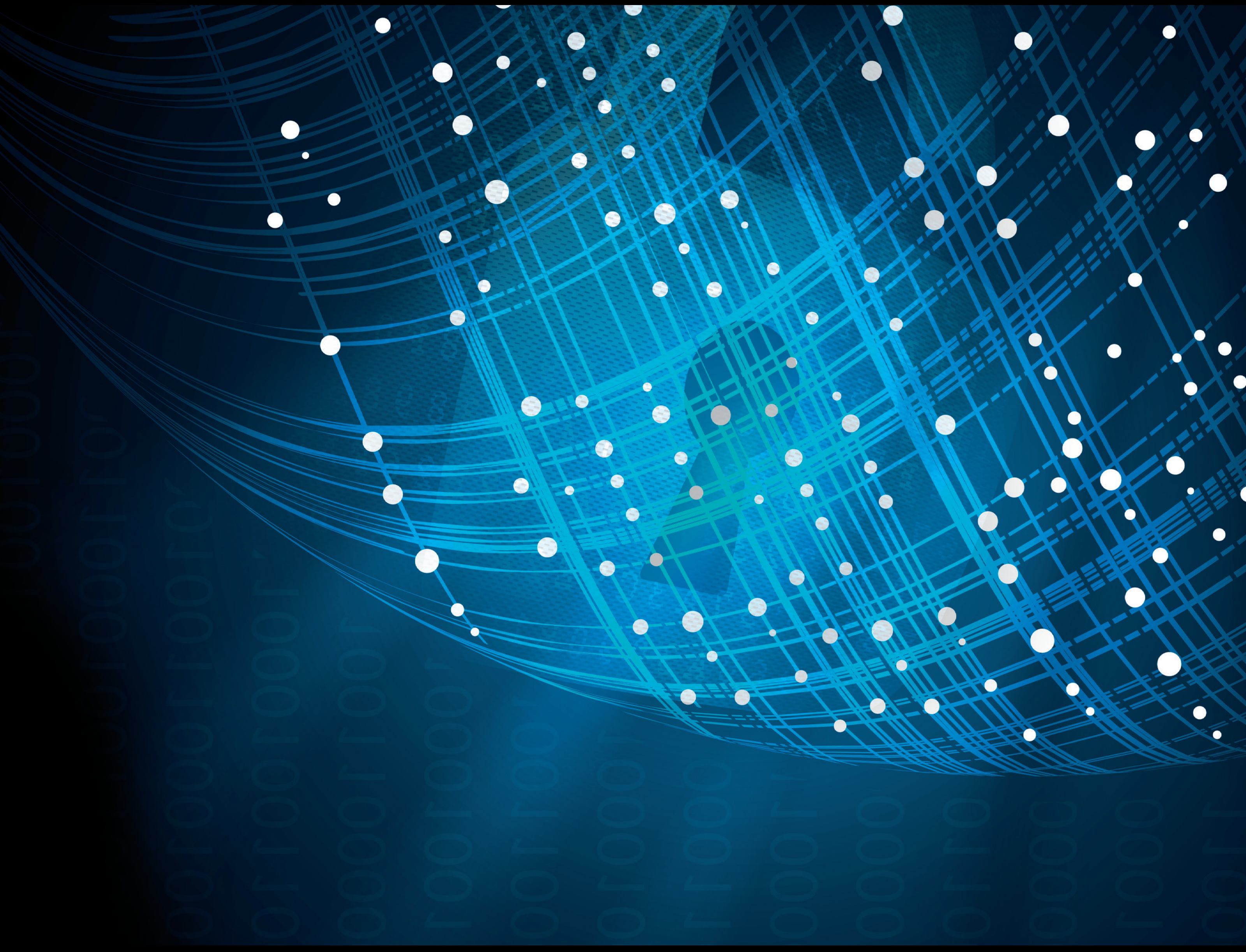
Security and Communication Networks

Secure Deployment of Commercial Services in Mobile Edge Computing

Lead Guest Editor: Xiaolong Xu
Guest Editors: Xuyun Zhang, Gautam Srivastava, Hao Wang, and Wanchun Dou



Secure Deployment of Commercial Services in Mobile Edge Computing

Security and Communication Networks

Secure Deployment of Commercial Services in Mobile Edge Computing

Lead Guest Editor: Xiaolong Xu
Guest Editors: Xuyun Zhang, Gautam Srivastava,
Hao Wang, and Wanchun Dou

Chief Editor

Associate Editors

Academic Editors

Contents

Corrigendum to “Rational Protocols and Attacks in Blockchain System”
Tao Li, Yuling Chen, Yanli Wang, Yilei Wang, Minghao Zhao, Haojia Zhu, Youliang Tian, Xiaomei Yu, and Yixian Yang
Corrigendum (2 pages), Article ID 9856969, Volume 2021 (2021)

A Survey on Secure Deployment of Mobile Services in Edge Computing
Mengmeng Cui, Yiming Fei, and Yin Liu
Review Article (8 pages), Article ID 8846239, Volume 2021 (2021)

