, E., Wang, Z., Smaragdis, P.: Sudo RM-RF: efficient networks for universal audio source separation. In: 2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6. IEEE (2020)
14. Chen, J., Mao, Q., Liu, D.: Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation. arXiv preprint arXiv:2007.13975 (2020)
15. Zeghidour, N., Grangier, D.: Wavesplit: End-to-end speech separation by speaker clustering. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 2840–2849 (2021)
16. Lam, M.W., Wang, J., Su, D., Yu, D.: Effective low-cost time-domain audio separation using globally attentive locally recurrent networks. In: 2021 IEEE Spoken Language Technology Workshop (SLT), pp. 801–808. IEEE (2021)
17. Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., Zhong, J.: Attention is all you need in speech separation. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 21–25. IEEE (2021)
18. Zhang, Z., He, B., Zhang, Z.: Transmask: a compact and fast speech separation model based on transformer. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5764–5768. IEEE (2021)
19. Yang, X., Bao, C.: Embedding recurrent layers with dual-path strategy in a variant of convolutional network for speaker-independent speech separation. arXiv preprint arXiv:2203.13574 (2022)
20. Yang, G.-P., Tuan, C.-I., Lee, H.-Y., Lee, L.: Improved speech separation with time-and-frequency cross-domain joint embedding and clustering. arXiv preprint arXiv:1904.07845 (2019)
21. Sturmel, N., Daudet, L., et al.: Signal reconstruction from stft magnitude: a state of the art. In: International conference on digital audio effects (DAFx), pp. 375–386 (2011)
22. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Paszke, A., et al.: Automatic differentiation in pytorch (2017)
24. Le Roux, J., Wisdom, S., Erdogan, H., Hershey, J.R.: Sdr–half-baked or well done? In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 626–630. IEEE (2019)

25. Tzinis, E., Venkataramani, S., Wang, Z., Subakan, C., Smaragdis, P.: Two-step sound source separation: training on learned latent targets. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 31–35. IEEE (2020)
26. Liu, Y., Wang, D.: Divide and conquer: a deep casa approach to talker-independent monaural speaker separation. In: IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 27, no. 12, pp. 2092–2102 (2019)

Design of Query Based Gallery Selector and Mask-Aware Loss for Person Search

Qiang Hua[1](✉), Ao Sun[1], Yu-Chen Liu[1], Feng Zhang[1], Chun-Ru Dong[1], and Da-Chuan Xu[2]

[1] College of Mathematics and Information Science, Hebei University, Baoding 071002, People's Republic of China
huaq@hbu.edu.cn

[2] Beijing Institute for Scientific and Engineering Computing, Beijing University of Technology, Beijing 100124, People's Republic of China

Abstract. Person search is a challenging computer vision task that aims to simultaneously locate and identify a query person from panoramic images. To address the issue of scene similarity and its impact on search accuracy and efficiency, we propose a query based gallery selector module that employs cosine similarity to calculate the similarity between candidate images in the gallery and the query persons feature embedding, then selects and reorders images in the gallery based on their similarity to the query person, thus improving the accuracy and efficiency of searching. Furthermore, we introduce a mask-aware mechanism that improves the localization loss function for predicted bounding boxes. During training, the network is guided to increase its robustness in occluded scenarios. Experimental results on public person search datasets PRW and CUHK-SYSU demonstrate the effectiveness of our proposed method.

Keywords: Person Search · Query-Based Gallery Selector · Mask-aware loss

1 Introduction

Person search is a challenging computer vision problem, where the task is to locate and identify a given target person from a gallery of real-world scene images, which is widely applied in many applications, such as intelligent surveillance and assisted driving.

As a joint task of person detection and re-identification (re-id), person search not only requires dealing with the challenges existing in these two individual sub-tasks, but also needs to jointly optimize the diverse objectives of both sub-tasks together. Existing methods can be mainly divided into two classes: one-step [1–3] and two-step [4–9] approaches.

Two step approaches usually adopt two independent networks for detection and re-id, respectively. First use a detection network to detect persons from the images, then use another re-id network to perform re-id based on the detected persons. Different from two-step approaches, one-step approaches aim to perform detection and re-id in a single network. As shown in Fig. 2, proposal from dense anchors using a RPN sub-network followed by separate detection and re-id steps.

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 249–259, 2024.
https://doi.org/10.1007/978-981-99-8211-0_23

In summary, our contributions are as follows:

We propose a Query-Based Gallery Selector (QBGS) for selecting more valued scene in gallery and then do person search stage.

We propose a similarity measurement method for pedestrian search based on the mask reward mechanism called Mask-Aware IoU (MAIoU) and MAIoU Focal loss(MAIF) is proposed, which suppresses the anchor boxes with higher similarity in the positioning regression task, further improving the search results. Experimental results demonstrate the effectiveness of our proposed method in enhancing person search performance, see as Fig. 1.

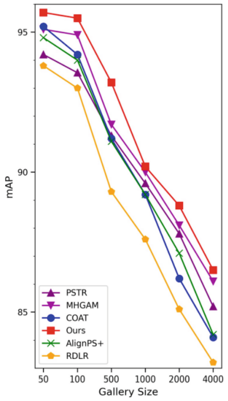


Fig. 1. Experimental results for different gallery size settings on the CUHK-SYSU dataset.

2 Related Work

2.1 Pedestrian Detection

The goal of pedestrian detection is to locate and identify people within a scene and draw a bounding box around them. It is a critical task in the field of computer vision, particularly in the context of person search.

Recent years, many deep learning methods have been proposed for pedestrian detection. One of the most widely used methods is the Region-based Convolutional Neural Network (R-CNN) family [10]. These methods use a two-stage detection approach, first generate region proposals and then classify them as pedestrian or non-pedestrian.

Pedestrian detection in person search is a more challenging task than generic pedestrian detection, as the aim is to identify a specific individual among a large number of pedestrians. Therefore, many methods have been proposed to integrate pedestrian detection into person search.

In summary, pedestrian detection is a crucial step in person search, and various methods have been proposed for this task. Both deep learning-based and traditional computer vision methods have been explored, and techniques for integrating pedestrian detection with person re-identification have been developed. The incorporation of contextual information is a promising direction for future research in this area.

2.2 IoU design

IoU stands for Intersection over Union, it is a commonly used evaluation metric in computer vision and object detection tasks[10]. It measures the overlap between the predicted bounding box and the ground truth bounding box.

The IoU is calculated as the ratio of the intersecting area between the predicted and ground truth bounding boxes to the union of the two bounding boxes. It is expressed as a value between 0 and 1, where a value of 1 indicates perfect overlap, and a value close to 0 indicates little to no overlap, it used to evaluate the performance of object detection algorithms, as well as for tuning their hyper-parameter.

A detection result with IoU $>$ 0.5 is generally considered to be reliable. Existing pedestrian search methods based on Faster R-CNN do not design the corresponding intersection-over- union based on the characteristics of the pedestrian detection task, which limits their assistance to the subsequent re-identification task.

3 Methods

3.1 Base Model

Our baseline model is an end-to-end architecture based on SeqNet [11]. We make modifications to the model backbone, simplifying the two-stage detection process and improving the training flow to achieve better performance. The architecture of our model is depicted in Fig. 2.

Following SeqNet usage of the first four CNN blocks (conv1–4) from ResNet50 for backbone features, we use the analogous layer after RPN called global RCNN layers and copy an embedding head for re-id. After adaptive max pooling and the similarity measurement method of the detection box and the GT box are replaced by the mask-aware IoU based on the mask reward mechanism.

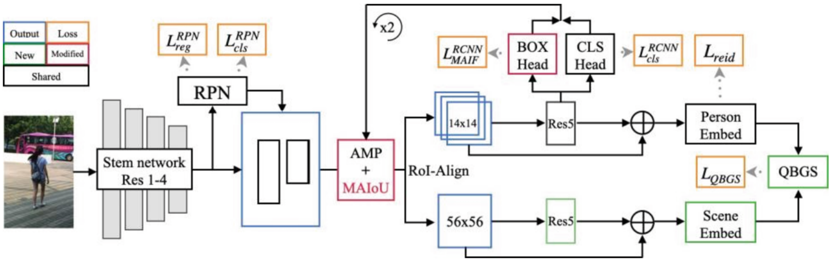


Fig. 2. Ours baseline model, modules modified from SeqNet are colored red, new modules are colored green. (Color figure online)

Our proposed network incorporates another branch dedicated to learning scene embedding for the gallery images while performing pedestrian feature embedding through RoI-Align. Since the gallery scenes have a larger view compared to individual pedestrian bounding boxes, the corresponding feature map size is set to 56 × 56.

3.2 Query Based Gallery Selector

The goal of query based gallery selector is to design a module that can re-rank the gallery scenes based on the cosine similarity scores s_{qbgs} of the gallery-scene feature embedding. When pedestrian appear in their scenes, the optimization of QBGS is to pull the pedestrian feature embedding as close as possible to the scene feature embedding, as shown in Fig. 3.

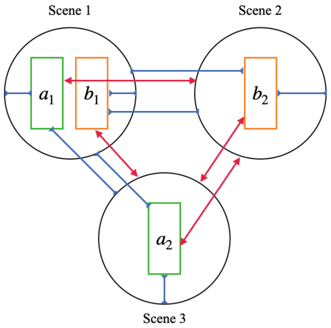


Fig. 3. a, b refers different person appear in different scenes, blue connectors represent attraction, meaning two embedding are pushed together, red connectors represent repulsion, meaning two embedding are pulled apart.

During the inference stage, a subset of images is selected from the gallery, and the gallery images with higher scores are re-ranked, as shown in Fig. 4. Images with scores lower than the hard threshold λ_{qbgs} (chosen as 0.5) are discarded, leaving behind a subset of gallery images to be used for pedestrian detection and re-id tasks.

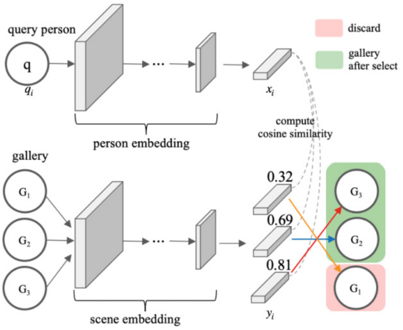


Fig. 4. Illustration of QBGS module, G_1 discard because sore of G_1 lower than 0.5.

Let $x_i \in \mathbb{R}^d$ be the feature embedding obtained from the network for the query pedestrian q_i and correspondingly, s_j be the scene in the gallery where the pedestrian appears. Let $y_j \in \mathbb{R}^d$ denote the scene feature embedding obtained from the network. The set of all query embedding $\mathrm{x_i}$ is denoted by X and the set of all scene embedding

y_j is denoted by Y, where $N = |X|, M = |Y|$. The indicator function for a positive query-scene pair is defined as follows:

$$I_{i,j}^{Q} = \begin{cases} 1, & if\ q_i\ in\ s_j \\ 0, & otherwise \end{cases}. \tag{1}$$

The index set for further selection of negative sample pair defined as $K_{i,j}^{Q} = \{k \in 1, \ldots, M | I_{i,j} = 0\}$, which cosine similarity calculation formula is defined as:

$$\sin(u, v) = \frac{u^T v}{\|u\| \ \|v\|}, u, v \in \mathbb{R}^d. \tag{2}$$

Loss function for calculating the positive sample pairs of query-scene using cross-entropy is as follows:

$$\ell_{i,j}^{Q} = -\log\left(\frac{exp\left(sim\left(x_i, y_j\right)/\tau\right)}{\sum_{k \in K_{i,j}^{Q}} exp\left(sim\left(x_i, y_j\right)/\tau\right)}\right). \tag{3}$$

where τ is temperature coeffiicient. In summary, the overall loss function of the QBGS module considers all query-scene pairs and sums up the losses for all positive sample pairs as:

$$L_{qbgs} = \sum_{i=1}^{N} \sum_{j=1}^{M} I_{i,j} \ell_{i,j} \tag{4}$$

3.3 Mask Aware Design

In real world pedestrian search, lower parts of pedestrian are often heavily occluded, as shown in Fig. 5. The annotated GT box cannot capture the complete information of the pedestrian while avoiding redundant information caused by occlusion. Therefore, the predicted boxes generated from such GT boxes may suffer from poor quality.

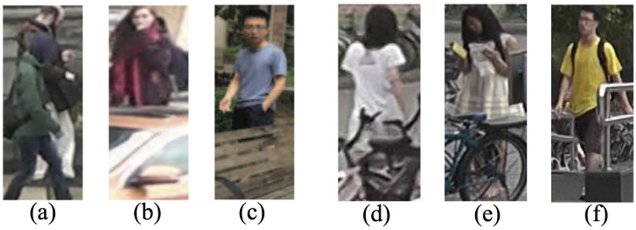


Fig. 5. (a), (b), (c) represent the occlusion issues of pedestrians in the real world, while (d), (e), (f) represent the occlusion issues of pedestrians captured by different cameras in the PRW dataset.

To address the aforementioned issues, we propose a mask-aware intersection over union (MAIoU) metric based on a mask reward mechanism. By designing masks with different coverage ratios, we improve search performance without introducing any additional parameters or network structures.

Mask Over Box: Inspired by the article [12] we adopted the Mask-over-box (MOB) as an evaluation metric to measure the proportion of masked region on the anchor box.

$$MOB\left(B^{GT}, M\right) = \frac{\left|B^{GT} \cap M\right|}{\left|B^{GT}\right|} \in [0, 1]. \tag{5}$$

Here, B^{GT} refers to the GT box and M represents a manually designed mask used to cover a certain part of the GT box. The MOB value is a hyper-parameter, in our study, we set MOB in {0.5, 0.6, 0.7, 0.8, 0.9, 1.0} (Fig. 6).

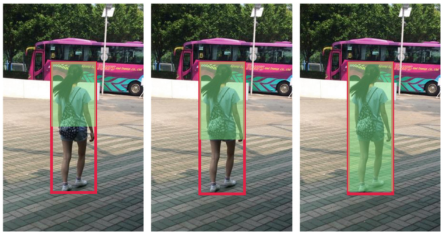


Fig. 6. Mask visualization with different *MOB*, from left to right: 0.5, 0.6, 1.0

MAIoU: Based on the previously mentioned MOB evaluation metric, we have designed and proposed MAIoU mechanism. MAIoU replaces the intersection area between the detection box and the ground truth box with the intersection area between the detection box and the mask region and using MOB^{-1} as the scaling factor. The mathematical expression for MAIoU is as Formula (6), a clearer comparison is shown as Fig. 7.

$$MAIoU = \frac{1}{MOB\left(B^{GT}, M\right)} \cdot \frac{\left|B^{det} \cap M\right|}{\left|B^{det} \cup B^{GT}\right|} \tag{6}$$

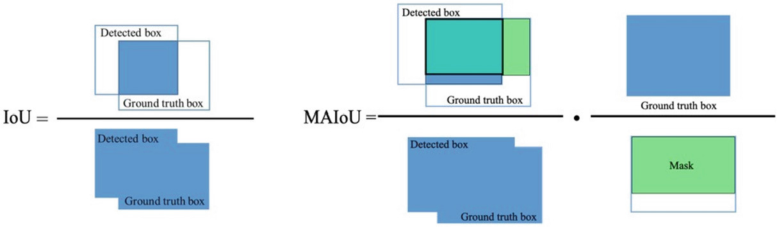


Fig. 7. Different visuals of IoU design, left: IoU, right: MAIoU

3.4 Loss Function

As shown in Fig. 2, the overall architecture of the network utilizes a combined loss function consisting of six individual loss functions. These loss functions are used to supervise the training of pedestrian detection, re-id, and QBGS tasks. The pedestrian detection component employs a two-stage Faster R-CNN approach. The formulation of L_{det} as follows:

$$L_{det} = \sum_{m \in M} L_{cls}^{m} + L_{reg}^{m}, M = \{RPN, RCNN1, RCNN2\} \tag{7}$$

The regression loss for detection boxes generated in the second stage of the network is specifically designed to improve the accuracy of box localization L_{reg}^{RCNN} we refers to RetinaNet [13] by using MAIoU as scale factor to re-design regression loss function named (MAIoU Factor, MAIF), as L_{MAIF}^{RCNN} in Eq. (8).

$$L_{MAIF}^{RCNN} = \mu \cdot L_{reg}^{RCNN}, \mu = -(1 - MAIoU)\gamma \log(MAIoU). \tag{8}$$

Here, γ is hyper-parameter, chosen as 0.4 in our study, the commonly chosen of L_{reg}^{RCNN} is the smooth L1 loss defined as:

$$L_{reg}^{RCNN} = \begin{cases} 0.5x^2, & if \ |x| < 1 \\ |x| - 0.5, & otherwise \end{cases}. \tag{9}$$

For re-id, we adopt the Online Instance Matching (OIM) loss[14] as a supervision for training the re-identification task, denoted as L_{oim}.

$$L_{reid} = L_{oim} = -E_x\left[\log(p_t)\right], t = 1, 2, \ldots L. \tag{10}$$

Here p$_i$ as (11), τ use to smooth the probability distribution, 0.1 is used in our study.

$$p_i = \frac{\exp\left(v_j^T x/\tau\right)}{\sum_{j=1}^{L} \exp\left(v_j^T x/\tau\right) + \sum_{k=1}^{Q} \exp\left(u_k^T x/\tau\right)}. \tag{11}$$

The final loss function of the entire model consists of three components: the detector loss, the re-identification loss, and the QBGS module loss.

$$L = L_{det} + L_{reid} + L_{qbgs}. \tag{12}$$

4 Experimental Comparisons

4.1 Datasets and Evaluation

For our experiments, we use the two standard person search datasets, CUHK-SYSU [14], and Person Re-identification in the Wild (PRW) [9].

CUHK-SYSU comprises a mixture of imagery form hand-held cameras, and shots from movies and TV shows, resulting in significant visual diversity. It contains 18,184 scene images annotated with 96,143 person bounding boxes form known and unknown persons, with 8,432 known identities. The standard test retrieval partition for CUHK-SYSU dataset has 29, 00 query persons, with a gallery size of 100 scenes per query. **PRW** comprise video frames form six different cameras. It contains 11,816 scene images annotated with 43,110 person bounding boxes from unknown and known persons, with 932 known identities. The standard test retrieval partition for PRW dataset has 2, 057 query persons, and use all 6, 112 test scenes in the gallery, excluding the identity.

As in other works, we use the standard re-id metrics of mean average prediction (mAP), and top-1 accuracy (top-1). For detection metrics, we use recall and average prediction.

4.2 Experimental result

Table 1 presents the experimental results of our method on the CUHK-SYSU and PRW datasets. Compared to previous methods, our approach demonstrates improvements in both mAP and top-1 accuracy.

Table 1. Experimental result on two datasets

method	CUHK-SYSU		PRW	
	mAP	top-1	mAP	top-1
BINet [15]	90.0	90.7	45.3	81.7
NAE [16]	91.5	92.4	43.3	80.9
AlignPS [17]	93.1	93.4	45.9	81.9
SeqNet [11]	93.4	94.1	45.8	81.7
OIM + + [18]	93.1	93.9	46.8	83.9
MHGAM [19]	94.9	**95.9**	47.9	**88.0**
PSTR [1]	93.5	95.0	49.5	87.8
Ours	**95.5**	95.7	**51.3**	86.9

To further validate the effectiveness of the proposed Gallery Selector Module and MAIoU. Table 2 presents the ablation experiments conducted on the PRW dataset. For the selection of different MOB ratios, additional experiments were performed on the PRW dataset, as shown in Table 3. The baseline model employed the IoU along with QBGS module, further experiments were conducted on the PRW dataset to explore the selection of the γ in MAIF, as presented in Table 4.