A Group Recommendation System of Network Document Resource Based on Knowledge Graph and LSTM in Edge Computing
Yuezhong Wu, Qiang Liu, Rongrong Chen, Changyun Li, and Ziran Peng
Research Article (11 pages), Article ID 8843803, Volume 2020 (2020)

cHybriDroid: A Machine Learning-Based Hybrid Technique for Securing the Edge Computing
Afifa Maryam, Usman Ahmed, Muhammad Aleem, Jerry Chun-Wei Lin, Muhammad Arshad Islam, and Muhammad Azhar Iqbal
Research Article (14 pages), Article ID 8861639, Volume 2020 (2020)

A Prediction Approach for Video Hits in Mobile Edge Computing Environment
Xiulei Liu, Shoulu Hou, Qiang Tong, Xuhong Liu, Zhihui Qin, and Junyang Yu
Research Article (6 pages), Article ID 8857564, Volume 2020 (2020)

Deploying GIS Services into the Edge: A Study from Performance Evaluation and Optimization Viewpoint
Jianbing Zhang, Bowen Ma, and Jiwei Huang
Research Article (13 pages), Article ID 8822990, Volume 2020 (2020)

High-Performance Routing Emulation Technologies Based on a Cloud Platform
Jianyu Chen, Xiaofeng Wang, Leiting Tao, and Yuan Liu
Research Article (17 pages), Article ID 8825749, Volume 2020 (2020)

An Intelligent Real-Time Traffic Control Based on Mobile Edge Computing for Individual Private Environment
Sa Math, Lejun Zhang, Seokhoon Kim, and Intae Ryoo
Research Article (11 pages), Article ID 8881640, Volume 2020 (2020)

Routing Strategy for LEO Satellite Networks Based on Membership Degree Functions
Jian Zhou, Qian Bo, Lijuan Sun, Juan Wang, and Xiaoyong Yan
Research Article (9 pages), Article ID 8822985, Volume 2020 (2020)

A Novel Machine Learning-Based Approach for Security Analysis of Authentication and Key Agreement Protocols
Behnam Zahednejad, Lishan Ke, and Jing Li
Research Article (15 pages), Article ID 8848389, Volume 2020 (2020)

Qualitative Analysis of Commercial Services in MEC as Phased-Mission Systems
Yuhuan Gong and Yuchang Mo
Research Article (11 pages), Article ID 8823952, Volume 2020 (2020)

Rational Protocols and Attacks in Blockchain System
Tao Li, Yuling Chen, Yanli Wang, Yilei Wang, Minghao Zhao, Haojia Zhu, Youliang Tian, Xiaomei Yu, and Yixian Yang
Research Article (11 pages), Article ID 8839047, Volume 2020 (2020)

A Novel IM Sync Message-Based Cross-Device Tracking
Naixuan Guo, Junzhou Luo, Zhen Ling, Ming Yang, Wenjia Wu, and Xiaodan Gu
Research Article (15 pages), Article ID 8891664, Volume 2020 (2020)

A Robust Image Watermarking Approach Using Cycle Variational Autoencoder
Qiang Wei, Hu Wang, and Gongxuan Zhang
Research Article (9 pages), Article ID 8869096, Volume 2020 (2020)

Artificial Intelligence for Securing IoT Services in Edge Computing: A Survey
Zhanyang Xu, Wentao Liu, Jingwang Huang, Chenyi Yang, Jiawei Lu, and Haozhe Tan
Review Article (13 pages), Article ID 8872586, Volume 2020 (2020)

Collaborative Intelligence: Accelerating Deep Neural Network Inference via Device-Edge Synergy
Nanliang Shan, Zecong Ye, and Xiaolong Cui
Research Article (10 pages), Article ID 8831341, Volume 2020 (2020)

A Review of Techniques and Methods for IoT Applications in Collaborative Cloud-Fog Environment
Jielin Jiang, Zheng Li, Yuan Tian, and Najla Al-Nabhan
Review Article (15 pages), Article ID 8849181, Volume 2020 (2020)

An Improved Elman Network for Stock Price Prediction Service
Bo Liu, Qilin Wu, and Qian Cao
Research Article (9 pages), Article ID 8824430, Volume 2020 (2020)

A Dynamic Privacy Protection Mechanism for Spatiotemporal Crowdsourcing
Tianen Liu, Yingjie Wang, Zhipeng Cai, Xiangrong Tong, Qingxian Pan, and Jindong Zhao
Research Article (13 pages), Article ID 8892954, Volume 2020 (2020)

Packet-Based Intrusion Detection Using Bayesian Topic Models in Mobile Edge Computing
Xuefei Cao, Yulong Fu, and Bo Chen
Research Article (12 pages), Article ID 8860418, Volume 2020 (2020)

A Task Offloading Method with Edge for 5G-Envisioned Cyber-Physical-Social Systems
Jielin Jiang, Xing Zhang, and Shengjun Li
Research Article (9 pages), Article ID 8867094, Volume 2020 (2020)

Contents

A Multifault Diagnosis Method of Gear Box Running on Edge Equipment
Xiao-ping Zhao, Yong-hong Zhang, and Fan Shao
Research Article (13 pages), Article ID 8854236, Volume 2020 (2020)

A Survey on Services Provision and Distribution of Official and Commercial Intellectual Property Platforms
Yang Wang, Haijin Gui, and Lei Ma
Review Article (14 pages), Article ID 8833416, Volume 2020 (2020)

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9856969, 2 pages
https://doi.org/10.1155/2021/9856969

Corrigendum

Corrigendum to "Rational Protocols and Attacks in Blockchain System"