5 Conclusions and Future Work

To address the issues of low gallery utilization and severe occlusion in supervised person search, we proposes a method for measuring anchor box similarity based on gallery selection and mask reward mechanism for the target person in supervised person search

Table 2. Ablation experiment on PRW

method	mAP	top-1
baseline	49.8	85.3
baseline + QBGS	50.6(↑0.8)	85.9(↑0.6)
baseline + QBGS + MAIoU	50.9(↑1.1)	86.5(↑1.2)
baseline + QBGS + MAIoU + MAIF	51.3(↑1.5)	86.9(↑1.6)

Table 3. Hyper parameter sensitivity experiments for different MOB. Default mask policy is top-down,* refers bottom-up

method	detection		Re-id	
	recall	AP	mAP	top-1
baseline	93.1	95.9	50.6	85.9
MOB=1.0	93.0	95.6	50.5	86.1
MOB=0.9	93.3	96.0	50.7	86.0
MOB=0.8	93.4	96.1	50.4	86.2
MOB=0.7	93.0	95.7	50.4	85.3
MOB=0.6	93.2	96.0	50.9	86.5
MOB=0.5	93.3	96.0	50.7	86.1
*MOB=0.6	92.8	95.5	50.4	85.5

Table 4. Hyper parameter sensitivity experiments for MAIF with different γ, baseline use MAIoU + QBGS with MOB = 0.6.

method	detection		Re-id	
	recall	AP	mAP	top-1
baseline	93.2	96.0	50.9	86.5
$\gamma = 0.1$	93.1	95.9	51.0	86.5
$\gamma = 0.2$	94.2	96.8	51.0	86.5
$\gamma = 0.3$	93.6	96.5	50.4	87.2
$\gamma = 0.4$	93.9	96.9	51.3	86.9
$\gamma = 0.5$	92.1	95.1	49.8	84.5
$\gamma = 1.0$	94.1	96.7	50.5	86.6
$\gamma = 2.0$	92.5	95.6	50.4	86.4

and improves the localization loss function for the detected anchor boxes. The query based gallery selector module can select the gallery for person search based on different threshold settings, resulting in different proportions of the gallery being selected. The setting of MOB ratio can be adjusted according to the actual occlusion issues in different datasets, aiming to better adapt to real-world scenarios. Experiments have demonstrated the effectiveness of the proposed methods mentioned above, proposed method can be easily transferred to numerous models that utilize Faster R-CNN as the detection backbone network.

Acknowledgment. This work was supported by Beijing Natural Science Foundation Project under Grant Z200002.

References

1. Cao, J., et al.: PSTR: end-to-end one-step person search with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9458–9467 (2022)
2. Chang, X., Huang, P.-Y., Shen, Y.-D., Liang, X., Yang, Y., Hauptmann, A.G.: RCAA: relational context-aware agents for person search. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11213, pp. 86–102. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01240-3_6
3. Chen, D., Zhang, S., Ouyang, W., Yang, J., Schiele, B.: Hierarchical online instance matching for person search. Proc. AAAI Conf. Artif. Intell. **34**(07), 10518–10525 (2020). https://doi.org/10.1609/aaai.v34i07.6623
4. Chen, D., Zhang, S., Ouyang, W., Yang, J., Tai, Y.: Person search via a mask-guided two-stream CNN model. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11211, pp. 764–781. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_45
5. Dong, W., Zhang, Z., Song, C., Tan, T.: Instance guided proposal network for person search, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2585–2594 (2020)
6. Han, C., et al.: Re-id driven localization refinement for person search. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9814–9823 (2019)
7. Lan, X., Zhu, X., Gong, S.: Person search by multi-scale matching. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV. LNCS, vol. 11205, pp. 553–569. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01246-5_33
8. Wang, C., Ma, B., Chang, H., Shan, S., Chen, X.: TCTS: a task-consistent two-stage framework for person search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11952–11961 (2020)
9. Zheng, L., Zhang, H., Sun, S., Chandraker, M., Yang, Y., Tian, Q.: Person re-identification in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1367–1376 (2017)
10. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks (2015)
11. Li, Z., Miao, D.: Sequential end-to-end network for efficient person search. Proc. AAAI Conf. Artif. Intell. **35**(3), 2011–2019 (2021). https://doi.org/10.1609/aaai.v35i3.16297
12. Oksuz, K., Cam, B.C., Kahraman, F., Baltaci, Z.S., Kalkan, S., Akbas, E.: Mask-aware IoU for Anchor Assignment in Real-Time Instance Segmentation (2021)
13. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)

14. Xiao, T., Li, S., Wang, B., Lin, L., Wang, X.: Joint detection and identification feature learning for person search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3415–3424 (2017)
15. Dong, W., Zhang, Z., Song, C., Tan, T.: Bi-directional interaction network for person search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2839–2848 (2020)
16. Chen, D., Zhang, S., Yang, J., Schiele, B.: Norm-aware embedding for efficient person search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12615–12624 (2020)
17. Yan, Y., et al.: Anchor-free person search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7690–7699 (2021)
18. Lee, S., Oh, Y., Baek, D., Lee, J., Ham, B.: Oimnet++: prototypical normalization and localization-aware learning for person search. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, l (eds.) Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X, pp. 621–637. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-20080-9_36
19. Li, Y., Xu, H., Bian, M., Xiao, J.: Cross-scale global attention feature pyramid network for person search. Image Vision Comput. **116**, 104332 (2021). https://doi.org/10.1016/j.imavis.2021.104332

A Privacy-Preserving Blockchain Scheme for the Reliable Exchange of IoT Data

Mnar Alnaghes[1](✉), Nickolas Falkner[1], and Hong Shen[1,2]

[1] University of Adelaide, Adelaide, South Australia, Australia
{Mnar.Alnaghes,Nickolas.falkner,Hong.shen}@adelaide.edu.au
[2] Macao Polytechnic University, Macao, China

Abstract. The Internet of Things (IoT) system has been claimed to deliver comfort and a more satisfactory lifestyle. The data flows that connect IoT sub-systems are critical to the success of the whole system. However, there are concerns about how privacy-preserving these flows are. Most existing single-server architecture solutions to the privacy problem have limitations regarding user privacy and anonymity. They may lead to revealing users' regular activities and affects their integrity and confidentiality. Despite Blockchain technology is suitable for improving IoT systems' security and privacy, there is a lack of privacy protection in blockchain when accessing personal user data due to privacy threats from internal parties. Thus, we aim to build a protected environment for users to exchange data and maintain privacy with an efficient authentication solution. We consider Blockchain technology using Elliptic Curve Integrated Encryption Scheme (ECIES) and message authentication regulation to enhance security and privacy for the transmitted data. This provides reliable auditing of the users' access history and efficient authentication within the system. We evaluate the outcomes to show the usefulness of this approach, targeting IoT data, and compare it with current work in the field.

Keywords: IoT · Blockchain · Privacy · Security · Encryption · Authentication

1 Introduction

The global demand for IoT devices has been growing due to the market demand for shorter and more efficient ways of converting real-world objects into smart *things*. For instance, there is global growth in the market for smart home and smart city technology [3], where the devices can be accessed and controlled remotely. An authorized user can control devices to perform tasks, regardless of the user's geographical location. Furthermore, IoT devices can be programmed to detect irregularities or typical conditions, trigger alerts or automated processes, and take action to prevent damage. However, the growth of IoT users has raised real concerns about data security and privacy. As IoT systems enable devices to collect and share personal user data with other devices to maintain user states across applications, they can introduce several flaws in user activity monitoring. Consequences include the exfiltration of sensitive data from these devices. The data from a single IoT device may not be considered sensitive or critical by itself, but collectively these data can be revealing.

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 260–271, 2024.
https://doi.org/10.1007/978-981-99-8211-0_24

Ultimately, IoT users have no control over their critical information and data, as there is no strict security model or admission control. They do not decide how it is going to be collected or processed. Thus, there is a need for a secure remote user authentication system that protects users' privacy rights and gives them the ability to control their transferred data. Similarly, IoT network elements, such as sensor nodes, gateways, and devices, should require and establish trust between each other before sharing data, using an authentication process. The proposed authentication process involves blockchain technology, as it is a publically existing and sufficiently lightweight solution that allows IoT devices to exchange data reliably and securely without exhausting the capabilities of resource-constrained devices.

Blockchain technology has many properties, such as decentralization, persistence, anonymity, and auditability, that can be practical in developing more secure and reliable IoT systems [12]. Blockchain solutions have improved users' data access and privacy due to their nature, allowing nodes to deliver services equally and assure availability [1]. This technology stores the users' data in blocks and prevents them from being modified. However, most current blockchain solutions are aimed at a single application scenario and cannot provide data transparency and auditability for all users' private data. The original data are stored and shared on the blockchain in plaintext in specific scenarios [9]. It lacks data privacy protection concerning user access policy while accessing private data in the IoT system. It suffers from problems such as privacy attacks. The access regulation is still plaintext, which can be collected and statistically analyzed to compromise users' privacy. Transferred IoT data must be secured to maintain its confidentiality and authenticity while obtaining the required permissions for accessing data in the network. We must be able to trace the originator of a malicious transaction, which would subvert security protocols if we were to block that transaction and preserve privacy.

As user access policies authorize and manage user permissions for accessing data, our goal in this work is to offer privacy preservation of users' access policy to protect users' data confidentiality and authenticity in IoT systems. We maintain users' privacy by preserving users' integrity approval before data transfer occurs within the system. To enhance the protection strength of the request transaction, we generate private and public keys using a secure hash function (SHF) [15]. We then stop the malicious actions possible from catching the key's values by use of a Key Derivation Function (KDF), for users' privacy protection. The main contributions of the work are:

- Defending against privacy attacks and providing reliable user data access auditing in IoT.
- Securing the transferred user data in an IoT network by guaranteeing its confidentiality and authenticity using ECIES encryption.

The rest of the paper is organized as follows: Section 2 clarifies the background, Sect. 3 discusses the related work, Sect. 4 explains the proposed scheme, Sect. 5 discusses the results of the proposed approach. Finally, Sect. 6 concludes the paper and suggests some future research directions.

2 Preliminaries

Blockchains are distributed decentralised digital ledgers that are both tamper-evident and tamper-resistant. They enable an arbitrary group of users to record transactions, and no published transactions can easily be modified after a certain number of network operations have elapsed [16]. A blockchain-integrated IoT system (BC-IoT system) is described as an IoT network that includes some blockchain components to conduct transactions. This section provides an overview of the background information on IoT and blockchain.

2.1 IoT Systems

The Internet of Things (IoT) [17] is the name given to a set of devices attached to the Internet or different communication networks that interact to exchange data among themselves. We can transform any object into an IoT device by adding hardware: usually sensors and some processing capability. For instance, an IoT camera senses the surrounding environment to track the traffic in crowded cities. Besides using IoT devices to make day-to-day life more manageable, IoT is used in many different domains and applications. For example, in the environment, smart sensors can help in fighting against climate change as they can observe water levels, safety-related events, and extreme weather conditions to predict a timetable for an event. Figure 1 shows IoT layers that consist of the sensing, network, processing, and application layers. The sensing layer manages IoT things within the system. In the network layer, IoT data is transmitted between the cloud and IoT devices through gateways. The data processing layer controls IoT levels for data within the system. The application layer is the interface between the IoT things and the network that handles the configuration and presentation of IoT data.

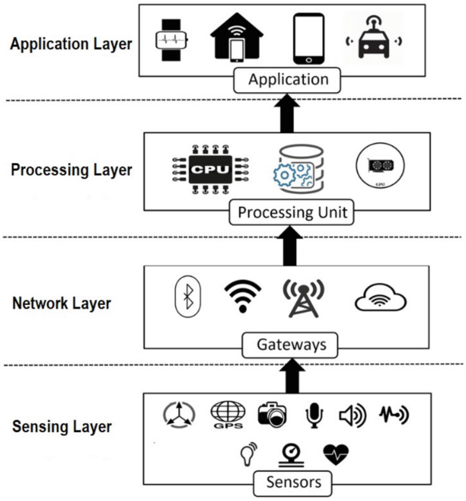


Fig. 1. The Structure of IoT Layers

2.2 Blockchain Technology

A blockchain is a decentralized ledger that securely records all transactions created on the blockchain system [12]. The ledger is shared among all distributed nodes who wish to participate. All nodes can check the ledger from the first transaction within the system until the most recent transaction. There is no centrally defined opacity as it is decentralized and not maintained or held by a central entity. Figure 2 shows the decomposed layers of a blockchain [19], which consists of the data, network consensus, ledger topology, and contract and application layers. The data layer conducts the data encapsulation function. The data generated from a blockchain application is verified and stored in a block. The block is connected to the previous one through the block's header and a hash value. The process results in an ordered chain of blocks duplicated among all nodes. The generated blocks are delivered to all nodes in the network layer. Blockchain is a peer-to-peer network where peers act as participants and offer storage for the distributed ledger of blocks. Consensus algorithms support preserving data integrity in the system. When a participant transmits a transaction, the data transaction is encrypted using a cryptographic algorithm before being confirmed by the other nodes to inspect if the transaction is valid [16]. If the majority of nodes approve the transaction, a new block is created in the chain [16]. The advantage of blockchain technology over traditional technology is that it permits two parties to execute encrypted transactions over the internet without the intervention of a third-party entity.

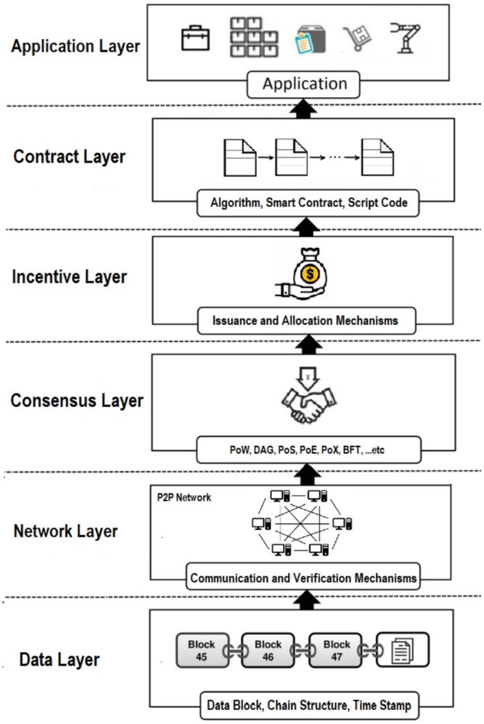

Application Layer
Application
Contract Layer
Algorithm, Smart Contract, Script Code
Incentive Layer
Issuance and Allocation Mechanisms
Consensus Layer
PoW, DAG, PoS, PoE, PoX, BFT, ...etc
Network Layer
P2P Network
Communication and Verification Mechanisms
Data Layer
Block 45
Block 46
Block 47
Data Block, Chain Structure, Time Stamp

Fig. 2. The Structure of Blockchain Layers

3 Related Work

Multiple studies focus on security and privacy for IoT [2–5], however, most of these studies did not consider the user's privacy in the process of data sharing in the systems. In [6], the idea is to introduce a secure data collection approach using homomorphic encryption on a blockchain, which uses an optimization algorithm. The authors focus on solving the privacy issue without considering the user's privacy when exchanging data with a service provider. In [8], the authors present a privacy-preserving medical data sharing scheme based on an encryption-using blockchain. However, their method affects the computation time by asking users to change the key whenever the transaction is updated, which is high cost for IoT systems. While in [10], the authors propose a distributed access control to protect sensitive data. However, it generates high transmission overheads and may uncover user personal details.

In [11], they represent an access control scheme that protects the system against different security attacks within the IoT networks. However, the keys generated for authenticating the transaction do not enhance the strength of the encrypted transactions. And, It shows high computation and storage overheads. To provide authentication and privacy in [12], The researchers in [12] proposed a Hyper-ledger Fabric-based access control framework combined with attributed-based access control (ABAC). It implements three types of smart contracts, namely device, policy, and access, to assure efficient access control under significant requests in the IoT environment. However, this scheme requires the user to establish a one-to-one key sharing channel, which inevitably increases the communication cost. The authors in [13] implement a blockchain-based model to improve the security of a publish/subscribe system. This model uses key encryption to ensure the confidentiality of the transmitted data. However, their method cannot stop cyber security attacks, compelling users to compromise their data protection. In [14], the authors proposed a hybrid neighbor scheme to manage the P2P network of blockchain, where a peer is elected into a leadership position to take charge of topology management. However, it uses bandwidth inefficiently, which increases the computational time as the network grows.

Blockchain-based technology adoption in IoT networks requires high computational time and energy consumption associated with blockchain [7]. This problem has formed the need to build an IoT system that provides time and energy-efficient services taking into account protecting users' data.

4 Proposed Scheme

Our method extends and improves the Lin et al. scheme in [9] by adjusting the encryption process. We focus on building a protected system for users to exchange data and maintain access policy privacy. We also aim to deliver resistance against various attacks, such as data leakage and loss, by providing reliable auditing for user access regulation. We use the modified ECIES algorithm [18] to encrypt transaction data requests and the SHF algorithm to generate all keys to stop malicious actions from catching the keys' values to enhance users' privacy preservation. Figure 3 illustrates our proposed scheme and Figure 4 illustrates the flowchart of our proposed scheme. Meanwhile, Algorithm 1 illustrates its Pseudo-code.

4.1 The Proposed Scheme Phases

The proposed scheme has five phases as follows:

Algorithm 1 The Pseudo-code of the Proposed Scheme

```
Obtain $\alpha^n \leftarrow \gamma^n$.
Calculate $h(x) \leftarrow x$, where $h(x)$ is the special hash function, obtained by message $x$.
Calculate $\delta_{private} \leftarrow h(x)$.
Calculate $\delta_q \leftarrow \delta_{private} * (E_{x_q}, E_{y_q})$, where $(E_{X_q}, E_{Y_q})$ corresponding to the $X_q$ and $Y_q$ coordinates of the point q.
Construct T $\leftarrow \delta_q$, where T is the transaction.
Sign T by GSign algorithm.
Encryption:
$C_{blockchain} \leftarrow Encrypt(T_{request}, \delta_q)$, where $C_{blockchain}$ is the encrypted access $T_{request}$ that is added to the blockchain system.
Authentication:
if Tag Match $(D_{target}, C) \leftarrow Decrypt(C_{blockchain}, \delta_{private})$, where $D_{target}$ is the target device and C is control information.
else Decline $C_{blockchain}$
end
```

- **System Setup:**
  - We invoke the setup algorithm for getting the keys to sign in and verify the transactions.
  - We take the security parameter γ^n to generate the public parameters α^n.
  - We generate a hash function $h(x)$ using SHA-256 to calculate private $\delta_{private}$ and public δ_q keys, avoiding calculating the private and public key from publically exposed points on the elliptic curve that can be easily compromise user privacy, which enhances the derived key security and ensures message confidentiality.

$$h(x) = H(x) \qquad where\ H : (0, 1)^* \rightarrow (0, 1)^{256} \tag{1}$$

  - Instead of selecting publicly exposed points on the elliptic curve randomly, we use secure hash values to obtain the private key using points on the elliptic curve that are tagged with a large prime value.

$$\delta_{private} = \upsilon(h) \qquad where\ \upsilon : (0, 1)^* \rightarrow (0, 1)^{ksize} \tag{2}$$

$$\delta_q = \delta_{private} * (Ex, Ey) \tag{3}$$

- Request Control:
  - We use the Key Derivation (KD) algorithm to obtain several keys from a single confirmed key to ensure that an attacker will not be able to identify its origins.
  - After generating the keys, we construct the request transaction and sign it by the group signing algorithm (GSign) [9]. Then, It will be encrypted and authenticated using the keys and MAC algorithm. The encryption procedure is provided by:

$$C_{blockchain} = Encrypt(T_{request}, \delta_q) \tag{4}$$

$C_{blockchain}$ defines the encrypted request transaction that is uploaded to the blockchain grid.

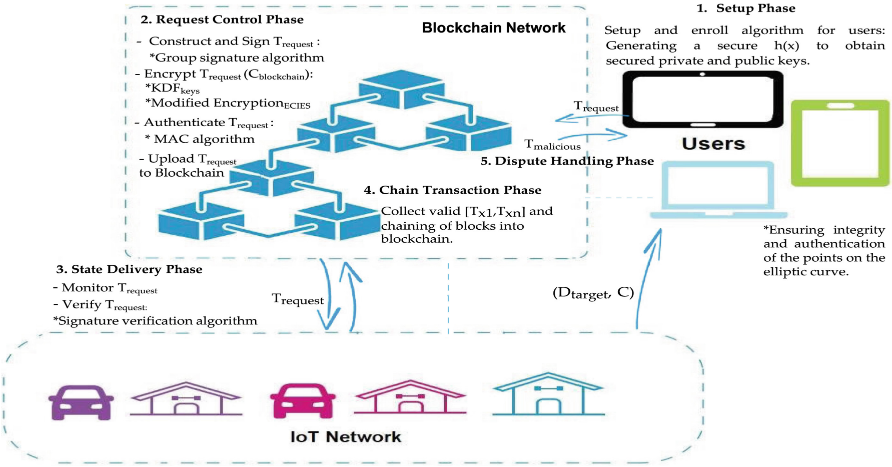

2. Request Control Phase
- Construct and Sign $T_{request}$:
*Group signature algorithm
- Encrypt $T_{request}$ ($C_{blockchain}$):
*KDF_{keys}
*Modified $Encryption_{ECIES}$
- Authenticate $T_{request}$:
* MAC algorithm
- Upload $T_{request}$ to Blockchain
Blockchain Network
1. Setup Phase
Setup and enroll algorithm for users: Generating a secure h(x) to obtain secured private and public keys.
$T_{request}$
$T_{malicious}$
Users
5. Dispute Handling Phase
4. Chain Transaction Phase
Collect valid [T_{x1},T_{xn}] and chaining of blocks into blockchain.
*Ensuring integrity and authentication of the points on the elliptic curve.
3. State Delivery Phase
- Monitor $T_{request}$
- Verify $T_{request:}$
*Signature verification algorithm
$T_{request}$
(D_{target}, C)
IoT Network

Fig. 3. The Proposed IoT-Blockchain Platform

- State Delivery:

  The participating nodes observe and verify the access request by the signature verification algorithm. The transaction is decrypted when it is valid, and the decryption procedure is provided by:

$$(D_{target}, C) = Decrypt(C_{blockchain}, \delta_{private}) \tag{5}$$

- Chain Transaction:

  The participating nodes in the blockchain are responsible for recovering transactions in the smart contracts where signatures are confirmed to check transaction validity. If it is valid, they are gathered, and the block is created [9]. The process is as follows:

  - Within the transaction grouping duration, collect all valid transactions [$Tx1, Txn$] and define invalid transactions to be illegitimate and discarded.
  - Using the Practical Byzantine Fault Tolerance, the participating nodes chain valid transactions into a block. Then, this block is chained into the blockchain when all the nodes reach an agreement.

  After that, the user access policy is adjusted to manage permissions set by the user.

- **Dispute Handling:**
  The abnormal behavior, such as frequent changes in the device's state, is detected for tracing unusual transactions *Txmalicious* . Then, using the GTrace algorithm [9], the transaction associated with abnormal behavior is retrieved. This step helps in stopping attacks.

4.2 The Proposed Scheme Computation Time

As computation time is affected by the number of users in direct proportion, we calculate it by omitting the variable that depends on the users' number. Thus, we calculate the initial time and the time of transaction generating and confirming. We calculate $T_{initial}$ by summing the time for generating the hash function T_{hash}, the time of calculating the public/private keys $T_{P\ keys}$, and the time for generating the public parameter $T\alpha$.

$$T_{initial} = T_{hash} + T_{Pkeys} + T_{\alpha} \tag{6}$$

While we calculate *Ttransaction* by summing the time of generating one transaction $TG(t)$, and the time of its verification by a node $TV\ (t)$:

$$T_{transaction} = \sum T_G(t) + \sum T_V(t) \tag{7}$$

Therefore, the computation time is calculated by:

$$T_{computation} = T_{transaction} + T_{initial} \tag{8}$$

5 Discussion and Analysis

This section compares our proposed scheme with the study in [9] as both methods are ECIES-based for protecting IoT transaction data and maintaining its confidentiality and privacy. We selected the MATLAB R2020a platform on a personal computer where the system configuration is Windows 10 with an Intel Core i7 to execute a prototype of our approach. It supplies plenty of cryptographic technologies producing protected and privacy-preserving applications. We used NIST P-256 to be able to randomly select the parameters for producing the elliptic curve's public parameter. We also used the SHA-256 hash function for generating the keys to enhance the correlation of the transactions and transferred data. We used a data set of four groups of machines from online resources, which are 45, 135, 225, and 450. We collected samples from these groups to create the transaction requests.

For evaluation, we consider characteristics like the *ID*, *T ype*, and *SerialNo.* of the device used for generating the transaction. For comparison, we took three samples; one from our results, one from [9], and one from the 50-device group dataset. These samples include the encrypted transactions in the request control phase. As we noticed a linear relationship between the elliptic curve points for the generated and encrypted transactions in the correlation analysis, we consider the correlation coefficient (ρ) between these

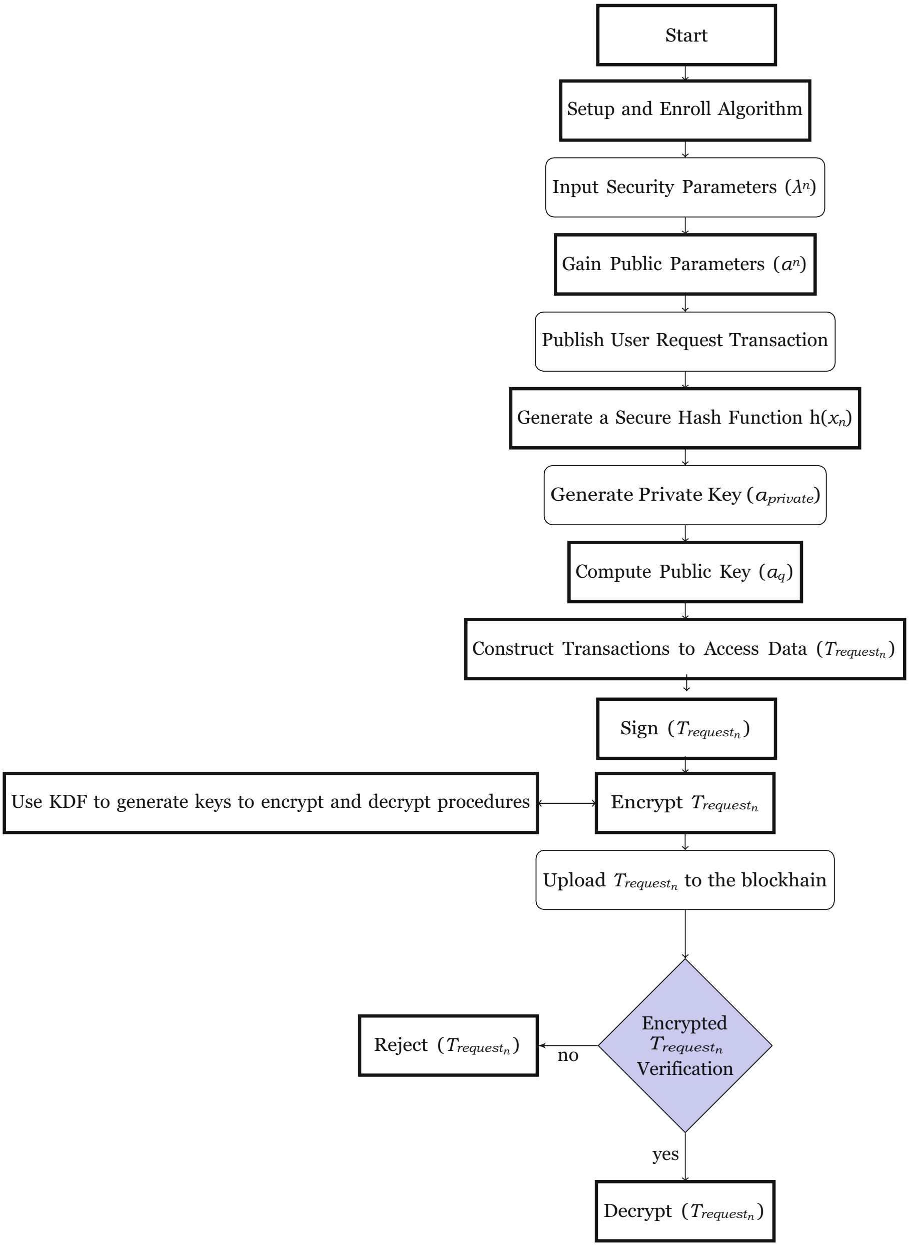

Start
Setup and Enroll Algorithm
Input Security Parameters (λ^n)
Gain Public Parameters (a^n)
Publish User Request Transaction
Generate a Secure Hash Function h(x_n)
Generate Private Key ($a_{private}$)
Compute Public Key (a_q)
Construct Transactions to Access Data ($T_{request_n}$)
Sign ($T_{request_n}$)
Use KDF to generate keys to encrypt and decrypt procedures
Encrypt $T_{request_n}$
Upload $T_{request_n}$ to the blockhain
Encrypted $T_{request_n}$ Verification
no
Reject ($T_{request_n}$)
yes
Decrypt ($T_{request_n}$)

Fig. 4. The Flowchart of the Proposed Scheme

transactions in our comparison. Table 1 shows the device ID samples with the generated transactions and the correlation coefficient values of the three samples. The value is

Table 1. ID Samples Including Created and Encrypted Transactions

No	ID Samples	Created Trans	Encrypt. Trans. of [9] Scheme	ρ [9]	Encrypt. Trans. of Our Scheme	$\rho ours$
1	2c450f2386	$<01\|pk1\|2c450f\ 2386\|o>$	mNgxrrlzztep	0.351	$<@N\ 2@5*/gh/sEj/M>$	0.310
2	2z473b0523	$<01\|pk2\|2z473b0523\|o>$	zShoiFnopsltr	0.360	$<35>JBIO@+\ *29/ba>$	0.312
3	2v897x0533	$<01\|pk3\|2v897x0533\|r>$	twchjkioAans	0.362	$<C5!(/78@' gmRb+523>$	0.313
4	4r633h2768	$<01\|pk4\|4r633h2768\|w>$	kGniopHcqts	0.366	$<+4!*@xo78?//@br>$	0.311
5	5j722g7358	$<01\|pk5\|5j722g7358\|r>$	VzBsirblemqxj	0.330	$<2bgh?+5f*63/' bL+>$	0.290

reduced from 0.351 to 0.310 in the first sample, which shows that the security stability of the encrypted transaction is enhanced.

Similar to ID characteristics, we also studied other characteristics, which are the device type and serial number. We took five samples from each dataset of 45, 135, 225, and 450 device groups. Table 2 shows the computation time and the correlation coefficient comparison of the device type characteristic whereas Table 3 shows the computation time comparison and the correlation coefficient of the device serial number characteristic of the three samples. The displayed results in Tables 2, 3 and 4 are before uploading the transactions into the blockchain network. It is shown that the proposed scheme enhances the correlation coefficient between the created and encrypted transactions and extends the security stability of the encrypted transaction. Furthermore, we computed the average computation time for the proposed scheme and [9], as shown in Figure 5 and Table 4. It demonstrates that the computation time is improved compared with [9].

Table 2. Time of Type Samples

Type Samples	[9] Scheme Time	Our Scheme Time	ρ [9]	$\rho ours$
Camera	97.71	92.70	0.34	0.28
Clock	95.20	89.81	0.33	0.29
Speaker	97.21	90.60	0.35	0.31
Television	104.3	96.5	0.33	0.28
Lamp	100.30	91.50	0.34	0.29

Table 3. Serial Number Samples

Serial Number Samples	[9] Scheme Time	Our Scheme Time	ρ [9]	$\rho ours$
72022180704001	103.8	97.3	0.35	0.31
72022180704002	107.2	99.6	0.34	0.29
72022180704003	109.6	102.7	0.36	0.32
72022180704004	105.14	99.0	0.34	0.28
72022180704005	99.3	96.9	0.33	0.29

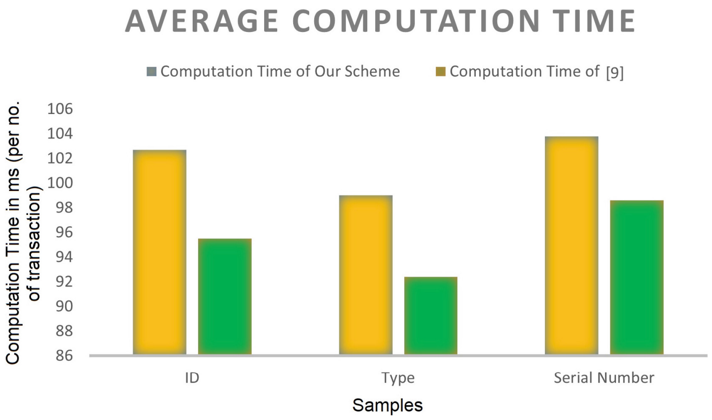


Fig. 5. Average Time of Our Scheme Compared with [9]

Table 4. The Average Results

Device Samples	[9] Scheme Time	Our Scheme Time	ρ [9]	$\rho ours$
ID	102.7	95.5	0.34	0.30
Type	99.0	92.4	0.35	0.31
Serial Number	103.8	98.6	0.34	0.30

6 Conclusion and Future Work

IoT systems can encounter serious issues related to data security and user privacy. In this study, we introduced a Blockchain-based integrated with ECIES and SHF scheme to maintain users' privacy and protect their data in IoT systems. This scheme delivers a protected environment that permits the access user to request and maintain the transaction and response messages. Our technique is a new element adopted by [9] to secure the authenticity and confidentiality of the transferred transaction and response messages. Our scheme decreases the computation duration by 7 ms per transaction, on average, when compared to [9]. For Future work, there is a need to investigate different cryptographic methods to build a protected platform for users to exchange data securely in an IoT system. We need to consider the issues related to analyzing and employing other approaches to incorporate within the blockchain system instead of only concentrating on assuring the authenticity and confidentiality of the transaction and response messages for performing more promising users' privacy in the IoT networks.

References

1. Andoni, M., et al.: Blockchain technology in the energy sector: a systematic review of challenges and opportunities. Renew. Sustain. Energy Rev. **100**, 143–174 (2019)

2. Liu, Y., Wang, K., Lin, Y., Xu, W.: LightChain: a lightweight blockchain system for industrial Internet of Things. IEEE Trans. Ind. Inform. **15**(6), 3571–3581 (2019). https://doi.org/10.1109/TII.2019.2904049
3. Uddin, M., Stranieri, A., Gondal, I., Balasubramanian, V.: A survey on the adoption of blockchain in IoT: challenges and solutions. Blockchain Res. Appl. **2**, 100006 (2021)
4. Kumar, P., et al.: PPSF: a privacy-preserving and secure framework using blockchain-based machine-learning for IoT-driven smart cities. IEEE Trans. Netw. Sci. Eng. **8**(3), 2326–2341 (2021)
5. Bowden, R., Keeler, H., Krzesinski, A., Taylor, P.: Block arrivals in the Bitcoin blockchain. CoRR abs/1801.07447. arXiv:1801.07447
6. Yan, X., Wu, Q., Sun, Y.: A homomorphic encryption and privacy protection method based on blockchain and edge computing. Wirel. Commun. Mob. Comput. **2020**, 1–9 (2020)
7. Islam, A., Shin, S.Y.: BUAV: a blockchain based secure UAV-assisted data acquisition scheme in Internet of Things. J. Commun. Netw. **21**(5), 491–502 (2019)
8. Jie, Xu., et al.: Healthchain: a blockchain-based privacy preserving scheme for large-scale health data. IEEE IoT J. **6**(5), 8770–8781 (2019)
9. Lin, C., He, D., Kumar, N., Huang, X., Vijayakumar, P., Choo, K.-K.: HomeChain: a blockchain-based secure mutual authentication system for smart homes. IEEE IoT J. **7**(2), 818–829 (2020)
10. Skarmeta, A., Hernandez-Ramos, J., Moreno, M.: A decentralized approach for security and privacy challenges in the Internet of Things. In: IEEE World Forum on Internet of Things (2014)
11. Ding, S., Cao, J., Li, C., Fan, K., Li, H.: A novel attribute-based access control scheme using Blockchain for IoT. IEEE Access **7**, 38431–38441 (2019)
12. Liu, H., Han, D., Li, D.: Fabric-IoT: a blockchain-based access control system in IoT. IEEE Access **8**, 18207–18218 (2020)
13. Lv, P., Wang, L., Zhu, H., Deng, W., Lize, Gu.: An IOT-oriented privacy-preserving publish/subscribe model over Blockchains. IEEE Access **7**, 41309–41314 (2019)
14. Baniata, H., Anaqreh, A., Kertesz, A.: DONS: dynamic optimized neighbor selection for smart blockchain networks. Fut. Gener. Comput. Syst. **130**, 75–90 (2022). https://doi.org/10.1016/j.future.2021.12.010
15. Gnatyuk, S., Kinzeryavyy, V., Kyrychenko, K., Yubuzova, K., Aleksander, M., Odarchenko, R.: Secure hash function constructing for future communication systems and networks. In: Zhengbing, Hu., Petoukhov, S.V., He, M. (eds.) Advances in Artificial Systems for Medicine and Education II, pp. 561–569. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-12082-5_51
16. Lim, M., Li, Y., Wang, C., Tseng, M.: A literature review of blockchain technology applications in supply chains: a comprehensive analysis of themes, methodologies and industries. Comput. Ind. Eng. **154**, 107133 (2021)
17. Stoyanova, M., Nikoloudakis, Y., Panagiotakis, S., Pallis, E., Markakis, E.: A survey on the Internet of Things (IoT) forensics: challenges, approaches, and open issues. IEEE Commun. **22**, 1191–1221 (2020)
18. Mihailescu, M.I., Nita, S.L.: Blockchain search using searchable encryption based on elliptic curves. In: Barolli, L., Hussain, F., Enokido, T. (eds.) AINA 2022. LNNS, vol. 451, pp. 471–481. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-99619-2_45
19. Nartey, C., et al.: Blockchain-IoT peer device storage optimization using an advanced time-variant multi-objective particle swarm optimization algorithm. EURASIP J. Wirel. Commun. Netw. **2022**(1), 1–27 (2021)

R-RPT-A Reliable Routing Protocol for Industrial Wireless Sensor Networks

Kripanita Roy and Myung-Kyun Kim(✉)

Department of Electrical, Electronic and Computer Engineering, University of Ulsan,
Daehak-Ro 93, Nam-Gu, Ulsan 44610, Republic of Korea
mkkim@ulsan.ac.kr

Abstract. Wireless Sensor Networks (WSN) are extensively used to monitor and control physical environments. Effective energy management and maintaining reliability are key considerations in Wireless Sensor Networks and routing plays a crucial role in achieving these objectives. The Routing Protocol for Low Power and Lossy Networks (RPL) is being adopted in Low-power and Lossy Networks (LLNs) to facilitate the connectivity of Wireless Sensor Networks within the Internet of Things (IoT). Although RPL has been significantly used in IoT routing, it still has extensive challenges. One of the most basic challenges is related to the reliability of routing. However, RPL lacks a load balancing mechanism, which is essential for maximizing the lifetime of sensor nodes by preventing the occurrence of overloaded nodes and the potential congestion that can result from it. To improve this issue, a new routing protocol was proposed called the reliable routing protocol (R-RPT) to maximize the reliability of data collection in large scale wireless sensor network. R-RPT aims to establish multiple bidirectional routes between a sensor node and a root node. R-RPT selects the parent node based on the evaluation of various criteria related to reliability. In addition, R-RPT achieves load balancing efficiently by sending data packets via the route with lighter workload. The simulation results obtained through Cooja simulator demonstrated that the proposed R-RPT routing protocol outperforms existing routing protocols in terms of packet delivery ratio, routing packet overhead and end-to-end packet delay.