Tao Li,[1,2,3] Yuling Chen,[1,2] Yanli Wang,[3] Yilei Wang,[3] Minghao Zhao,[4] Haojia Zhu,[3] Youliang Tian,[1,2] Xiaomei Yu,[5] and Yixian Yang[1,2]

[1]*State Key Laboratory of Public Big Data, Guizhou University, Guiyang, Guizhou 550025, China*
[2]*College of Computer Science and Technology, Guizhou University, Guiyang, Guizhou 550025, China*
[3]*School of Information Science and Engineering, Qufu Normal University, Rizhao 276825, China*
[4]*School of Software, Tsinghua University, Beijing 100084, China*
[5]*School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China*

Correspondence should be addressed to Yuling Chen; ylchen3@gzu.edu.cn

Received 29 April 2021; Accepted 29 April 2021; Published 29 May 2021

In the article titled "Rational Protocols and Attacks in Blockchain System" [1], a number of references in Figures 2–4 are incorrect due to an error made during the preparation of the manuscript. The corrected figures with the updated numbering are shown as follows.

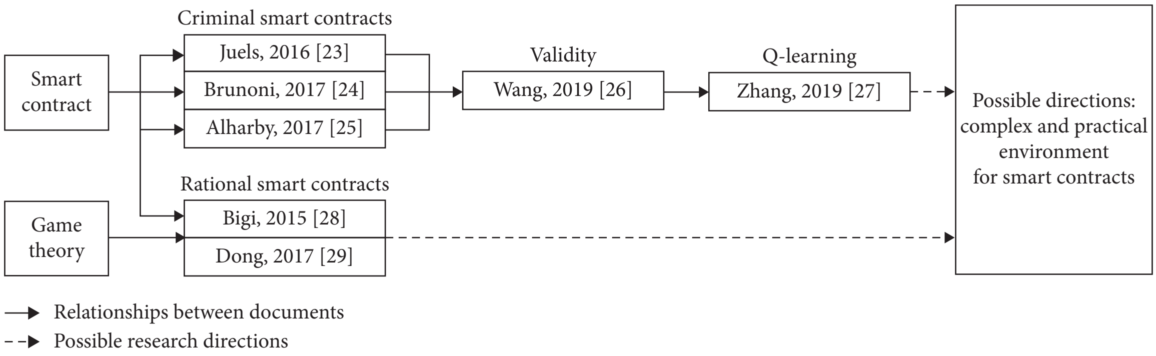


Figure 2: The possible directions for smart contracts.

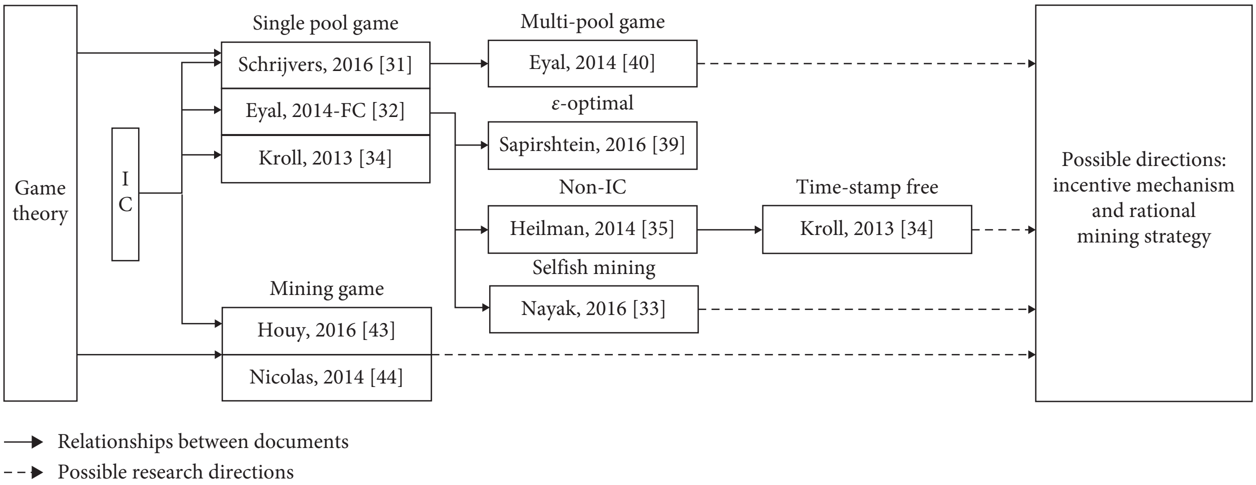


Figure 3: The possible directions for rational mining attacks.