Keywords: RPL · Wireless Sensor Network · Routing protocols · Reliability · energy efficient

1 Introduction

Wireless Sensor Networks (WSNs) have gained considerable attention in recent years as a cost-effective solution for monitoring and controlling various industrial processes. These networks consist of many small, low-power sensor nodes [1], that are capable of sensing and gathering data from the surrounding environment. In industrial settings, WSNs enable real-time monitoring of parameters such as temperature, humidity, pressure, and vibration, among others, providing valuable insights into the state of critical infrastructure and processes. Periodic traffic, characterized by the regular transmission

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 272–282, 2024.
https://doi.org/10.1007/978-981-99-8211-0_25

of data from sensor nodes at fixed intervals, is common in industrial WSNs. For instance, in a temperature monitoring application, sensors may periodically report temperature readings to a central control unit. However, the reliable routing of periodic traffic poses significant challenges due to factors such as network congestion, interference, node failures, and limited energy resources. Consequently, there is a requirement for a routing protocol that not only efficiently handles energy management to prolong node lifespan but also ensures reliable message transmission and reception.

In response to the critical nature of the challenge, the Internet Engineering Task Force (IETF) established the Routing Over Low-power and Lossy networks (ROLL) working group with the objective of developing a standardized routing protocol. As a result, the IPv6 Routing Protocol for Low-power and Lossy networks (RPL) was specifically designed to address the routing requirements of wireless sensor networks operating in environments characterized by low power and lossy conditions [2].

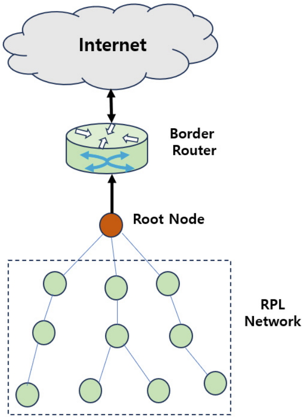


Fig. 1. System architecture of overall RPL network.

RPL is specifically designed to support to devices, whether hosts or routers, that operate with limited resources, as well as links that may be constrained or prone to loss. In Fig. 1, the overall architecture of the RPL based network is shown. The IPv6 border router is connected to the internet. Root node collects information about the RPL network such as node id, parent node id, all neighbors, and their ranks. RPL exceeds in rapidly establishing network routes and efficiently adapting to changing topologies as required. In IoT, each node collects the desired data according to its task and sends them to the root node. RPL is the only protocol that has been accepted as a routing standard for IoT. RPL is well adapted to the resource limitation of IoT devices and the specific environment of Low-power and Lossy networks (LLN). This protocol has created directed acyclic graph (DAGs) for connecting nodes with the root based on routing metrics. This protocol generates directed acyclic graphs (DAGs) to establish connections between nodes and the root based on routing metrics. It is worth mentioning that, depending on the applications, different routing metrics can be designed and used to construct the DAG graph, which is a key area of focus in research discussions.

While RPL effectively addresses the requirements of the IoT [3], it still faces several unresolved challenges, with one of the most crucial being the need for enhanced reliability support. In RPL, parents are responsible for constructing the communication topology in the form of a Destination-Oriented Directed Acyclic Graph (DODAG). The selection of parents is determined based on Objective Functions (OF). The Objective Function defines how RPL nodes translate one or more metrics into ranks, and how to select parents and optimize routes in a DODAG. RPL generally has two OFs called OF0 and ETX to select parents. In the case of OF0, the selection of the preferred parent is determined by either the rank or the minimum hop count, while in the case of ETX, it is based on the expected transmission count.

We emphasized two crucial elements of IoT networks, namely routing reliability, when taking into account this RPL problem. In this paper, an approach known as R-RPT was presented to address these problems. Based on the RPL protocol, R-RPT was developed to increase dependability by optimizing the OF and routing operations. The performance of R-RPT is generally divided into three main steps. In the first step, the state of network nodes and paths are evaluated in terms of reliability with the aim of improving routing and data exchange. In the second step, the DODAG graph is formed on R-RPT proposed, mechanism for predicting and preventing errors with the approach of controlling the selection of parents and leaf nodes. Implementation and performance evaluations of R-RPT are in comparison with previous studies based on Cooja simulator. The remainder of this research paper is structured as follows: related work of RPL protocol is provided in Sect. 2. In Sect. 3 provides an overview and explanation of the objective functions employed in the RPL protocol. The details of the proposed R-RPT methodology will be presented in Sect. 4. In Sect. 5, the proposed method based on Cooja software will be simulated and evaluated. Finally, we present the conclusion of our article in Sect. 6.

2 Related Work

The reliability of communication is a crucial aspect in Wireless Sensor Networks (WSNs), particularly for their application in industrial environments [4–6]. Wireless links, particularly in low-power wireless networks like WSNs, are prone to higher levels of unreliability compared to wired links. Factors such as interference, attenuation, and fading contribute to the increased susceptibility of wireless networks to disruptions in signal transmission. This unreliability poses a greater challenge in maintaining consistent and dependable communication within WSNs [7, 8].

As mentioned earlier, the Objective Function (OF) plays a crucial role in determining the construction of a Destination-Oriented Directed Acyclic Graph (DODAG) by guiding the selection of optimal parent nodes. Within the context of RPL, there exist two standardized Objective Functions: Objective OF0 [9], and MRHOF [10]. These OFs provide predefined criteria and algorithms for nodes to evaluate and choose the most suitable parent node, ensuring efficient and effective routing within the network.

OF0 was designed as the default OF for RPL. It selects the path to the nearest grounded root, creating a DODAG that can fulfill the application's requirements. MRHOF, one of the standardized Objective Functions (OFs), employs a parent selection strategy based on minimum path cost. To prevent oscillation and ensure stability in parent selection, MRHOF incorporates a mechanism called hysteresis. This mechanism introduces a threshold that dictates the conditions for a parent change. Specifically, the difference in path costs between the current parent and a potential alternative parent must exceed the specified threshold for a switch to occur. By implementing hysteresis, MRHOF aims to maintain a stable parent selection process within the network, avoiding frequent and unnecessary changes that could impact overall performance. But Parent selection in RPL is done by the OFs of OF0 or MRHOF. This way of selection is inefficient and does not cover the requirements of reliability. The effects of this issue will become more noticeable especially when the network traffic is increased and will lead to some important problems such as increasing errors, data loss, and loss of data exchange quality.

Ad hoc on-demand distance vector (AODV) [11], protocol uses many RREQ messages and one RREP message in the path-discovery process. The problem of this routing protocols is the number of request messages and the corresponding reply to messages, resulting in potential inefficiencies. Secondly, the current metric for path selection only considers the number of hops, disregarding other crucial factors like the remaining energy of nodes. To tackle these challenges, various proposals have emerged. Some studies suggest modifying the metric to incorporate additional link characteristics, giving them more significant weight alongside the number of hops. These efforts aim to create more balanced and effective routing protocols that consider multiple factors for optimal path selection. This protocol has only one metric, the number of hops.

3 RPL Objective and Functions

In this section, we discuss key concepts and features vital for understanding the operation of RPL. The fundamental topological structure in RPL is known as a DODAG (Destination-Oriented Directed Acyclic Graph), which originates from a designated node referred to as the DODAG root. The DODAG represents a directed acyclic graph structure that serves as a fundamental component within the RPL routing protocol. Each node in the DODAG is assigned a rank. It represents the location of a node within the DODAG. The rank strictly increases in the downstream direction of the DAG and decreases in the upstream direction. Each node identifies a stable set of parents on a path towards the DODAG root, and associates itself with a preferred parent, which is selected based on the Objective Function (OF).

3.1 Objective Function

The Objective Function (OF) functions as a path selection mechanism when nodes choose their parent node during the formation of the DODAG. It guides the decision-making process for selecting the appropriate parent nodes at each level, ensuring the construction of a cohesive and efficient DODAG [12]. The Objective Function (OF) operates by utilizing a routing metric as a reference and applying a specific algorithm to establish links within the DODAG. Its primary objective is to optimize the routing metric employed for forming links between sensors. By organizing the path selection process, the objective function aims to identify and determine the most optimal path, thereby facilitating the establishment of the best possible route within the network.

3.2 Minimum Rank with Hysteresis Objective Function (MRHOF)

The Minimum Rank with Hysteresis Objective Function (MRHOF) is an objective function that utilizes the minimum Expected Transmission Count (ETX) for selecting parent nodes. ETX represents the anticipated number of transmissions required for a packet to be successfully received at its intended destination. MRHOF leverages this metric to determine the optimal parent node selection, aiming to minimize the number of transmissions needed for reliable communication within the network.

3.3 Objective Function Zero (OF0)

Objective Function Zero (OF0) is an objective function that prioritizes the selection of parent nodes based on the minimum hop count to reach the root node. Each node calculates its rank by considering the number of hops required to reach the root node. Nodes with a lower hop count are assigned higher priority links according to OF0. By higher priority paths with fewer jumps, OF0 aims to establish more efficient and direct routes within the network.

4 Research Methodology

In this paper, we propose energy and load aware composite routing metric (R-RPT). R-RPT is a composite routing metric, and it is based on the combination of traffic load and expected transmission count (ETX).

1) Link ETX: The routing protocol based on ETX does not provide a guarantee for selecting optimal routes, resulting in frequent route failures under high load conditions. The continuous generation of RREQs during route discoveries can lead to a significant increase in traffic load within a specific timeframe The link ETX represents the efficiency of data delivery in both the forward and reverse directions of a link The forward data delivery (f_d) signifies the likelihood of a data packet reaching its intended recipient successfully. On the other hand, the reverse data delivery (f_r) denotes the probability of a successfully received ACK packet from the recipient. The link ETX [13], calculates from Eq. (1).

$$\mathrm{ETX} = \frac{1}{(f_d \times f_r)} \tag{1}$$

2) Data Traffic Load: Network data traffic refers to the amount of data being transferred across the network within a specific time frame. Load balancing is a technique employed to evenly distribute the network traffic across various nodes, primarily focusing on balancing the number of child nodes present in each parent node [14]. In R-RPT, Traffic Load of Path(p) calculation is based on the cumulative of node traffic or child set.

$$\text{Traffic\_Load\_Path }((p)) = \sum\nolimits_{Z=1}^{n} Node_{TrafficLoad}(Z) \tag{2}$$

In R-RPT, the traffic of a node is calculated based on the number of children it has in its parent node.

$$Node_{TrafficLoad}(Z) = \sum\nolimits_{Z=1}^{n} LeafNode\_count \tag{3}$$

The participant node chooses its parent node by considering the parent node with the least accumulated number of children within the DODAG. The traffic load is calculated using Eqs. (2) and (3). When the number of children in a parent node within the DODAG increases, R-RPT reconstructs the DODAG structure.

3) Rank Calculation: In R-RPT, the DODAG rank is determined by calculating the parent rank and adding the rank increase value. The rank increase value is derived from the step value and the metrics. The minimum rank among lower or same level node will become parent node. In R-RPT; we are considering ETX and data traffic in rank calculation. R-RPT start from root node with rank value 0. Data traffic defines the amount of packet transmitted within a given time slot. Route Request Packet (RREQ) send to its neighbour nodes. Node 0 sends RREQ (node_id, level, rank).

The receiver node will calculate its rank based on RREQ packet. The rank calculates from below Eq. (4)

$$Rank = SenderRank + \frac{ETX}{DataTraffic\_load} \tag{4}$$

As an example, a network with ETX value in each link with remaining data traffic load in each node is given as shown in Fig. 2.

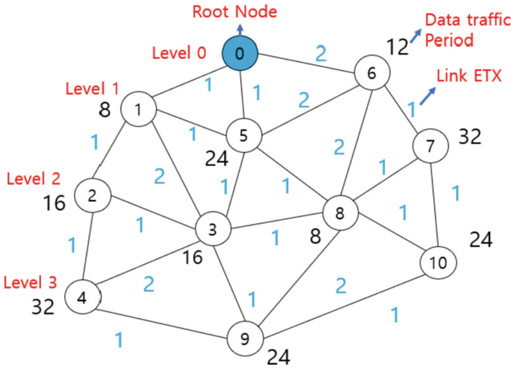


Fig. 2. Example network in initialization state

Figure 2, illustrates the initial state of the DODAG tree initialization. In this example, we have a DODAG consisting of 10 nodes, each node having a data traffic period and an associated ETX value. The root node starts with rank value of 0. Initially, the root node (0) broadcasts an RREQ message with node_id, level, rank [0,0,0]. Nodes 1, 5, and 6 receive the RREQ and calculate their ranks based on the routing metrics. Followed by, node 2,3,8,7 receive RREQ and perform rank calculation. After rank calculation, we will select the minimum rank node as parent. All sensor nodes send a node information packet to its parent node until it reaches the root node. Once the root node gathers all the node Information; it proceeds to formulate a schedule for data transmission.

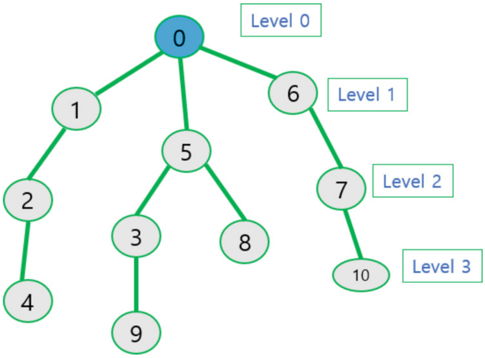


Fig. 3. Example network after selection parent for making routing state.

From Fig. 3, it is visible that by considering the routing metrics, we have successfully constructed a tree structure that ensures reliable end-to-end routes. The R-RPT protocol works to select parents based on the rank of nodes. If the value of a node's rank is smaller, the node is more suitable to be selected as the preferred parent based on multi-metric evaluation. After the construction of the DODAG, each sender node is assigned a predefined upward route to send its incoming traffic to the DODAG root. This default route is determined by selecting the most preferred parent for each node.

5 Performance Evaluation

We conducted the simulation of R-RPT, and it is an advancement of the standard version of RPL protocol. To configure the scenarios, a range of 10 to 200 T-mote sky were randomly distributed within the network's coverage area. It equipped with a CC2420 radio transmitter [15], and utilizing IEEE 802.15.4 MAC, achieved a data transfer rate of 250 kb/s. The network environment spanned an area of 300 m x 300 m, with a designated root node responsible for collecting data transmitted by the network members. The root node was positioned in the upper part of the middle area. Various scenarios were evaluated by considering variable network traffic levels ranging from 100 to 500 packets per minute (PPM) to assess the performance of R-RPT. Additional configuration parameters for the simulation scenarios can be found in Table 1.

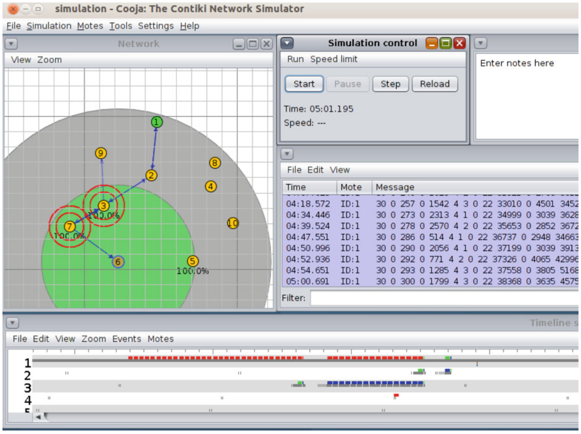
Performance of the proposed protocol has been evaluated through COOJA [16], a well-known simulator under Contiki Operating System. We have selected Contiki because it accurately emulates the behavior of real motes as the simulator uses the real code implemented on motes. Contiki is an open source, highly portable, multi-tasking operating system for memory efficient networked embedded systems and wireless sensor networks. Both interface and plug-ins can easily be added to COOJA, enabling users

Table 1. Simulation Parameters of R-RPT

Parameters	Value
Node Type	T-mote sky
Network Area	300 m × 300 m
Traffic rate	100–500 packets per minute (PPM)
Operating System	Contiki version (3.0)
Transmission range	80 m
Waiting time of RREQ packet	200 ms
Simulation Run time	3600 s

to quickly add custom function functionality for specific simulation. We compared the performance of the objective function of proposed protocol with AODV and MRHOF.

Our experiments deployed in a random topology, illustrated in Fig. 4, which enables nodes to directly reach the sink or establish contact with each other to reach the sink. This is particularly applicable to nodes located at the edges of the network. The topology utilized in our study comprises two distinct types of nodes. The node labelled as number 1 and depicted in green represents the Sink node. On the other hand, the non-sink (sender) nodes, depicted in yellow, have been randomly positioned within area.



```
Time: 05:01.195
Speed: ---
```

```
Time       Mote  Message
04:18.572  ID:1  30 0 257 0 1542 4 3 0 22 33010 0 4501 3452
04:34.446  ID:1  30 0 273 0 2313 4 1 0 22 34999 0 3039 3628
04:39.524  ID:1  30 0 278 0 2570 4 2 0 22 35653 0 2852 3672
04:47.551  ID:1  30 0 286 0 514 4 1 0 22 36737 0 2948 34663
04:50.996  ID:1  30 0 290 0 2056 4 1 0 22 37199 0 3039 3913
04:52.936  ID:1  30 0 292 0 771 4 2 0 22 37326 0 4065 42996
04:54.651  ID:1  30 0 293 0 1285 4 3 0 22 37558 0 3805 5168
05:00.691  ID:1  30 0 300 0 1799 4 3 0 22 38368 0 3635 4575
```

Fig. 4. COOJA Simulator windows with 10 Nodes.

For computation of link quality Packet Reception Rate (PRR) is used. In our simulation, Node 1 broadcasted 500 packets and the other nodes counted the number of packets successfully received. Figure 5, shows the packet delivery ratio in terms of the node density. As the node density increases, the proposed protocol consistently achieves a nearly 100% packet delivery ratio. In contrast, both the AODV and MRHOF protocols exhibit a slight decrease in packet delivery ratio with increasing node density.

The obtained result indicates that the proposed protocol reliably transmits data packets through a highly dependable path. In the case of MRHOF, the slight variation in the packet delivery ratio with respect to node density can be attributed to the presence of several weak links along the selected path of the protocol.

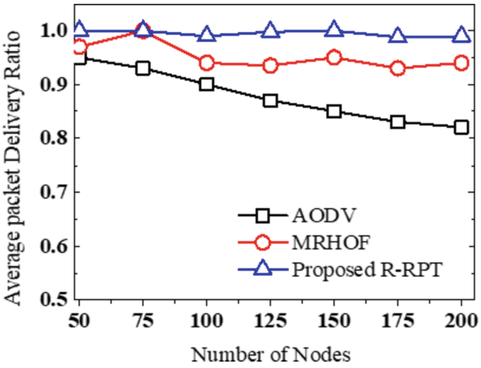
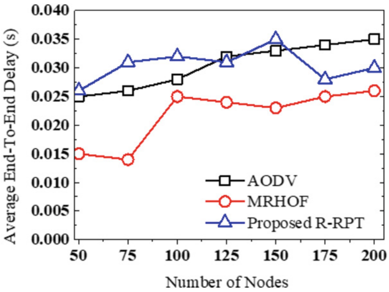


Fig. 5. Packet delivery ratio with respect to the number of nodes.

Figure 6, illustrates the average end-to-end delay. It is observed that the proposed protocol exhibits slightly higher delay compared to AODV and MR-HOF. This can be attributed to the fact that the proposed protocol occasionally selects a longer path with more hops in order to prioritize the selection of the most reliable path.



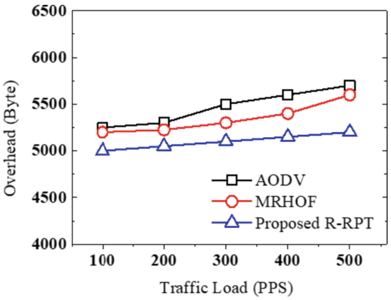


Fig. 6. End-to-End delay with increasing number of nodes.

Fig. 7. Control Overhead with respect to the traffic rate.

Figure 7, shows the results of control overheads under the influence of varying traffic loads. It is evident that as the traffic rate increases, there is a slight increase in control overheads. The increase in traffic has led to a higher impact of congestion, resulting in changes in traffic patterns. Consequently, there has been an increase in overheads required to effectively manage these changes. R-RPT effectively controls and manages traffic to ensure stability and guarantee reliable communications. By effectively mitigating the negative effects of congestion, R-RPT plays a crucial role in preventing the escalation of overheads caused by increased traffic. In contrast to other protocols, which are adversely affected by congestion, R-RPT effectively manages and controls these effects.

R-RPT has demonstrated superior performance in large-scale scenarios as compared to other protocols. In addition to constructing the network graph based on the most reliable nodes, R-RPT also takes into account the reliability of the routes. Overall, the combination of selecting reliable nodes and considering route reliability in R-RPT makes it particularly well-suited for larger-scale scenarios. The larger dimensions of the scenario pose greater challenges, such as increased distance between nodes, potential interference, and more complex network topologies. Its ability to handle these challenges results in a clear superiority over other protocols, leading to improved performance, stability, and reliability in communication networks.

6 Conclusion

In this paper, we proposed a routing protocol known as the Reliable Routing Protocol (R-RPT), specifically designed to maximize data collection reliability in large-scale wireless sensor networks. The primary objective of R-RPT is to enhance the reliability of data transmission and ensure robust communication throughout the network. Using the COOJA simulator, we conducted a performance comparison between R-RPT, AODV, and MRHOF. The objective was to select the route with the minimum value for the objective function, which typically corresponds to a shorter path with lower traffic. This chosen route is used for transmitting data to the DODAG root. By evaluating and comparing the performance of these protocols, we achieved insights into their effectiveness in route selection and data delivery within the simulated network environment. Based on the simulation results, it is evident that R-RPT outperforms AODV and MRHOF in terms of network lifetime, packet delivery ratio, and end-to-end delay. As part of our future work, we have planned to implement and evaluate the R-RPT protocol in Low Power and Lossy Networks (LLN) and deploy it within a real-time environment. By conducting these real-time deployments, we will explore to validate and optimize the R-RPT protocol's functionality, reliability, and efficiency in actual operational conditions.

Acknowledgements. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2022–00155933, IoT-based Intelligent Smart Dust Collection Platform for Electric Dust Collector).

References

1. Kharrufa, H., Al-Kashoash, H.A., Kemp, A.H.: RPL-based routing protocols in IoT applications: a review. IEEE Sens. J. **19**(15), 5952–5967 (2019)
2. Liu, X., Sheng, Z., Yin, C., Ali, F., Roggen, D.: Performance analysis of routing protocol for low power and lossy networks (RPL) in large scale networks. IEEE Internet Things J. **4**(6), 2172–2185 (2017)
3. Shahbakhsh, P., Ghafouri, S.H., Bardsiri, A.K.: RAARPL: end-to-end reliability-aware adaptive RPL routing protocol for internet of things. Int. J. Commun. Syst. **36**(6), e5445 (2023)

4. Dargie, W.: A quantitative measure of reliability for wireless sensor networks. IEEE Sens. Lett. **3**(8), 1–4 (2019). Art no. 7500904
5. Sulieman, N.I., Gitlin, R.D.: Ultra-reliable and energy efficient wireless sensor networks. In: 2018 IEEE 19th Wireless and Microwave Technology Conference (WAMICON), pp. 1–4. Sand Key, FL, USA (2018)
6. Sachan, S., Sharma, R., Sehgal, A.: An extensive learning on the reliability of mobile wireless sensor networks. In: 2021 5th International Conference on Information Systems and Computer Networks (ISCON), pp. 1–5. Mathura, India (2021)
7. Anastasi, G., Conti, M., Di Francesco, M.: A comprehensive analysis of the MAC unreliability problem in IEEE 802.15.4 wireless sensor networks. In: IEEE Transactions on Industrial Informatics, vol. 7, no. 1, pp. 52–65 (2011)
8. Zhao, J., Govindan, R.: Understanding packet delivery performance in dense wireless sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, pp. 1–13 (2003)
9. Thubert, P.: Objective function zero for the routing protocol for lowpower and lossy networks (RPL). Internet Engineering Task Force (IETF), Fremont, CA, USA, RFC 6552 (2012)
10. Gnawali, O., Levis, P.: The minimum rank with hysteresis objective function. Internet Engineering Task Force (IETF), Fremont, CA, USA, RFC 6719 (2012). Sorace, R.E., Reinhardt, V.S., Vaughn, S.A.: High-speed digital-to-RF converter. U.S. Patent 5 668 842, Sept. 16 (1997)
11. Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. In: Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications, pp. 90–100. LA, USA, New Orleans (1999)
12. Wang, Z., Zhang, L., Zheng, Z., Wang, J.: An optimized RPL protocol for wireless sensor networks. In: 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), pp. 294–299. Wuhan, China (2016)
13. Couto, D.S.J.D., Aguayo, D., Bicket, J., et al.: A high-throughput path metric for multi-hop wireless routing. Wireless Netw. **11**, 419–434 (2005)
14. Qasem, M., Al-Dubai, A., Romdhani, I., Ghaleb, B., Gharibi, W.: A new efficient objective function for routing in Internet of Things paradigm. In: 2016 IEEE Conference on Standards for Communications and Networking (CSCN), pp. 1–6. Berlin, Germany (2016)
15. Roy, K., Kim, M.-K.: Applying quantum search algorithm to select energy-efficient cluster heads in wireless sensor networks. Electronics **12**, 63 (2023)
16. Contiki: The Open Source OS for the Internet of Things. http://www.contiki-os.org

Action Segmentation Based on Encoder-Decoder and Global Timing Information

Yichao Liu[1], Yiyang Sun[1], Zhide Chen[1](✉), Chen Feng[1], and Kexin Zhu[2]

[1] College of Computer and Cyberspace Security, Fujian Normal University, Fuzhou 350007, China
zhidechen@fjnu.edu.cn

[2] Department of Information Engineering, Sun Yat-Sen University of Taiwan, Kaohsiung 80424, China

Abstract. Action segment has made significant progress, but segmenting and recognizing actions from untrimmed long videos remains a challenging problem. Most state-of-the-art (SOTA) methods focus on designing models based on temporal convolution. However, the limitations of modeling long-term temporal dependencies and the inflexibility of temporal convolutions restrict the potential of these models. To address the issue of over-segmentation in existing action segmentation algorithms, which leads to prediction errors and reduced segmentation quality, this paper proposes an action segmentation algorithm based on Encoder-Decoder and global temporal information. The action segmentation algorithm based on Encoder-Decoder and global timing information proposed in this paper uses the global timing information captured by LSTM to assist the Encoder-Decoder structure in judging the action segmentation point more accurately and, at the same time, suppress the excessive segmentation phenomenon caused by the Encoder-Decoder structure. The algorithm proposed in this paper achieves 93% frame accuracy on the constructed real Taiji action data set. The experimental results prove that this model can accurately and efficiently complete the long video action segmentation task.

Keywords: Encoder-Decoder · LSTM · Tai Chi · action segmentation

1 Introduction

In computer vision research, human movements range from simple limbs to complex whole-body movements. The longer the duration of the action sequence, the more complex the action recognition and prediction. Long-sequence actions bring two significant challenges to action recognition and prediction [13]: 1. The difficulty of feature extraction increases as long-sequence actions are rich in limb movements and require global consideration; 2. The network prediction capability is required to be high compared to simple actions such as waving and walking, as the actions in competitive sports are more complex and require neural networks with strong learning capabilities to predict complex actions. Long-time and complex human action recognition and prediction are still very challenging tasks in computer vision.

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 283–294, 2024.
https://doi.org/10.1007/978-981-99-8211-0_26

Tai Chi is a typical long sequence action with its long duration, high movement complexity, and wide range of motion. The sample used in this paper is 24-style simplified Tai Chi, also known as simplified taijiquan [12]. This paper proposes an Encoder-Decoder and global timing information-based action segmentation method. Our method uses a long and short-term memory network to capture the global timing information. Moreover, our method uses an Encoder-Decoder to identify and segment the actions in more detail, better performing in the authentic video segmentation task.

2 Related Works

The recognition task requires the overall classification of the video, while the segmentation task requires the classification of each video frame. For short trimmed videos (2–10 s), architectures used for action recognition include two-stream 2D CNNs [2], CNNs combined with LSTMs [3], 3D CNNs [4], and more recent architectures include two-stream 3D-CNN (I3D) [5].

Inspired by temporal convolution's success in speech synthesis, researchers have explored its application to temporal action segmentation tasks. The conventional framework for this is Ms-tcn [15], utilizing temporal convolution with increasing dilation to maintain constant resolution. Lea et al. [11] introduced a temporal convolutional network for action segmentation and detection using an encoder-decoder architecture, which includes temporal convolution and pooling in the encoder and up-sampling and deconvolution in the decoder. However, temporal pooling may lose fine-grained information needed for precise recognition. Lei and Todo-rovic [18] build on top of and use deformable convolution instead of standard convolution and add the residual stream to the encoder-decoder model. The encoder-decoder model is a widely used neural network model in Seq2Seq tasks and has demonstrated success in various applications such as machine translation [19], speech recognition [1], and image generation [20]. This paper utilizes a multi-layer convolutional neural network (CNN) to build an encoder for processing input sequences, employing a one-dimensional convolution kernel to extract local correlation features. This structure effectively solves the vanishing gradients problem. At the same time, LSTM is used as a global temporal information extractor and suppresses the over-segmentation phenomenon that the encoder-decoder structure may cause.

Due to the direct application of video action segmentation in real-life human activities, models with overconfident and incorrect predictions can lead to disastrous consequences. Among all available solutions to overcome overconfidence, probability ensemble is one of the most effective approaches [8]. In this paper, we use the probability ensemble approach to integrate the multilayer output of the decoder with the LSTM temporal information results and later perform the final classification.

The I3D model [14] extends 2D to 3D convolution and pooling kernels in a deep image classification network, seamlessly learning spatiotemporal features. Pre-trained on the Kinetics dataset, it enhances training data, making the model robust and generalizable. Many approaches, like TCN networks [9], RNNs, or attention modeling [10], build upon the I3D architecture to extract segment-level features and add sequence-level processing for recognition or segmentation. This paper's model also utilizes segment-level features from I3D and models temporal relationships based on these features.

3 Encoder-Decoder and Global Timing Information Model

The proposed model consists of three components: Encoder, Decoder, and LSTM. The encoder uses a convolutional structure with progressive down-sampling to extract high-level semantic information, enhancing robustness for different-resolution videos. A symmetric decoder, connected to the encoder for multi-scale feature fusion, progressively up-samples to reduce spatial location information loss. The LSTM layer captures global video timing information, smoothing codec results and suppressing over-segmentation. The decoder and LSTM outputs are probabilistically integrated for the final segmentation result, as illustrated in Fig. 1.

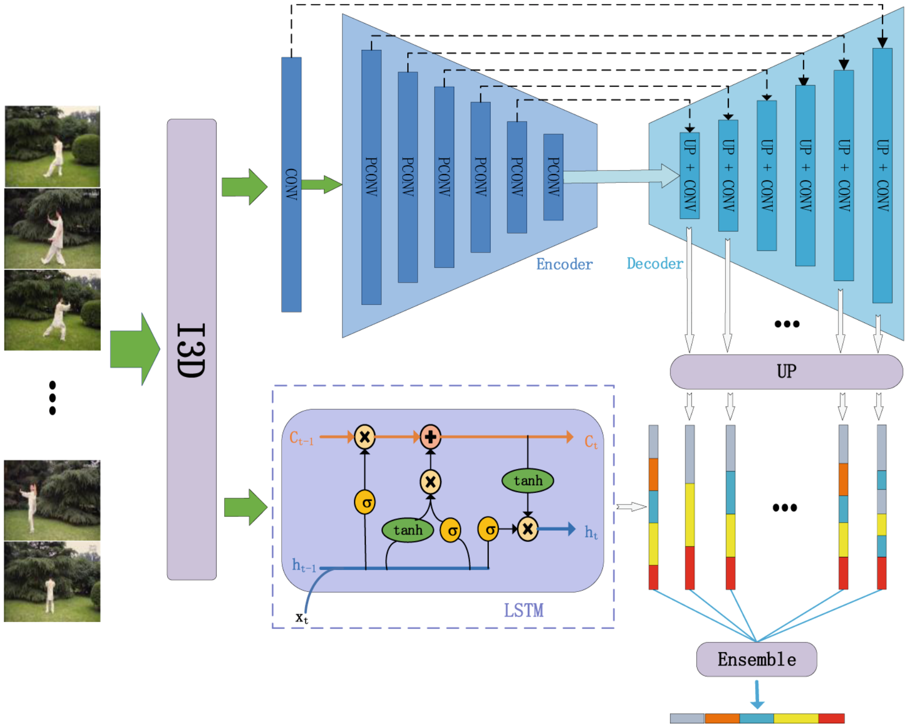


Fig. 1. Model overall structure.

The model's input is the feature sequence extracted by the I3D network in the original video. In this paper, we use 16 frames as a group, firstly extract the optical flow information, then convolve the optical flow information and the original RGB image information separately in 3D, and finally connect the convolution results to get 2048-dimensional feature sequences.

The following section describes each of the three main components of the model: Encoder, Decoder, and LSTM.

3.1 Encoder

The encoder part mainly consists of a one-layer convolutional structure ($CONV$) and a 6-layer $E_i : i \leq 6$ pooled convolutional structure ($Pool - CONV$, later referred to as $PCONV$) to reduce the size of the feature map to a lower dimensional representation. At the same time, as many low-level and high-level features are extracted as possible, so the extracted spatial and global information can be used to segment accurately.

Each $PCONV$ consists of a 1D maximum pooling layer ($Pool$) and a $CONV$ layer, which halves the temporal resolution of its input. The $CONV$ structure contains a convolutional layer, a batch normalization layer, and an activation layer.

3.2 Decoder

The decoder network is structurally symmetric to the encoder; it has six decoder layers $\{D_i : i \leq 6\}$; each decoder layer contains an up-sampling unit (UP) and a one-dimensional convolutional block ($Conv1D$). The up-sampling unit is interpolated using linear interpolation. For each $i \geq 1$, the up-sampling unit interpolates the input time step to twice its length, and the feature dimension remains constant.

It is then connected to $E_{(6-i)}$, a jump connection made to the output of the $(6-i)$-th encoder block. The output of the i-th decoder block thus has the same time and potential dimensions as $6-i$.

3.3 LSTM

The average length of a 24-style Tai Chi full-action video is around 6 min, a typical long sequence. Unlike commonly used datasets in action segmentation, such as GTEA [34], 50Salads [6], and Breakfast [7], the 24 Tai Chi movements have a fixed order, and therefore, the time sequence contains important global temporal information. In the problem of dealing with time sequences, LSTM and RNN are commonly used network structures, and LSTM alleviates the gradient disappearance and gradient explosion problems existing in RNN to some extent through the gate structure. This experiment uses LSTM with $Conv1D$ for global time series information capture. LSTM performs global timing information capture, and $Conv1D$ projects the information from LSTM into label space.

3.4 Probability Ensemble

The full-convolutional network structure is prone to over-segmentation errors, and a probability ensemble can effectively reduce over-segmentation. In this paper, we use the probability ensemble to synthesize the results. As shown in Fig. 1, each decoder layer $D_i : 1 <= i <= 6$ has the corresponding output vector $Y_i : 0 <= i <= 5$, and the LSTM corresponding output vector is L. The probability ensemble module integrates the decoder's output vector with the LSTM layer's output vector.

$$P^{ens} = [P_1, P_2, P_3, P_4, P_5, P_6, P_7] \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \end{bmatrix} \tag{1}$$

where P^{ens} represents the final probability vector sequence after integration, which is size $Tin \times N\_Class$. α_1 is the weight of the LSTM layer's output, $\alpha_i : 2 <= i <= 7$ represents the decoder D_{i-1} layer's output.

Finally, the category with the highest probability at the current moment is taken as the predicted label for each moment, calculated as shown in Eq. 2:

$$\hat{y}_t = argmaxP_t^{ens} \tag{2}$$

$\hat{y}_t$ denotes the classification label at moment t, and P_t^{ens} denotes the integration probability vector at moment t.

4 Experiment

4.1 Dataset and Implementation Details

In this paper, we construct a real Tai Chi dataset. The dataset is collected from 24 Tai Chi videos on the web, and each video contains all the stances or 12 stances of the 24 Tai Chi styles. Tai Chi is characterized by long duration, high movement complexity, and a wide range of motion, a typical long sequence of movements. In the action recognition task, the number of action categories is one of the challenge points, and the more action categories in the dataset, the greater the recognition difficulty. Therefore, we labeled the movements appearing in each video under the guidance of professionals. The original video frame rate is 30 fps, and we resample it to 16 fps, with 13 videos as the training set and 4 videos as the test set. In our experiments, we train using an Adam optimizer for 1000 epochs, with learning rates of 3×10^{-4}, weight decay of 3×10^{-4}, and batch size of 20.

4.2 Evaluation Metrics

This experiment uses three assessment metrics: *Accuracy*, *Segmental edit distance*, and $F1$.

Accuracy
Accuracy Is the average frame accuracy, calculated as the number of correctly classified frames divided by the total number of frames:

$$Acc = \frac{correct}{total} \tag{3}$$

corrent represents the number of correctly classified frames, *total* represents the number of incorrectly classified frames.

Segmental Edit Distance

Edit distance, also known as the *Levenshteindistance*, is commonly used in NLP to measure the degree of difference between two strings. The *Edit distance* is defined as the number of delete, insert, and replace operations required to transform string *a* into string *b*. The larger the *edit distance*, the greater the difference between the strings. The calculation is as follows.