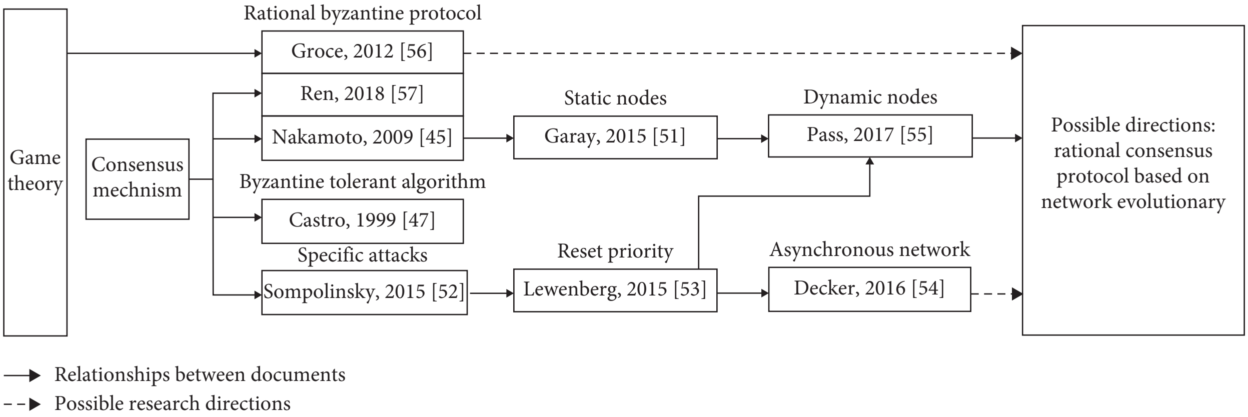


Figure 4: The possible directions for consensus mechanism.

References

[1] T. Li, Y. Chen, Y. Wang et al., "Rational protocols and attacks in blockchain system," *Security and Communication Networks*, vol. 2020, Article ID 8839047, 11 pages, 2020.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 8846239, 8 pages
https://doi.org/10.1155/2021/8846239

Review Article

A Survey on Secure Deployment of Mobile Services in Edge Computing

Mengmeng Cui ,[1] Yiming Fei ,[2] and Yin Liu [3]

[1]*School of Applied Technology, Nanjing University of Information Science & Technology, Nanjing 21000, China*
[2]*School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 21000, China*
[3]*Yunnan Air Traffic Mannagment Sub-Bureau, CAAC, Changshui International Airport, Kunming 650000, China*

Correspondence should be addressed to Yiming Fei; feiyiming@nuist.edu.cn

Received 10 August 2020; Revised 19 October 2020; Accepted 17 December 2020; Published 4 January 2021

Academic Editor: David Megías Jiménez

Mobile edge computing (MEC) is an emerging technology that is recognized as a key to 5G networks. Because MEC provides an IT service environment and cloud-computing services at the edge of the mobile network, researchers hope to use MEC for secure service deployment, such as Internet of vehicles, Internet of Things (IoT), and autonomous vehicles. Because of the characteristics of MEC which do not have terminal servers, it tends to be deployed on the edge of networks. However, there are few related works that systematically introduce the deployment of MEC. Also, secure service deployment frameworks with MEC are even rare. For this reason, we have conducted a comprehensive and concrete survey of recent research studies on secure deployment. Although numerous research studies and experiments about MEC service deployment have been conducted, there are few systematic summaries that conclude basic concepts and development strategies about secure service deployment of commercial MEC. To make up for the gap, a detailed and complete survey about relative achievements is presented.

1. Introduction

In 2013, MEC was first introduced when Nokia Siemens and IBM developed an MEC platform where applications can run directly. Later, MEC was standardized by the European Telecommunications Standards Institute (ETSI) and Industry Specification Group (ISG). Also, European 5G Infrastructure Public Private Partnership regards MEC as a prime emerging technology for 5G networks [1].

In recent years, wearable devices, sensors, and a lot of devices of internet of things (IoT) such as wearable devices become more universal [2]. According to the research of Ericsson, it is estimated that 32 billion terminal devices will be connected to the mobile network by 2030 [3]. Due to the explosive growth of the amount of terminal equipment and data, it is not hard to see that online service providers will face significant challenges in securing reliable and low-latency connections for terminal users [4]. To solve the problems, researchers decide to deploy computation resources, network control functions, and cached data near microbasic stations and macrobasic stations. This model is called mobile edge computing [4]. Usually, edge servers cover specific geographic areas so that users can connect to them easily. A large number of edge servers will be deployed in a distributed manner so that they can cover different geographic areas. Their coverage often overlaps, which may lead to wasting resources.

Because the coverage of MEC is not large enough, operators have to cost more to serve the users. Furthermore, sometimes user requests cannot be processed by the closest edge servers, and how to transfer them to another server is also a problem [5]. On the other hand, problems such as the risk of user data leakage and safety of terminal devices are urgent to be solved.

Because the above problems are caused by service deployment, this article refers to the problems as secure deployment of mobile services in edge computing. Secure deployment of mobile services has been taken into consideration in three aspects as follows.

1.1. MEC Service Deployment. Because edge computing is considered as one of the key technologies to meet low latency, mission-critical, and IoT services requirements in the future, MEC service deployment is required to be flexible and efficient. Also, it is required to be secure and easy to maintain. Services which are provided to users need deploying in the MEC servers themselves. That is why MEC servers can handle users' requests correctly when the MEC server is deployed correctly. If requests are sent to any MEC server, the receiving MEC server will not deploy the service themselves and send it to the neighboring one. Moreover, if there are a lot of neighboring MEC nodes which are running one requested service, the one of them that is chosen must ensure the QoS of the service. We make a model which is named round trip time (RTT) between mobile devices and MEC servers so that we can choose the best destination node. On the other hand, there may be a strain on processing time if a few mobile devices access the same MEC for services. Therefore, the service discovery protocol needs to consider the processing burden on MEC [6].

MEC is an approach complementary to network functions virtualization (NFV) based on a virtualized platform. In fact, while NVF is focused on network functions, the MEC framework allows applications to run at the edge of network. The infrastructure from MEC to NFV or functions on networks is quite similar; thus, in order to enable operators to benefit from their investment as much as possible, it will be helpful to reuse infrastructure management of NFV which hosts VNFs (virtual network functions) and MEC applications on the same platform [7].

MEC service deployment allows operators to run core services near the end-devices. And it enables users and content providers to serve and adjust context-aware services. To meet the requirements of 5G and fill the huge requirements of users and operators in the future, MEC needs to be deployed. Also, the correct deployment of MEC can solve the problem of lack of flexibility [8].

1.2. Computation Offloading. Generally speaking, there are three scenarios for offloading: local execution means the whole computation is done locally and does not transfer to MEC [9]. Full offloading means contrary to local execution, the whole computation is offloaded completely, so partial offloading means a part of computation is offloaded while the rest is done locally [10]. Correct and appropriate service deployment can reduce the pressure of computation offloading. Nowadays, billions of mobile devices are connected to the Internet. Because researchers usually assume that mobile computation offloading relies on a central cloud, it is a huge challenge for limited computation on the central cloud [11]. So computation offloading is a very pivotal technology for MEC [12]. Computation offloading and resource allocation are both parts of the universal system, and they both contribute to user experience, which cannot be guaranteed by the optimization of one single segment [13].

1.3. Data Placement. Because of the rapid growth of MEC services, major service providers now use a lot of geographically dispersed data centers so that the users can get better service experiences. In this way, users can avoid waiting a long time for data transmission [14]. Mobile devices produce a lot of data, which is stored for analysis. However, due to the limited storage of mobile devices, data need to be placed on remote data centers to process further [15]. Generally, data placement is divided into two parts which are random placement and planned placement. In random placement, the sensors are randomly distributed, and in planned placement, the sensors are deployed selectively [16]. The specific requirements of a good strategy data placement are as follows: (1) the scientific workflow structure is complex and datasets are large. Therefore, the data placement strategy should ensure high cohesion within the data center and low coupling among different data centers, thus reducing data transfer time between data centers that combine edge computing and cloud computing. (2) For security reasons, private datasets should be stored in the edge data center. Due to the limited storage capacity of edge data centers, some datasets must be transferred across different data centers. Placing low latency data sets with limited bandwidth and fixed private data sets is a challenge [17].

The organizational structure of the article is as follows: first of all, we introduce the basic concepts and definitions of mobile devices and MEC. Next, we generally overview secure deployment frameworks with commercial MEC and propose a new framework. Meanwhile, we introduce some methods and technologies of deployment. Then, we point out some challenges of secure deployment. Moreover, we provide some solutions to fill these gaps. And then, we discuss some open issues and problems. At last, we make a rough summary of the secure deployment of mobile service.

2. Basic Concepts and Definitions

In this section, we review the basic concepts and definitions of MEC.

2.1. Service Deployment. Usually, service deployment is based on virtualization technology. In other words, a deployed service is a VM or a collection of VMs. The service is composed of functional and nonfunctional requirements for one deployment target [18]. The deployment core of MEC is NFV, software-defined network (SDN), and cloud computing technology. NFV is a way to design, deploy, and manage network services. The main idea of NFV is to decouple the physical network devices from the functionality running on it [19]. Software-defined networks (SDNs) are controlled programmatically. Network state is managed by logically centralized control programs with a global network view and written directly to the switch for using standard API [20].

2.2. Mobile Edge Computing. As shown in Figure 1, data of terminal devices, including but not limited to mobile devices and vehicles, are transferred to MEC originally, then MEC will complete most of the computation tasks, and the remaining unsolvable are transferred to the cloud. The basic idea of MEC is to "sink" the functions of the central cloud

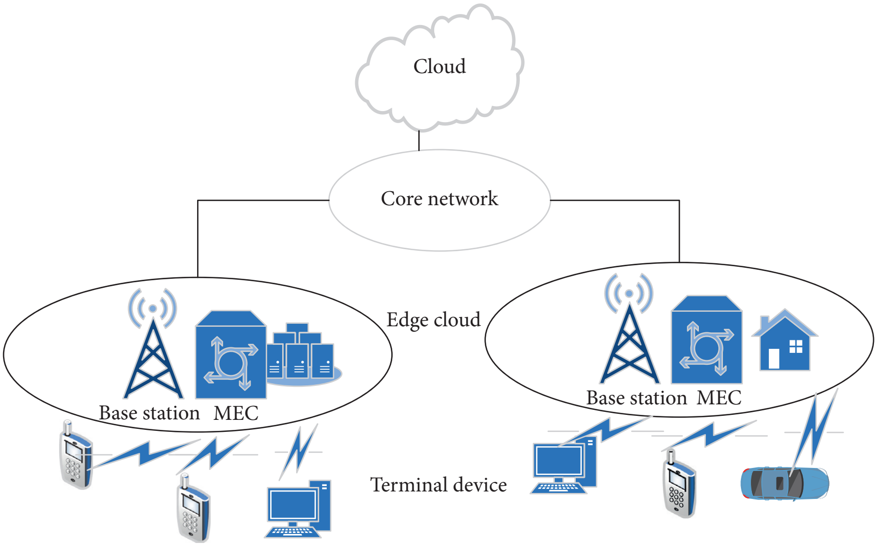


Figure 1: The architecture of MEC.