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) if min(i,j) = 0, \\ \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & otherwise. \end{cases} \tag{4}$$

a, b represent two strings, and $lev_{a,b}(i,j)$ represents the distance between the first i characters in a and the first j characters in b. The *edit distance* between a, and b is the distance when $i = |a|, j = |b|$, i.e., $lev_{a,b}(|a|, |b|)$.

The *segmented edit distance* considers only one edit operation, i.e., the deletion of the error region. The cost is assigned to this edit operation according to the method used to perform the deletion. The *segmented edit distance* is defined as follows.

$$\mathrm{SED}(\mathrm{T}, \mathrm{G}, \theta) = \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{SED}_2(\mathrm{T}_\mathrm{i}, \mathrm{G}_\mathrm{i}, \theta) \tag{5}$$

T is the segment set of segments of the test set. G is the segment set of ground truth segments, and N is the number of segments in G.

IoU F1 Score IoU (Intersection-over-union) is the intersection part of two bounding boxes divided by their union, and the IoU ratio of the union determines the segment overlap threshold.

$F1$-score is the summed average of precision and recall

$$\mathrm{F1} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{6}$$

IoU-based $F1$ is an evaluation metric to evaluate the segmentation capability of the model as follows:

$$\mathrm{precision} = \frac{\mathrm{TP}_{\mathrm{IoU}>\mathrm{n}}}{\mathrm{TP}_{\mathrm{IoU}>\mathrm{n}} + \mathrm{FP}_{\mathrm{IoU}>\mathrm{n}}} \tag{7}$$

$$\mathrm{recall} = \frac{\mathrm{TP}_{\mathrm{IoU}>\mathrm{n}}}{\mathrm{TP}_{\mathrm{IoU}>\mathrm{n}} + \mathrm{FN}_{\mathrm{IoU}>\mathrm{n}}} \tag{8}$$

$$\mathrm{F1} = \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{9}$$

where n represents the threshold value of IoU.

4.3 Analysis of Results

In this experiment, we select four representative models with different architectures in related fields for comparison, including MSTCN [15], MSTCN++, BCN [16], and C2F-TCN [17]. The MSTCN model uses temporal convolutional neural networks (Temporal Convolutional Networks, TCN) to aggregate timing information and uses multi-layer TCNs for stacking. The (Dual Dilated Layer) DDL structure is proposed in the MSTCN++ model, which combines large and small receptive fields, optimizes the structural design of MSTCN, and achieves better results. The BCN model improves over-segmentation and ambiguous frame classification difficulties using an adaptive cascaded network and a temporal regularization method incorporating action boundary information. For the validity of the global timing information in this model, this paper will also compare it with the C2F-TCN model of the codec structure.

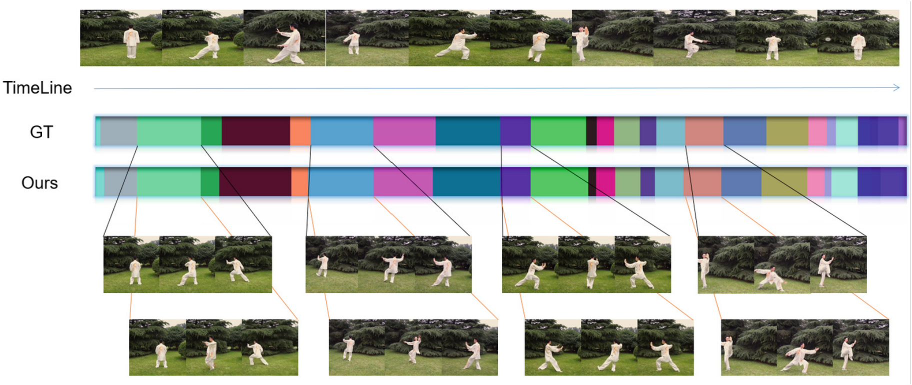


Fig. 2. Comparison of model segmentation effect and real segmentation point.

The comparison of the model segmentation results with the actual segmentation points is shown in Fig. 2, where different colors represent different actions. *GT* represents the real action periods, and *Ours* represents the action periods segmented by the model proposed in this paper. It can be intuitively observed that the model accurately determines the segmentation points of each action.

As shown in Table 1, our model outperforms the other methods in terms of $F1$ score, segmented edit distance, and average frame accuracy ($Accuracy$). Regarding the $F1$ score evaluation metrics, the $F1$ scores of the three different IOU thresholds are higher than the comparison models, and the accuracy and completeness rates are high. The most stringent $F1$ score among the $F1$ scores of the three IOU thresholds is $F1@50$. Our model's $F1@50$ evaluation score data has a maximum improvement of 16% and a minimum improvement of 13% compared with the convolutional model, and a 7% improvement compared with the C2F-TCN network with pure codec structure. The segmentation edit-distance evaluation metric can effectively measure the severity of the over-segmentation problem of the model, and there is at least a 2.5% improvement in this evaluation metric compared with other comparative models.

Table 1. Experimental results

model	F1@10	F1@25	F1@50	Edit	Acc
MSTCN	88.00	88.00	82.00	91.07	80.91
MSTCN + +	92.15	88.23	84.31	90.00	88.22
BCN	90.32	88.17	81.72	86.06	85.81
C2F-TCN	98.96	98.96	90.72	98.00	90.87
Ours	**100.00**	**100.00**	**97.91**	**100.00**	**93.35**

A more intuitive qualitative analysis of the segmentation effect of each model is given below.

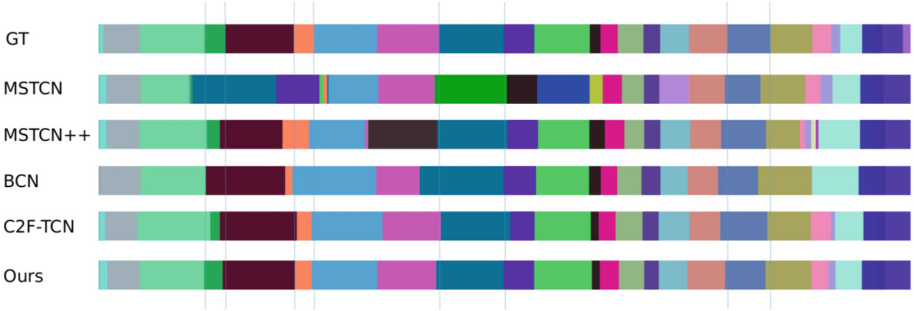


Fig. 3. Qualitative analysis of classification results.

Each color in the Fig. 3 indicates a sub-action, where *GT* indicates the true label. The segmentation fineness and stability of our methods are higher than other methods. Moreover, the MSTCN and MSTCN++ full convolutional structures show obvious classification errors and over-segmentation. The C2F-TCN structure is the codec structure, and the influence of the LSTM layer we add on top of the codec structure can obtain better classification results than the codec alone.

4.4 Analysis of results

Effect of Encoder-Decoder Probability Ensemble Weights
The probability ensemble is to multiply the output of each Encoder-Decoder layer and the LSTM layer output by the weight α, respectively, to obtain the integrated probability vector sequence.

The output of the decoder layer close to the encoder in the Encoder-Decoder structure is composite information containing much time-step information with a low temporal resolution. This output helps to mitigate the problem of over-segmentation of the model. On the other hand, the output of the decoder layer far from the encoder has a high temporal resolution, which facilitates the refinement of action segmentation, so a suitable combination of the two is needed.

In this section, we will investigate the impact of the probability vector of the output of the Encoder-Decoder part of the model on the final classification.

Table 2. Encoder-Decoder ablation experiment

$\alpha 1$	$\alpha 2$	$\alpha 3$	$\alpha 4$	$\alpha 5$	$\alpha 6$	$\alpha 7$	F1@10	F1@25	F1@50	Edit	ACC
1	1	0	0	0	0	0	82.35	82.35	78.09	74.50	83.28
1	1	1	0	0	0	0	91.66	87.50	81.25	83.75	89.24
1	1	1	1	0	0	0	96.90	94.84	90.72	94.07	91.51
1	1	1	1	1	0	0	**100**	**100**	**93.75**	**100**	**93.07**
1	1	1	1	1	1	0	98.96	96.90	88.65	98.00	92.04
1	1	1	1	1	1	1	98.94	98.94	92.63	97.91	91.72

α_1 is the weight of the LSTM layer output, $\alpha_i : 2 <= i <= 7$ represents the decoder D_{i-1} layer's output.

Table 2 shows that the highest $F1$, edit distance, and frame accuracy scores are obtained by decoders $D_i : 2 <= i <= 4$. It can also be found that the output of the decoder layers $D_i : 5 <= i <= 6$ with low temporal resolution plays a negative role in the final classification results.

Impact of LSTM Layer Global Timing Information

This section will discuss the global timing information obtained by the LSTM layer and the influence of the weights of the LSTM layer and the codec layer on the final classification.

Table 3. LSTM layer ablation experiment

$\alpha 1$	$\alpha 2$	$\alpha 3$	$\alpha 4$	$\alpha 5$	$\alpha 6$	$\alpha 7$	F1@10	F1@25	F1@50	Edit	ACC
0	1	1	1	1	0	0	98.96	96.90	87.65	98.00	92.34
1	1	1	1	1	0	0	**100**	**100**	**93.75**	**100**	**93.07**

α_1 is the weight of the LSTM layer's output, $\alpha_i : 2 <= i <= 7$ represents the decoder D_{i-1} layer's output.

Table 3 shows that with the addition of LSTM global timing information, all evaluation metrics are higher than the pure codec structure. Figure 4 shows the qualitative analysis of the LSTM layer ablation study. It can be seen from the figure that after adding LSTM, the segmentation boundaries are processed more accurately, and the codec action omission is eliminated at the same time.

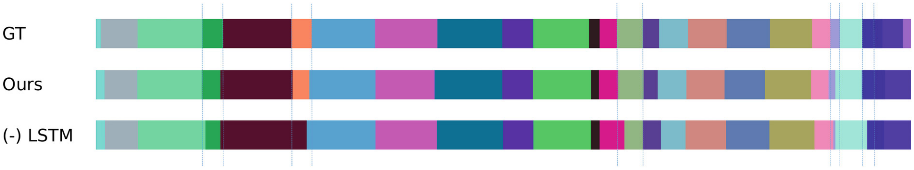


Fig. 4. LSTM ablation experiment qualitative analysis.

As can be seen from the Fig. 1, the decoder layer close to the encoder has low temporal resolution and does not help much in refining the segmentation. However, it is effective in reducing over-segmentation errors. In the following, we try to reduce the output weights of the decoder layer with lower resolution to observe the effect on the classification results and to give a combination of weights with optimal results.

Table 4. Combination of weights ablation experiment

$\alpha 1$	$\alpha 2$	$\alpha 3$	$\alpha 4$	$\alpha 5$	$\alpha 6$	$\alpha 7$	F1@10	F1@25	F1@50	Edit	ACC
3	3	3	2	1	0	0	97.95	97.95	87.75	96.15	92.36
2	3	2	2	1	0	0	**100**	**100**	**97.91**	**100**	**93.35**

The first row of data in the Table 4 shows the evaluation performance of the model after reducing the weight of the lower-resolution decoding layer, and the Fig. 5 shows its qualitative analysis. From the quantitative and qualitative analyses, it can be found that although the overall evaluation score of the model after reducing the lower resolution decoding layer is not low, it will be affected by a slight over-segmentation.

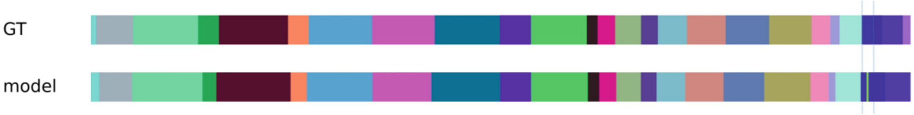


Fig. 5. Reduce the output weight qualitative analysis of the low-resolution decoding layer.

The second row of the table is the optimal weight combination given in this paper, increasing the weight of the output with the highest resolution of the decoder and appropriately decreasing the weight of the output with low resolution, the optimal result can be obtained.

5 Conclusion

This paper proposes an action segmentation model based on Encoder-Decoder and global timing information. Compared with the full convolutional structure, the model in this paper is less prone to over-segmentation errors. Furthermore, the global timing information captured by the LSTM can help the model produce more accurate and smoother classification results compared to the Encoder-Decoder-only structure.

In the long video action segmentation experiments, the proposed model in this paper showed better performance in terms of frame accuracy, F1 score, and edit distance compared to the classical models in the action segmentation field. Furthermore, experimental evaluation and ablation experiments demonstrate the proposed structure's effectiveness and superiority in long sequence problems like Tai Chi action videos.

Funding. This work is partial supported by National Natural Science Foundation of China (62277010), Science Foundation of Fujian Provincial Science and Technology Agency (2020J01132452), Fuzhou-Xiamen-Quanzhou National Independent Innovation Demonstration Zone Collaborative Innovation Platform Project (2022FX6), Fujian Provincial Health Commission Technology Plan Project (2021CXA001).

References

1. Zhang, Q., Lu, H., Sak, H., et al.: Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss. In: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7829–7833. IEEE (2020)
2. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1933–1941 (2016)
3. Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., Wook Baik, S.: Action recognition in video sequences using deep bi-directional lstm with cnn features. IEEE Access **6**, 1155–1166 (2017). 2
4. Donahue, J., et al.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2625–2634 (2015). 2
5. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308 (2017). 2, 6, 8
6. Stein, S., McKenna, S.J.: Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 729–738 (2013)
7. Kuehne, H., Arslan, A., Serre, T.: The language of actions: recovering the syntax and semantics of goal-directed human activities. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 780–787 (2014)
8. Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., Batra, D.: Why m heads are better than one: training a diverse ensemble of deep networks.arXiv preprint arXiv:1511.06314 (2015). 2, 4
9. Ding, L., Xu, C.: Weakly-supervised action segmentation with iterative soft boundary assignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6508–6516 (2018). 1, 2

10. Sener, F., Singhania, D., Yao, A.: Temporal aggregate representations for long-range video understanding. In: European Conference on Computer Vision, pp. 154–171. Springer (2020). Doi: https://doi.org/10.1007/978-3-030-58517-4_10
11. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
12. Sports Department of Peking University. Introduction to Type 24 Tai Chi. [2014–12–23]
13. Zhou, Y.: Research on Long Sequence Action Recognition and Prediction. Xiangtan University (2021). https://doi.org/10.27426/d.cnki.gxtdu.2021.001534
14. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724–4733 (2017)
15. Farha, Y.A., Gall, J.: Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3575–3584 (2019). 1, 2, 4, 5,7, 9
16. Wang, Z., et al.: Boundary-aware cascade networks for temporal action segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16. Springer, Cham (2020). Doi:https://doi.org/10.1007/978-3-030-58595-2_3
17. Singhania, D., Rahaman, R., Yao, A.: Coarse to fine multi-resolution temporal convolutional network. arXiv preprint arXiv:2105.10859 (2021)
18. Lei, P., Todorovic, S.: Temporal deformable residual networks for action segmentation in videos. In: Proceedings ofthe IEEE Conference on Computer vision and Pattern Recognition, pp. 6742–6751 (2018). 1, 2, 6, 7
19. Zhang, Z., Zhou, L., Ao, J., et al.: Speechut: bridging speech and text with hidden-unit for encoder-decoder based speech-text pre-training. arXiv preprint arXiv:2210.03730, 2022
20. Cao, S., Li, J., Nelson, K.P., et al.: Coupled VAE: improved accuracy and robustness of a variational autoencoder. Entropy **24**(3), 423 (2022)

Security Challenges and Lightweight Cryptography in IoT: Comparative Study and Testing Method for PRESENT-32bit Cipher

Van Nam Ngo[1], Anh Ngoc Le[2](✉), and Do-Hyeun Kim[3](✉)

[1] People's Police University of Technology and Logistics, Thuan Thanh, Vietnam
[2] Swinburne Vietnam, FPT University, Hanoi, Vietnam
ngocla2@fe.edu.vn
[3] Department of Computer Engineering, Jeju National University, Jejusi 63243, Republic of Korea
kimdh@jejunu.ac.kr

Abstract. The Internet of Things (IoT) stands out as one of the most remarkable innovations in recent times, offering a promising future for global connectivity. However, the rapid expansion of IoT ecosystems has led to a significant increase in the attack surface, posing risks to platforms, computing systems, multifunction protocols, and network access ubiquity. To mitigate these risks, it is crucial to adopt secure system design and development practices. Popular security solutions such as data encryption and authentication have been widely employed in IoT systems. Nonetheless, the unique constraints of IoT platforms present challenges in selecting suitable algorithms. In this paper, we provide an overview and analysis of the security challenges in IoT along with potential solutions. Additionally, we propose a testing methodology for the PRESENT-32bit cipher, based on an analysis of prevalent lightweight cryptography techniques. Our implementation results demonstrate the advantages of this approach.

Keywords: IoT · Hellman's method · lightweight cipher · cyber attack

1 Introduction

Internet of Thing (IoT) emerged in the early 2000s and has gradually played a significant role in various industrial sectors. Alongside AI, Big Data, and 3D printing, IoT's advantageous applications prevail in our world. The Internet of Things facilitates the interconnection of physical and virtual objects, enabling seamless communication between autonomous devices without human intervention. Additionally, IoT systems do not only involve direct communications between autonomous devices, which refers to the ability of the node to instantiate and exchange information with another node without human intervention, but also move data onto the internet for sensors, edge processors, and smart devices. It helps automate, keep track of all the processes, minimize costs, provide more services to the users, etc. Besides, the security issue is emphasized by

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 295–305, 2024.
https://doi.org/10.1007/978-981-99-8211-0_27

the lack of standards specifically designed for devices with limited resources, heterogeneous technologies. Mirai [1] – the most damaging denial of service attack in history and Stuxnet [2] - a nation-state cyber weapon targeting industrial SCADA IoT devices of Iran's nuclear program are two practical examples of consequences of the IoT's security issues.

The IoT security threat and challenges can be classified by an architectural view. There have been different architectures proposed for IoT environments such as three-layered, four-layered or five-layered architectures. In this paper, we provide overview and analysis from perspective of three main key layers of the IoT system model: perception/physical, network/transportation, application levels. Most of the papers, researchers choose this class [3]. The perception layer is related to the things or end-point devices: sensors, actuators, automobiles, RFID tags, etc. In the network layer, data transmission among the other layers is managed via different standards and protocols like 6LoWPAN, IPSec,.. Meanwhile, both application and services provided for users operate in the application layer. In this scenario the basic security requirement of the IoT is in integration of security in physical layer for data acquisition, in the transportation layer for data exchange and in application in order to guarantee the confidentiality, integrity and availability of data.

Many corresponding security solutions are provided for each layer. Cryptographic approach brings effective solutions to security problems of IoT systems. It appears in all stack levels: physical devices, communication systems, and networks. Encrypted data, which leaves a sensor and passes through gateways, mobile devices, and cloud systems, is not decrypted until viewed by an end-user. Each phase maintains a public key that is used to verify the authenticity of the next component loaded in the boot process. Trust Platform Module(TPM) is one particularly popular mechanism for key security. Cryptographic protocols are not dispensable from TLS and DTLS for MQTT and CoAP to network security. However, due to the many constraints of IoT devices, traditional cryptographic protocols are no longer suited to all IoT environments. So, various lightweight cryptographic algorithms and protocols have been proposed to secure data on IoT networks. There are some recent lightweight cryptographic protocols to secure IoT networks in resource-restricted systems such as LSC [4], PRESENT [5], SIMON/SPECK [6], AES-CCM/GCM [7], CRYPTREC,... It is so hard to choose appropriate lightweight cryptographic protocols to balance between security and performance. In our article, we proposed a protocol testing method for lightweight cipher by using Hellman's method TMTO [8].