data center to the network edge which is closer to the mobile end users, and operators deploy MEC near the users to provide necessary computing, storage, and other services for the mobile end users. Though the research in MEC is not detailed and rigorous, some researchers have proposed MEC definitions that an open cloud platform which uses some end-user clients is located on the mobile edge to perform a massive amount of real-time storage (rather than stored primarily in cloud data centers) [21]. MEC can offer a service environment that has the advantages of ultralow latency, high-bandwidth, and direct access to real-time network information. And it is closed to subscribers [22]. Unlike the centralized cloud servers or peer-to-peer mobile devices, the network operators usually manage MEC locally. The generic computing resources within the mobile edge hosts are virtualized and are exposed via application program interfaces (APIs). In this way, both users and operator applications can access it.

3. Secure Service Deployment Framework with Commercial MEC

In this section, we sort out several proposed frameworks about secure service deployment at first. And then, we put forward ours. The relative architectures and their features are listed in Table 1 [32].

3.1. Surveys on the Proposed Framework. Deng et al. [33] proposed a scheme of computation offloading to solve the scalability problems. The utility function of users is to make unloading decisions in turn according to the current interference environment and adjust the number of unloading users according to the estimated delay. Therefore, researchers developed the offloading interaction among multiple users as a sequential offloading decision game to solve the problems of scale ability. Users' utility in terms of experienced wait times and energy consumption is huge. The mobile users make the uninstall decision which is based on the following sequence in the current interference environment and adjusts the offloading users based on the estimated delay. They proposed a way called Nash. A Nash equilibrium is a state of a noncooperative game where no player can improve its utility by changing its strategy if the other players maintain their current strategies. The mobile device users at the equilibrium can achieve a mutually satisfactory solution and no user has the incentive to unilaterally deviate. Dynamic games which are equivalent with perfect information have a pure strategy Nash equilibrium. As the number of users increases, the proposed algorithm offloads the selection task to ensure the user experience. And the network made up of N small cells is shown in Figure 2; mobile devices upload data to MEC servers through nodes which are on the edge networks. One MEC server can provide service for many terminal devices so that operators can save cost-effectively.

Due to computation offloading, extralatency, and network load, Verbelen et al. [34] presented algorithms to partition a software application, composed of a number of components which has four parts in the cloud with different capacities while minimizing the communication cost between the components. They presented a multilevel KL-based algorithm as a fast partitioner. It allows real-time deployment calculations. The solution quality is improved by simulated annealing, but the cost is computation capacity. They used the way which is called computing the graph partitioning problem to assign computation offloading. Then, their goal is to execute all these tasks as fast as possible, thus minimizing the execution time of the slowest node.

Xiang et al. [35] proposed a joint offloading framework, which uses the characteristics of multiple applications to bundle offloading requests of code, thus saving additional energy. By sending code offloading requests in the form of bundles, the time for network interfaces to maintain a high

Table 1: Comparisons of frameworks based on different ways.

Properties	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]
Minimum execution time	Y	N	Y	N	N	N	N	N/A	N/A
Network latency	L	H	N/A	L	L	M	L	N/A	N/A
Transmission delay	M	H	H	N/A	N/A	M	M	N/A	N/A
Preexecution delay	H	N/A	H	L	N/A	N/A	H	L	H
Maximum privacy and security	N/A	N	N	N	N	Y	Y	N/A	N/A
Offloading overhead	H	H	H	N/A	N/A	H	H	N/A	N/A

Y = yes; N = no; H = high; M = medium; L = low; N/A = not applicable.

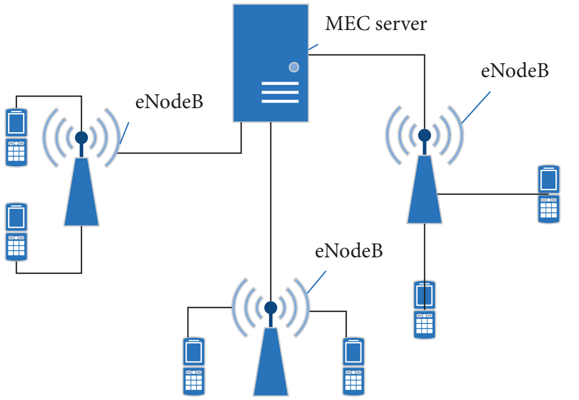


Figure 2: The scenario of multicell MEC.

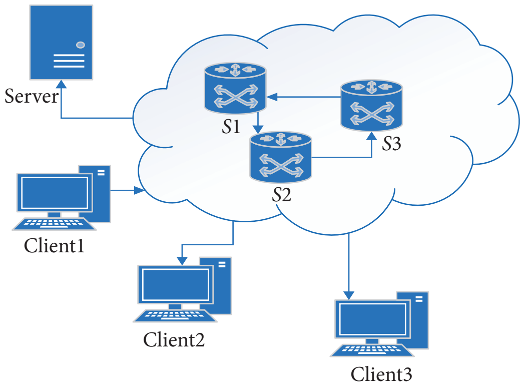


Figure 3: A communication instance of CDN-SDN.

power state is reduced, thus saving energy on mobile devices. The joint offloading problem is reduced to a joint optimization problem aiming at minimizing the response time and energy cost. Although the middleware framework reduces interaction latency, other elements, such as consumption graph modeling, optimal segmentation algorithms, two-step dynamic partitioning analysis, and intensive configuration, consume computing resources for mobile devices. Therefore, the computation-intensive nature of the framework increases the overall execution time of the application, hindering the vision of achieving seamless application execution.

Because many researchers have proposed their own frameworks, further comparisons of some frameworks which are based on different ways are given in Table 1.

3.2. A Framework of Commercial MEC. The idea of content delivery network (CDN) was first proposed in 1998, in which the content would be replicated on several proxy servers that were geographically closer to the user, as shown in Figure 3. As shown in Figure 3, each client will store the content that it originally requested from the server/controller, which will be provided to neighboring clients when the same content is requested. Distributing servers in multiple locations is the most common way to promote high performance and scalability [36]. CDN still faces the problems of limited resources as servers cannot meet the increasing demands of users. The deployment of servers depends on the location, such as in order to find the right location with the required capacity and then invest in the cost of deployment and installation [37].

Now video clips apps and major video websites are popular. As a result, users' habits of watching videos have also changed. Although all major video websites have chosen CDN services to provide users with a good video viewing experience, the current user interface architecture of a mobile communication network still has some inherent flaws. For example, users in the same city or county have the same requests such as watching the same film or downloading a document. All the video content needs to access the CDN service nodes of the video website on the backbone network through the provincial core network exits, which brings huge bandwidth pressure on the backhaul link.

Service providers can set up CDN at the mobile edge so that edge CDN can store popular videos. The differences between traditional CDN and CDN based on MEC are shown in Table 2. By requesting the hot content stored in the edge CDN, the hot content can be sent to users from the edge of the mobile network without the need to transmit the content from the central CDN node through the mobile core network. The hot content caching mechanism should be predefined or support dynamic updates to meet user requests. As shown in Figure 4, mobile devices usually need to transfer data from base station to MEC. With MEC and edge CDN, devices can process data quickly instead of processing data from central CDN or cloud. In this way, data processing will be greatly accelerated and data transfer can speed up.

4. Challenges of Secure Service Deployment

With the secure deployment of commercial MEC, many challenges need to be solved. Table 3 summarizes some challenges and solutions [38].

Table 2: Comparison between CDN based on MEC and traditional CDN.

Comparison	Traditional CDN	CDN based on MEC
Geographical location	Far from users	Closed to users
Receiving and sending resources	Weak ability	Strong ability
Coverage area	Small coverage area	Large coverage area
Kinds of service	Few kinds	More kinds
Cost	Low	High

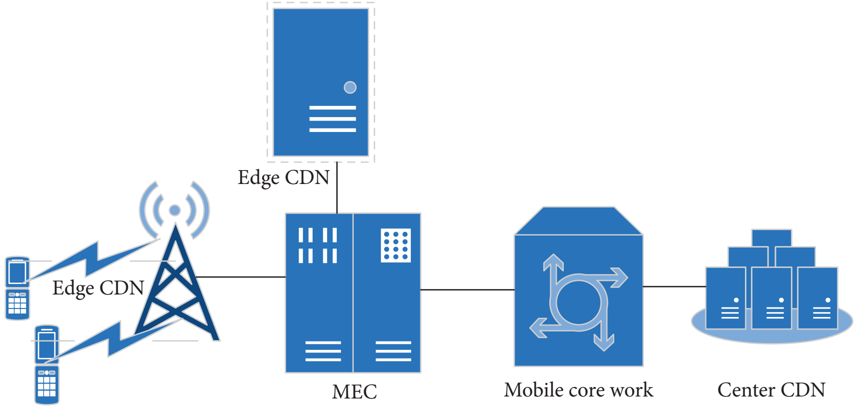


Figure 4: An edge CDN framework based on MEC.

Table 3: Challenges and methods of commercial MEC.

Challenges	Reference	Method used	Contribution
Risk of user data leakage	[42]	A VLAN based on security architecture	Increase protection to prevent accidental data leakage
Secure risk of data transmission	[50]	2D-DWT-1L or 2D-DWT-2L steganography	Enable to hide the confidential patient's data and transfer data secretly
Security of terminal device	[51]	A terminal lightweight anonymous security communication scheme	Support access authentication for massive terminal devices

4.1. Risk of User Data Leakage. In the course of doing business with commercial MEC, sometimes sensitive data must be handed over to supposedly third parties. In this background, it is necessary for researchers to propose a method to detect when the distributor's sensitive data have been leaked and fill this loophole.

Vaidya and Khobragade [39] used an algorithm which is called RSA encryption technology to ensure the security of user data. RSA can ensure coded data through distributed verification. The researchers used the method of reserving the RSA token properties so that they can address the problem of ensuring cloud data storage correction. Considering that the key calculation function belongs to a universal hash function family, researchers choose to store RSA technology, which can be completely integrated with the verification of erasure-coded data. Then, it shows how to verify the correctness of the storage and determine if the server is behaving abnormally.

Yu et al. [40] presented a data leakage prevention model called CBDLP. CBDLP consists of two parts, one is the training phase and the other is the detection phase. During the training phase, the training documents are divided into different clusters. In the detection phase, the documents are matched with the cluster diagram, respectively. So far, a number of specified commercial DLP solutions have reduced the risk of most accidental leaks [41].