The paper is organized as follows: Sect. 2 provides prevention corresponding methods and security solutions as well as their requirements. Section 3 discusses about lightweight cryptography in IoT environments and proposed testing approach with experiment results on Present-32bit. Finally, Sect. 4 concludes the paper.

2 Countermeasures and Security Solutions for IoT Systems

IoT requires security measures across three layers: the perception layer for data collection, the network layer for routing and transmission, and the application layer to ensure confidentiality, authentication, and data integrity. Establishing a secure IoT environment requires the meticulous integration of robust security practices throughout the entire development and operational lifecycle of devices. There are many mechanisms to ensure system safety such as: software running on IoT must be licensed; devices participating in the network need to authenticate themselves first to start transmitting data; setting up the firewall in the IoT network to filter packets sent to device, controlling traffic to ensure optimal use of capabilities limited handling; the device must receive patches and software updates in a way does not consume too much energy or compromise the safety of the device.

2.1 A. Type of Countermeasures to Attacks and Security Solutions in IoT systems

Authentication and Identity Management
The application scope of IoT is diverse in many fields, ranging from smart homes and smart cities to wearable devices, electronic health. Remarkably, it can even connect a number of billions of devices (forecasts indicate that by 2030, the average person will possess around 15 connected devices) [27]. The network consists of a large number of authenticated IoT devices that exchange information with each other. Authentication is the mainstay of the IoT network because all components undergo an authentication process before establishing communication. Communication on an IoT network can be person-to-machine, machine-to-machine and person-to-person. Nevertheless, traditional authentication methods can not be used directly in IoT networks due to the limitations of computing power and storage capacity. Consequently, addressing authentication challenges is crucial, especially when multiple users and devices require mutual authentication, emphasizing the importance of devising appropriate authentication management strategies [28].

Privacy and Access Control
Privacy emerges as a critical concern in IoT security, primarily due to the ubiquitous nature of the IoT environment. Data is exchanged over the internet, rendering user privacy a sensitive object. Personal information such as health data and travel schedules collected by IoT devices, as well as their sharing and management within the system, are also security issues that warrant thorough study. Such sensitive data within the IoT ecosystem could serve as an open invitation for attackers to exploit them in various ways. Access control means controlling access to resources by granting or denying them based on diverse set of criteria. Authorization is typically employed to enforce access controls effectively, fostering a secure connection between multiple devices and services. The main issue to be addressed in this scenario is making access control rules easier to create, understand and manipulate. There are some papers about those solutions as [29, 30, 31].

Intrusion Detection System
Intrusion is an unnecessary or malicious activity that is dangerous to sensor nodes. An Intrusion Detection System (IDS) [32] is used to monitor and detect malicious traffic within a network. IDS can be implemented as either software or hardware tools. It scrutinizes and investigates machine and user actions, identifying signatures of well-known attacks and categorizing malicious network activity. The IDS works as an alarm

or network observer, it avoids system damage by generating alerts before attackers initiate an attack.

Encryption

Security at the endpoints between IoT devices and Internet hosts is a matter of great importance. In the IoT world, an immense volume of raw data is continuously collected, necessitating real-time sensor data streams and techniques to transform this raw data into usable knowledge. Merely applying cryptographic schemes for encryption and authentication codes to packets is insufficient for resource-constrained IoT environments. End-to-end security in IoT ensures that both communication endpoints can confidently rely on the fact that their communication remains invisible to unauthorized parties, and data in transit remains unaltered. Some illustrative examples include [33] and [34].

2.2 Requirements and Criteria for Security Solutions in IoT Systems

Lightweight Solution.

Manufacturers typically produce most IoT equipment with low memory, computation capabilities, communication bandwidths, and power supplies. However, classical security algorithms do not perform optimally on IoT devices with such limited capabilities. The resource constraints posed by IoT devices present a significant challenge, necessitating the design of lightweight algorithms to ensure data confidentiality and integrity in the IoT environment, as well as to support real-time fog-based IoT services. Consequently, there is a demand for security systems that offer lightweight solutions.

Heterogeneity

IoT systems are diverse and interconnected through vast networks, resulting in variations in computing capabilities for running encryption algorithms. Constantly incorporating newly developed hardware platforms into IoT operating systems is essential. The proliferation of numerous IoT operating systems and the presence of heterogeneous devices present a challenge known as interoperability. Ensuring interoperability among security protocols implemented at different layers becomes critical for standardizing an effective IoT security mechanism. Likewise, achieving interoperability in access policies for multiple users and organizations poses a challenge for access control models. Moreover, IoT/IIoT devices are manufactured by various vendors, each adhering to its own set of standards, leading to conflicts when attempting to secure such devices. To address these challenges, a comprehensive and standardized IoT/IIoT framework must be established, integrating data models, ontologies, and data formats with protocols, applications, and services to ensure the interoperability and integrity of IoT/IIoT mechanisms, applications, and services.

Standardization

In case one of the things fails and stops sending data, it is necessary to discover another thing that can provide a similar set of data. IoT devices are not regularly updated due to their large number and geographical location so a universal standard is needed for security solutions to be deployed and run on all old and new devices.

3 Cryptography Deployments in the IoT

As per IoT reference architecture, IoT security has five functional components, identity management, authentication, authorization, key exchange and administration, trust, and reputation. Cryptographic primitives play a crucial role in achieving these objectives. There are two types of encryption algorithms which are known as asymmetric and symmetric key algorithms. The data encryption algorithms based on asymmetric keys such as RSA and ECC, offer a high level of security. But they are not preferable for IoT and WSN devices due to their limited resources like processing power, memory, and storage. Besides, data encryption algorithms based on symmetric keys like DES and AES, although they require less computational power and storage, still necessitate key exchange schemes or pre-stored keys. Symmetric ciphers serve mainly for message integrity checks, entity authentication, and encryption, whereas asymmetric ciphers additionally provide key-management advantages and nonrepudiation. However, IoT devices are characterized by limited computational power, limited memory, limited power supply, and limited battery life. These underscore the need to develop Lightweight Cryptographic (LWC) algorithms to ensure information security effectively.

These include algorithms that are fast, responsive, more energy and storage efficient than conventional encryption and decryption algorithms, and powered by optimized crypto engines. They are equipped with optimized crypto engines and typically exhibit one or more advantageous features, including minimal hardware implementation size, low computational demands for microprocessors or microcontrollers, cost-effectiveness, and robust security. It is difficult for a trade-off between security, cost, and performance in cryptographic algorithms, the key length is correlated with security and cost tradeoff, while the number of rounds in encryption provides a security, performance trade-off, and hardware architecture. It's generally easy to optimize any two of the three design goals security and cost, security and performance, or cost and performance; however, it is hard to optimize all three design goals simultaneously.

In 2013, NIST initiated a lightweight cryptography project to study the performance of the current NIST-approved cryptographic standards on constrained devices and to understand the need for dedicated lightweight cryptography standards, and if the need is identified, to design a transparent process for standardization. NIST held two Lightweight Cryptography Workshops in Gaithersburg, MD, to solicit public feedback on the constraints and limitations of the target devices, requirements and characteristics of real-world applications of lightweight cryptography [35]. In 2018, NIST published the submission requirements and evaluation criteria for the LWC competition. After the call, 57 algorithms were submitted to the competition. The NIST eliminated one of the algorithms from these submissions as it did not fulfill the requirements. After the first round, 32 algorithms were advanced to the second round. Subsequently, the third round of the competition began in 2021 after NIST announced the finalists of the competition.

Several research papers [36] have already established a comparison comparing software and hardware implementations of LWC algorithms.

3.1 Protocol Testing Method

Present-32 [5] is an ultra-lightweight cryptographic algorithm designed from AES and replaces eight S-boxes with just one (Fig. 1). Present is one of the top 10 algorithms that efficiently use memory, hardware, power consumption and small block and key size [37].

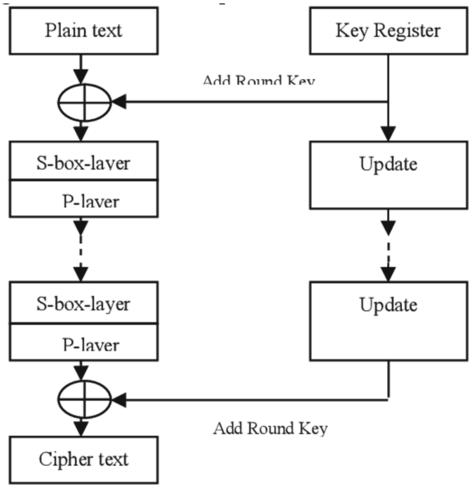


Fig. 1. The scheme of Present Cipher

Since its establishment, Present Cipher has been analyzed by researchers to evaluate the performance and security of several projects, including [38, 39, 40], and [41]. We employ a cryptanalytic time-memory trade-off method called Hellman and its modifications (DP and Rainbow) to attack the block cipher Present -32 bit. This algorithm enables the cryptanalysis of any N-key symmetric cryptosystem in $O(N^{2/3})$ operations with $O(N^{2/3})$ storage, provided a precomputation of O(N) is performed beforehand. Algorithm involves creating a table from the beginning to the end of the encrypted string, sorted by the end point to expedite the search for the key. If the key has not appeared yet, we consider another series. Theoretically, for a cryptosystem with a key space of N, we need to create a table of $mt = N$, where m is the number of the strings, and t is the length of each string. However, there are some challenges related to circular loops or two intersecting chains reduce the probability of finding the key. Therefore, we need to compute the evaluation of the relationship between m and t to determine the optimal number of strings and string length for the best performance. In addition, we tested other methods developed based on Hellman's algorithm, such as DP and RB methods.

Method Hellman for block ciphers includes preliminary and operational stage. In first stage, we make a table with m rows and 2 columns (created from metrix Hellman).

Formula of block ciphers like $C = E_k(P)$, where P – plaintext, C – ciphertext withlength n, k – key with length l. Besides, we use reductional functions f: $V_n \to V_l$: $f(k) = R(E_k(P))$.

In preliminary stage, we implement the following calculations to create a Hellman matrice[8]:

$$SP_1 = X_{1,1} \to^f X_{1,2} \to^f X_{1,3} \to^f \ldots. \to^f X_{1,t-1} \to^f X_{1,t} = EP_1$$

$$SP_2 = X_{2,1} \to^f X_{2,2} \to^f X_{2,3} \to^f \ldots. \to^f X_{2,t-1} \to^f X_{2,t} = EP_2$$

$$SP_m = X_{\mathrm{m},1} \to^f X_{m,2} \to^f X_{\mathrm{m},3} \to^f \ldots. \to^f X_{\mathrm{m},t-1} \to^f X_{\mathrm{m},t} = EP_\mathrm{m}$$

where $X_{i,j} = f\left(X_{i-1,j-1}\right) = R\left(E_{X_{i-1,j-1}}(P)\right), i \in \overline{1,m}, j \in \overline{1,t}$

Randomly choose m starting points $SP_1, SP_2, SP_3, \ldots, SP_m$ from the set of possible keys. Create m chains of length t. Only pairs (SP_i, EP_i) $i \in \overline{1,m}$ will be stored in memory, sorted byEP_i.

$$X_0 = C, X_1 = E_{X_0}(P), X_2 = E_{X_1}(P), \ldots$$

In operation stage, if $\exists i \in \overline{1,t}, X_i = EP_j j \in \overline{1,m}$, C belongs to the j-th step. The key k lies in front of C on the j-th node of the chain, so

$$Y_0 = SP_j, Y_1 = E_{Y_0}(P), \ldots$$

Find $C = Y_{t-i} = E_{Y_{t-i-1}}(P)$, then the key $k = Y_{t-i-1}$ was found.

For efficient search, it is proposed to use several Hellman matrices with different functions of the form $f_i = R_i(f(x))$, R_i modificational function of f.

3.2 Testing Result

We evaluate and compare the memory and time optimization as well as the probability of success of several methods in Table 1.

Table 1. Comparison table of cryptanalysis algorithms with Present cipher

Algorithm	Time complexibility	Data complexibility	Attack rounds	Possibility	Reference
Biclique	$2^{79.63}$	2^{23}	32		[39]
Correlation Power		2^{56}	32	0.8	[40]
Differential	2^{64}	2^{64} KP	16	0.99	[41]
Multidimensional linear	2^{72}	2^{64} KP	26	0.95	[42]
Hellman	2^{22}	2^{27}	32	0.61	Our result

Table 2. Correlation between string quantity and string length

Length t		100	200	500	1000	5000	10000
Chain numbers (log2)	HM	19.98	19.97	19.91	19.83	19.34	18.88
	DP	14.17	14.46	14.46	14.49	14.49	14.47
	RB	19.98	19.96	19.81	19.83	19.31	18.83

We calculated the sequence number from 220 keys, increasing the length from 100 to 10000. Our test was implemented in PC Core-i5 6300HQ RAM 8 GB.The calculation result is (Table 2).

If the string is too long, it will take longer to find the key, and it is easy to cycle or cross between two strings with different starting points.

Table 3 describes the correlation between chain length and properties of DP.

Table 3. The average length of the initial 1000-point sequence.

d	tmin - tmax	Theory	Experiment	Chain number
10	8–12	10.21	10.18	891
11	9–13	11.21	12.38	738
12	10–14	12.21	12.36	510
13	11–15	13.21	13.28	299

Table 4. Result in 1000 experiences with 3 methods

Methods	Memory	Time attack	Successful possibility
Brute-force attack	64 GB		1
Hellman attack	150 MB	4 min	0.62
Rainbow attack	163 MB	5 min	0.64
DP attack	42 MB	1 min	0.61

The larger the string length, the more strings with repeated endpoints. Specifically, the average length increased from 210 to 213, the number of chains decreased nearly 3 times (Table 2).

Through the calculation and analysis process, we selected m, t, r, as shown below, and compared three algorithms in terms of memory, time, and success probability.

Hellman's method: m = 2^16, t = 2^12, r = 104.

DP method: m = 2^16, t = 2^16, r = 426, d = 12.

RM method: m = 2^16, t = 2^12, r = 139.

4 Conclusion

The main goal of this paper was to provide an explicit survey of the most significant aspects of IoT, with particular focus on the security challenges involved in the Internet of Things. Furthermore, we assess the necessity for lightweight cryptography in the IoT networks and propose a testing method for lightweight cipher PRESENT-32bit. We conducted a comparison between our method and some other cryptanalysis methods that have been implemented with the PRESENT cipher, such as differential cryptanalysis, MITM and side-channel attack,… Additionally, we performed an analysis of memory, time implementation, as well as success possibility of our testing method and its modifications. In further research, we will explore a possible method that will speed up a real attack by utilizing distributed key lookup over the IoT network, where tables are looked up in parallel. We also plan to implement the testing method on more other lightweight cryptographic algorithms and analyze results. Moreover, we need modify the key size of PRESENT cipher and avoid using ECB mode in the implementation.

Acknowledgement. This research was supported by Energy Cloud R&D Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (2019M3F2A1073387), and this work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2021–0-00188, Open source development and standardization for AI enabled IoT platforms and interworking). Any correspondence related to this paper should be addressed to Dohyeun Kim.

References

1. Constantinos, K., Georgios, K., Angelos, S., Jeffrey, V.: DDoS in the IoT: mirai and other botnets. Computer **50**, 80–84 (2017)
2. Collins, S., McCombie, S.: Stuxnet: the emergence of a new cyber weapon and its implications. J. Policing, Intell. Counter Terror. **7**, 80–91 (2012)
3. Ray, P.: A survey on internet of things architectures. J. King Saud Univ. – Comput. Inform. Sci. **30**, 291–319 (2018)
4. Rana, M., Mamun, Q., Islam, R.: Lightweight cryptography in IoT networks: a survey. Futur. Gener. Comput. Syst. **129**, 77–89 (2022)
5. Bogdanov, A., et al.: L.N.IC. Science, Ed. Berlin, Heidelberg, Springer **2007**, 450–466 (2007)
6. Beaulieu, R., Treatman-Clark, S., Shors, D., Weeks, B., Smith, J., Wingers, L.: The SIMON and SPECK lightweight block ciphers. In: Proceedings of the 52nd Annual Design Automation Conference (2015)
7. Housley, R.: Using AES-CCM and AES-CGM authenticated encryption in the cryptographic message syntax(CMS). RFC Editor **5084**, 11 (2007)
8. Hellman, M.E.: A cryptanalytic time–memory trade-off. IEEE Trans. Inform. Theor. **26**(4), 401–406 (1980)
9. Tedjini, S., Andia-Vera, G., Zurita, M., Freire, R., Duroc, Y.: Augmented RFID Tags. In: 2016 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet) (2016)
10. Hahnel, D., Burgard, W., Fox, D., Fishkin, K., Philipose, M.: Mapping and localization with RFID technology. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ‘04. 2004 (2004)

11. Sim, B.-Y., Han, D.-G.: A study on the side-channel analysis trends for application to IoT devices (2020)
12. Li, P., Sun, L., Xiangyan, F., Lin, N.: Security in Wireless Sensor Networks," in Wireless Network Security: Theories and Applications, pp. 179–227. Heidelberg, Springer, Berlin Heidelberg, Berlin (2013)
13. Venugopalan, V., Patterson, C.D.: Surveying the hardware trojan threat landscape for the internet-of-things. J. Hardw. Syst. Secur. **2**(2), 131–141 (2018)
14. Jin, Y., Kupp, N., Makris, Y.: Experiences in hardware Trojan design and implementation. In: 2009 IEEE International Workshop on Hardware-Oriented Security and Trust (2009)
15. Danev, B., Luecken, H., Capkun, S., El Defrawy, K.: Attacks on physical-layer identification. In: WiSec '10: Proceedings of the Third ACM Conference on Wireless network security (2010)
16. Aras, E., Small, N., Ramachandran, G.S., St\'{e}phane, D., Joosen, W., Hughes, D.: Selective jamming of LoRaWAN using commodity hardware. In: MobiQuitous 2017: Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (2017)
17. Pelechrinis, K., Iliofotou, M., Krishnamurthy, S.V.: Denial of service attacks in wireless networks: the case of jammers. IEEE Commun. Surv. Tutor. **13**, 245–257 (2011)
18. Wallgren, L., Raza, S., Voigt, T.: Routing attacks and countermeasures in the RPL-based internet of things. Int. J. Distrib. Sensor Netw. **9**(8), 794326 (2013).
19. Crnogorac, J., Crnogorac, J., Vučinić, M., Kočan, E., Watteyne, T.: Dense multi-channel sniffing in large IoT networks. IEEE Access **10**, 105101–105110 (2022)
20. Andreica, G.R., Bozga, L., Zinca, D., Dobrota, V.: Denial of service and man-in-the-middle attacks against IoT devices in a GPS-based monitoring software for intelligent transportation systems. In: 2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet) (2020)
21. Sunardi, A.: Yudhana and Furizal, "Tsukamoto fuzzy inference system on internet of things-based for room temperature and humidity control,." IEEE Access **11**, 6209–6227 (2023)
22. Gomez, C., Chessa, S., Fleury, A., Roussos, G., Preuveneers, D.: Internet of things for enabling smart environments: a technology-centric perspective. J. Ambient Intell. Smart Environ. **11**(1), 23–43 (2019)
23. Tabaa, M., Monteiro, F., Bensag, H., Dandache, A.: Green industrial internet of things from a smart industry perspectives. Energy Rep. **6**, 430–446 (2020)
24. Park, M., Oh, H., Lee, K.: Security risk measurement for information leakage in IoT-based smart homes from a situational awareness perspective. Sensors **19**(9), 2148 (2019)
25. Altayaran, S., Elmedany, W.: Security threats of application programming interface (API's) in internet of things (IoT) communications. In: 4th Smart Cities Symposium (SCS 2021) (2021)
26. Balliu, M., Bastys, I., Sabelfeld, A.: Securing IoT Apps. IEEE Secur. Privacy **17**(5), 22–29 (2019)
27. Zikria, Y.B., Ali, R., Afzal, M.K., Kim, S.W.: Next-generation internet of things (IoT): opportunities, challenges, and solutions. Sensors, vol. **21**(4), 1174 (2021)
28. El-Hajj, M., Fadlallah, A., Chamoun, M., Serhrouchni, A.: A Survey of Internet of Things (IoT) Authentication Schemes. Sensors **19**(5), 1141 (2019)
29. Ding, S., Cao, J., Li, C., Fan, K., Li, H.: A Novel attribute-based access control scheme using blockchain for IoT. IEEE Access **7**, 38431–38441 (2019)
30. Mandal, S., Bera, B., Sutrala, A.K., Das, A.K., Choo, K.K.R., Park, Y.: Certificateless-Signcryption-Based Three-Factor User Access Control Scheme for IoT Environment. IEEE Int. Things J. **7**(4), 3184–3197 (2020)
31. Li, D., Han, D., Crespi, N., Minerva, R., Li, K.-C.: A blockchain-based secure storage and access control scheme for supply chain finance. J. Supercomput. **79**(1), 109–138 (2023)

32. Qaddoura, R., M. Al-Zoubi, A., Faris, H., Almomani, I.: A multi-layer classification approach for intrusion detection in IoT networks based on deep learning. Sensors **21**(9), 2987 (2021)
33. Hamza, R., Yan, Z., Muhammad, K., Bellavista, P., Titouna, F.: A privacy-preserving cryptosystem for IoT E-healthcare. Inf. Sci. **527**, 493–510 (2020)
34. Perazzo, P., Righetti, F., La Manna, M., Vallati, C.: Performance evaluation of attribute-based encryption on constrained IoT devices. Comput. Commun. **170**, 151–163 (2021)
35. McKay, K., Bassham, L., Sönmez Turan, M., Mouha, N.: Report on Lightweight Cryptography. In: NIST (2017)
36. Tropea, M., Spina, M.G., De Rango, F., Gentile, A.F.: Gentile, "security in wireless sensor networks: a cryptography performance analysis at MAC Layer. Future Internet, **14**(5), 145 (2022)
37. Thakor, V.A., Razzaque, M.A., Khandaker, M.R.A.: Lightweight cryptography algorithms for resource-constrained IoT devices: a review, comparison and research opportunities. IEEE Access **9**, 28177–28193 (2021)
38. Pareek, M., Mishra, G., Kohli, V.: Deep learning based analysis of key scheduling algorithm of PRESENT cipher. IACR Cryptol. ePrint Arch. **2020**, 981 (2020)
39. Jithendra, K.B., Shahana, T.K.: New Biclique cryptanalysis on full-round PRESENT-80 block cipher. SN Comput. Sci. **1**(2),(2020). https://doi.org/10.1007/s42979-020-0103-z
40. Lo, O., Buchanan, W.J., Carson, D.: Correlation power analysis on the PRESENT block cipher on an embedded device. In: Proceedings of the 13th International Conference on Availability, Reliability and Security (2018)
41. Vaudenay, S. (ed.): Progress in Cryptology – AFRICACRYPT 2008: First International Conference on Cryptology in Africa, Casablanca, Morocco, June 11-14, 2008. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
42. Pieprzyk, J.: The 10th Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1–5, 2010. Proceedings, Springer **2010**, 302–307 (2010)
43. Oechslin, P.: Making a faster cryptanalytic time-memory trade-off. In: Advances in Cryptology – CRYPTO 2003. Springer-Verlag, Boston (2003)
44. Perry, L.: IoT Security. In: Internet of Things for Architects, p. 515. Packt Publishing Lt, Mumbai (2018)
45. Wheeler, D.M., Fagbemi, D.D.: Security architecture for real IoT systems. In: The IoT architect's guide to attainable security and privacy, Boca Raton, CRC Press, p. 497 (2020)

The Prediction Model of Water Level in Front of the Check Gate of the LSTM Neural Network Based on AIW-CLPSO

Linqing Gao[1,2], Dengzhe Ha[3](✉), Litao Ma[4], and Jiqiang Chen[4]

[1] School of Water Conservancy and Hydroelectric Power, Hebei University of Engineering, Handan 056038, China
[2] Hebei Key Laboratory of Intelligent Water Conservancy, Handan 056038, China
[3] School of Civil Engineering, Tianjin University, Tianjin 300072, China
hadengzhe001129@163.com
[4] School of Mathematics and Physics Science and Engineering, Hebei University of Engineering, Handan 056038, China

Abstract. The water level in front of the check gate of water transfer projects is affected by physical factors such as rainfall, terrain and hydraulic structures. Its fluctuation trend has strong non-linear and stochastic characteristics, and it is difficult to predict accurately and efficiently by hydrodynamic model. To solve the problem of predicting water level in front of check gate, a long short term memory (LSTM) neural network based on adaptive inertia weight comprehensive learning particle swarm optimization algorithm (AIW-CLPSO) is proposed. The AIW and CLPSO are adopted to improve the global optimization ability and convergence velocity of PSO in the proposed model. The model was applied to the water level prediction in front of the Chaohu Lake check gate. The example of the water level prediction in front of the Chaohu Lake check gate shows that the proposed model can obtain the optimal parameters of LSTM neural network, which overcomes the limitations of difficult parameter selection and inaccurate prediction.

Keywords: Particle swarm optimization · long short term memory neural network · adaptive inertia weight · comprehensive learning particle swarm optimization · water level prediction

1 Introduction

Water resources scheduling is an effective means of alleviating the uneven distribution and shortage of water resources. The main forms include water resources scheduling of river basins and water resources scheduling of water transfer projects. The large water transfer projects mainly achieve water resources scheduling by setting hydraulic structures such as pumping station, check gate, and inverted siphon. The main functions of check gate include water conveyance, flood control, and ecosystem protection. The water level in front of check gate is an important control index for operating water transfer projects. Predicting the water level in front of the check gate is of great significance for water regulation, hydraulic engineering safety and ecosystem protection.

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 306–311, 2024.
https://doi.org/10.1007/978-981-99-8211-0_28

Deep learning is a research field that has developed rapidly in recent years. Originally, it is a branch of machine learning, which summarizes general laws from limited samples through algorithms, and the laws can be applied to data analysis [1]. The LSTM neural network introduces a gating mechanism to control the velocity of information accumulation, which includes selectively adding new information and forgetting the previously accumulated information. It is outstanding in the field of time series data prediction by virtue of its advantages [2].

In recent years, scholars at home and abroad have applied LSTM neural networks to predict hydrological time series. Hu et al. verified that the LSTM network has high prediction accuracy in terms of rainfall and runoff [3]. Zhang et al. studied the applicability of the LSTM network model in water level prediction and verified its accuracy [4]. To improve the efficiency and accuracy of LSTM prediction, some scholars have applied the combination of swarm optimization intelligent algorithm and the LSTM to study the prediction of hydrological time series. Xu et al. used the particle swarm optimization (PSO) algorithm to optimize the LSTM hyper-parameters, which improved the learning ability of the hydrological time series characteristics of the model and flood forecasting accuracy in different regions [5]. Du et al. proposed a kernel density estimation method based on the PSO algorithm, and combined it with the LSTM prediction model, to obtain the prediction interval of urban water storage [6]. Therefore, it is fair to say that the combination of the PSO and LSTM effectively improves the prediction ability of the LSTM.

The PSO can easily fall into locally optimal solution prematurely when dealing with multimodal problems, and its convergence velocity is slow. To improve the global optimization ability and the convergence efficiency of the optimal solution, this study proposes an adaptive inertia weight comprehensive learning particle swarm optimization (AIW-CLPSO) LSTM hyper-parameters optimization algorithm. Combined with the water level data in front of the Chaohu Lake check gate, the validity and stability of the LSTM hyper-parameters optimization algorithm based on AIW-CLPSO are verified, through the comparative analysis with the prediction results of LSTM and PSO-LSTM, and the application of the LSTM hyper-parameters optimization algorithm based on the AIW-CLPSO in predicting water level in front of the check gate at different time scales is explored.

2 LSTM Neural Network Based on Adaptive Inertia Weight Comprehensive Learning Particle Swarm Optimization

2.1 Comprehensive Learning Particle Swarm Optimization

The PSO algorithm uses the individual best position and global best position of particles to update the velocity and position of the particles. This method has a high convergence velocity, but it will reduce the population diversity and easily fall into the local optimal value in the multimodal state. To increase population diversity, Liang et al. proposed a comprehensive learning particle swarm optimization (CLPSO) algorithm [7]. This algorithm uses the individual best position of all particles to update the velocity and position of the particles.

The CLPSO algorithm has a strong exploration ability and performs prominently in dealing with multimodal problems [7]. It has a low convergence velocity because it cancels the learning link to the global best position.

2.2 Adaptive Updating Inertia Weight Strategy

The choice of inertia weight in the PSO algorithm, which is related to the convergence performance of the entire algorithm, is extremely important. The large value is conducive to global search, and the convergence velocity is fast, but it is difficult to achieve an accurate solution; The small value is good for local search and can achieve a more accurate solution, but it has a slow the convergence velocity.

In this study, the nonlinear adaptive inertia weight (AIW) coefficient formula is adopted to improve the convergence velocity of the CLPSO. The specific expression formula is as follows [8]:

$$\omega_i = \begin{cases} \omega_{\min} + \frac{(\omega_{\max} - \omega_{\min})(f_i - f_{\min})}{(f_{avg} - f_{\min})}, & f_i \le f_{avg} \\ \omega_{\max}, & f_i > f_{avg} \end{cases} \tag{1}$$

where $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum values of the inertia weight of the particle respectively. f_i indicates the fitness value of the ith particle. f_{avg} and $f_{\min}$ are the average and minimum fitness value, respectively, of all current particles. The inertia weight increases when the fitness value of each particle is consistent or locally optimal, and it decreases discretely.

2.3 Adaptive Inertia Weight Comprehensive Learning Particle Swarm Optimization

Combining the AIW and CLPSO, the obtained Adaptive inertia weight comprehensive learning particle swarm optimization (AIW-CLPSO) not only has fast convergence velocity but also has strong exploration ability and adaptability. Therefore, its optimization effect is ideal.

2.4 LSTM Neural Network Model Based on AIW-CLPSO

In this study, the three layer hidden layer of the LSTM is selected. The LSTM has four hyper-parameters that have an important impact on the prediction performance of the model, such as the number of neurons in the LSTM hidden layer (L_1, L_2, L_3) and learning batch size (batch_size). The four key hyper-parameters are selected as the characteristics of particle optimization. The fitness function (F) of an individual population with LSTM hyper-parameters is defined as follows [9]:

$$F = \frac{1}{2}\left[\frac{1}{P}\sum_{p=1}^{P}\left|\frac{y_p - \hat{y}_p}{y_p}\right| + \frac{1}{Q}\sum_{q=1}^{Q}\left|\frac{y_q - \hat{y}_q}{y_q}\right|\right] \tag{2}$$

where P and Q are the numbers of the training set and verification set data, respectively. y_p and $\hat{y}_p$ are the real and predicted values of the training set, respectively. y_q and $\hat{y}_q$

are the real and predicted values of the validation set, respectively. In PSO, the error function between the real and predicted values of the training set is considered the fitness function, but the verification set can be used to verify whether the model is over-fitted. The verification set error also has an important impact on the selection of model parameters. The average value of the two errors is selected in this study.

The LSTM model is adjusted and optimized using the AIW-CLPSO. We propose a LSTM neural network model based on the AIW-CLPSO. Figure 1 depicts a specific model.

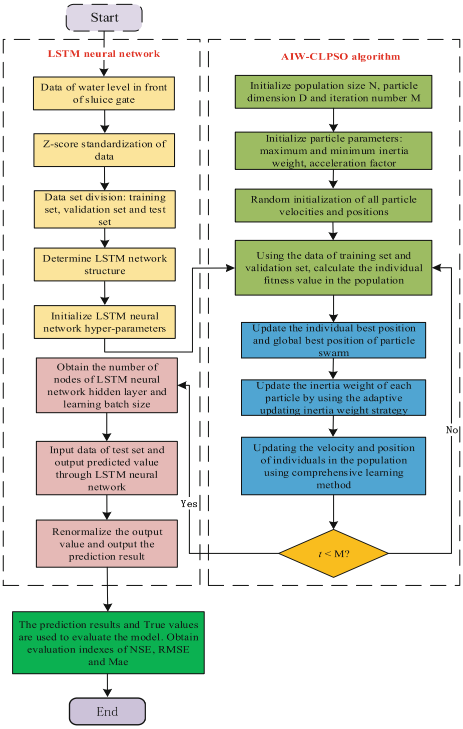

Start
LSTM neural network
Data of water level in front of sluice gate
Z-score standardization of data
Data set division: training set, validation set and test set
Determine LSTM network structure
Initialize LSTM neural network hyper-parameters
Obtain the number of nodes of LSTM neural network hidden layer and learning batch size
Input data of test set and output predicted value through LSTM neural network
Renormalize the output value and output the prediction result
AIW-CLPSO algorithm
Initialize population size N, particle dimension D and iteration number M
Initialize particle parameters: maximum and minimum inertia weight, acceleration factor
Random initialization of all particle velocities and positions
Using the data of training set and validation set, calculate the individual fitness value in the population
Update the individual best position and global best position of particle swarm
Update the inertia weight of each particle by using the adaptive updating inertia weight strategy
Updating the velocity and position of individuals in the population using comprehensive learning method
t < M?
Yes
No
The prediction results and True values are used to evaluate the model. Obtain evaluation indexes of NSE, RMSE and Mae
End

Fig. 1. A flowchart of the LSTM neural network model based on the AIW-CLPSO.

3 An Example of Water Level Prediction in front of Chaohu Lake Check Gate

This section analyzes the effectiveness of the proposed LSTM neural network based on the AIW-CLPSO by predicting the water level in front of the Chaohu Lake check gate and compares it with the prediction results of the LSTM and PSO-LSTM.

The water level in front of the Chaohu Lake check gate from January 1, 2015, to December 31, 2017, was selected as the training and validation sets, and the water level from January 1, 2018, to June 24, 2018, was selected as the test set, in which the water level monitoring frequency is 1h. The water level in front of the Chaohu Lake check gate 24h before the water level to be predicted was selected as the input data, and the predicted water level data in steps of 1h is selected as the output data.

To solve the problem of insufficient prediction accuracy of a single model, the PSO and AIW-CLPSO are used to optimize the LSTM hyper-parameters.

Related parameter settings: inertia weight is 0.8 and learning factor is 2 in the basic PSO. The LSTM neural network is composed of an input layer, three LSTM hidden layers, and an output layer. The maximum number of evolutionary iterations in the PSO and AIW-CLPSO is 30, and the population size is 20. The minimum inertia weight $\omega_{\min} = 0.4$ and maximum inertia weight $\omega_{\max} = 0.9$ in the AIW-CLPSO. The value range of the number of the three hidden layer cells in the LSTM is $L_1, L_2, L_3 \in [1, 256]$ and the learning batch size is batch_ size $\in$ [5, 128]. The output data is the predicted water level with a step length of 1 h.

Table 1 shows that the NSE of the three models is greater than 0.97, RMSE is less than 0.04, and MAE is less than 0.03. Among them, the AIW-CLPSO-LSTM model has excellent performance and each evaluation index is also the best, followed by the PSO-LSTM. Therefore, the AIW-CLPSO-LSTM has the highest accuracy in the prediction experiment.

Table 1. Evaluation Index of each model on the test set

Prediction Model	Evaluation Index		
	NSE	RMSE/m	MAE/m
LSTM	0.9773	0.0337	0.0224
PSO-LSTM	0.9840	0.0283	0.0182
AIW-CLPSO-LSTM	**0.9851**	**0.0273**	**0.0174**

According to the AIW-CLPSO-LSTM model and experimental process, particles comprehensively learn the optimal dimensions of each individual to obtain a large amount of disturbance in the early stage of the AIW-CLPSO iteration. This enlarges the optimization space of the problem, improves the diversity of the particle population, and obtains a larger solution space.

4 Conclusions

In this study, an LSTM neural network based on the AIW-CLPSO algorithm is proposed, and its application in predicting water level in front of a check gate at different time scales is explored. From the construction process of the optimization algorithm, and experimental results of the model, the following can be observed:

1. The LSTM neural network based on the AIW-CLPSO algorithm proposed in this study has high prediction accuracy and stability in predicting the water level in front of the check gate under different time scales. Simultaneously, it is also used to predict

and analyze of other hydrological time series, which can be further explored and improved.

2. Although the LSTM neural network based on AIW-CLPSO has high accuracy and stability in the prediction of water level in front of the check gate, there is still a big gap in the prediction of extremely high or extremely low water level. Controlling extremely high or extremely low water level in front of the check gate is of great significance for water scheduling, hydraulic engineering safety and ecosystem protection, which will be the focus of our next research.

Acknowledgements. The paper is supported by the Young Program of the National Natural Science Foundation of China (62006068). The authors also acknowledge Professor Minghu Ha for his guidance.

References

1. Qiu, X.P.: Neural Networks and Deep Learning. China Machine Press, Beijing (2020)
2. Yin, Z.K., Liao, W.H., Wang, R.J., Lei, X.H.: Rainfall-runoff modelling and forecasting based on long short-term memory (LSTM). South-to-North Water Transfers Water Sci. Technol. **17**(6), 1–9 (2019)
3. Hu, C.H., Wu, Q., Li, H., Jian, S.Q., Li, N., Lou, Z.Z.: Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. Water **10**(11), 1–16 (2018)
4. Zhang, D., Lindholm, G., Ratnaweera, H.: Use long short-term memory to enhance internet of things for combined sewer overflow monitoring. J. Hydrol. **556**, 409–418 (2018)
5. Xu, Y.H., et al.: Research on particle swarm optimization in LSTM neural networks for rainfall-runoff simulation. J. Hydrol. **608**, 127553 (2022)
6. Du, B.G., Huang, S., Guo, J., Tang, H.T., Wang, L., Zhou, S.W.: Interval forecasting for urban water demand using PSO optimized KDE distribution and LSTM neural networks. Appl. Soft Comput. **122**, 108875 (2022)
7. Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans. Evol. Comput. **10**(3), 281–295 (2006)
8. Kang, L.L., Dong, W.Y., Tian, J.S.: Opposition-based particle swarm optimization with adaptive Cauchy mutation. Comput. Sci. **42**(10), 226–231 (2015)
9. Ren, X.Q., Liu, S.L., Yu, X.D., Dong, X.: A method for state-of-charge estimation of lithium-ion batteries based on PSO-LSTM. Energy **234**, 121236 (2021)

Author Index

J. S. Park et al. (Eds.): PDCAT 2023, LNEE 1112, pp. 313–314, 2024.
https://doi.org/10.1007/978-981-99-8211-0