4.2. Secure Risk of Data Transmission. For commercial MEC, the information transmission security is crucially important. The hackers take the IP addresses illegally so that they can impersonate other legitimate users to affect the security and stability of communication data transmission. Sometimes, the hackers send a large number of instructions and data to the terminal of the mobile device, which makes the communication network appear to be blocked negatively. Due to data transmission via the wireless network, the hackers analyze the frequency so that they can complete wiretapping work [42]. In severe cases, communication data will be tampered by hackers, causing a negative phenomenon of data loss.

To solve similar problems, Papadimitratos et al. [43] proposed an overview of the secure message transmission (SMT) protocol. SMT is used to establish a security association (SA) between the two terminal communication nodes: the source and the destination. Since the related nodes are chosen to adopt a secure communication scheme, the authentication capability between them is essential. For example, the trust relationship can be instantiated by knowing the public key at the other end of the communication. However, no terminal node needs to be safely associated with any remaining network nodes. Therefore, SMT does not need to perform encryption operations on these intermediate nodes.

Okaya and Ozdemir also proposed a novel named secure data aggregation (SDA) protocol for fog computing-based SGs (FCSG). Through employing homomorphic encryption, the proposed protocol not only ensures data privacy but also reduces a large of data which is stored in the cloud servers. Moreover, having related servers reduces server response time and creates less data traffic compared to cloud-based smart grids (SGs) [44].

4.3. Security of Terminal Device. At present, most users of mobile cloud services use cloud services without security protection. For example, private data such as user address books, text messages, and memos of mobile terminal devices are directly synchronized with the cloud platform by default. These private data are in the cloud, so that operators can call users' data easily on the platform. With the widespread use of mobile terminal applications, operators can easily capture the user's location information which not only includes the user's closest geographic location but also deduce the user's potential location privacy. It is dangerous for personal privacy.

Researchers proposed a dynamic path quorum system for mobile hoc networks and designed a dynamic path quorum generation algorithm. And they proposed a distributed access control mechanism for mobile hoc networks based on the quorum system, which is different from the traditional one depending on a single node itself. Compared with the access control mechanism, this access control mechanism has the stronger antiattack ability and higher reliability and effectively improve the resource sharing and protection level of the mobile hoc network [45].

5. Open Issues and Challenges

According to the research studies and experiments that have been discussed above, a few crucial open issues on secure deployment of mobile services in edge computing are concluded.

5.1. Privacy Security. Although the hype around MEC tends to encourage people to think that it is a universal panacea, promoters usually ignore the privacy security caused by the MEC. When users use services provided by MEC, their location information may be exposed. For example, the popularity of in-vehicle MEC may lead to misuse of vehicle location information. The service provider may monitor the user's trajectory without being allowed by users [46]. So the privacy security of commercial MEC must be solved urgently. Although many existing studies treat that information security and information privacy threats separately, we believe that only studying information privacy is not enough and there is a lot of related work to be done [47].

5.2. Data Transfer. The evolution of new services and the growth of information on the Internet has caused the origin of ideas, concepts, and paradigms. However, traditional network infrastructure requiring advanced network policies and configuration protocols are inefficient. And it supports significant limitations, high levels of scalability, and high amount of traffic [48]. As known to all, 5G is a key driver of MEC. It means that speed and stability of transfer play an important role in MEC. However, due to different geographical locations, receiving, and sending equipment, high-quality service of MEC is hard to be ensured. In other words, the transfer of data is required to promote the joint optimization of commercial MEC.

5.3. Access Control. Due to the outsourcing feature of edge computing, if there are no effective authentication mechanisms, any malicious users with an unauthorized identity may abuse the service resources at the edge. This leads to a huge security challenge for secure access control systems. For example, virtualized resources of edge server clouds can be accessed and modified by edge devices if they have certain privileges [49].

6. Conclusion

Commercial MEC will play an important role in daily life in the near future. MEC has excellent business prospects. It has a great influence on the society. Operators can combine different industry application scenarios, mature 4G networks, and stronger 5G networks to actively practice the deployment and application of MEC. Predictably, when the commercial MEC framework is completed, it is of great benefit to the city and people.

In this paper, a comprehensive and detailed survey on secure deployment of mobile services in edge computing is presented. Firstly, this paper reviews the basic driving force of conducting the survey about MEC. And then, the related concepts and definitions are introduced. Afterward, this paper provides an overview of frameworks and crucial techniques. Finally, several open issues are enumerated to guide our future research directions. In a word, this survey is presented to promote further progress of commercial MEC.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] N. Abbas, Y. Zhang, Y. Taherkordi, and T. Skeie, "Mobile Edge Computing: a Survey architecture, applications, approaches and challenges," *IEEE Internet of Things Journal*, vol. 5, pp. 454-455, 2018.

[2] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, and L. Qi, "Trust-oriented IoT service placement for smart cities in edge

computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4084–4091, 2020.

[3] A. B. Ericsson, "Ericsson mobility report 2017," 2017.

[4] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2020.

[5] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, 2019.

[6] T. D. Nguyen, E. N. Huh, and M. Jo, "Decentralized and revised content-centric networking-based service deployment and discovery platform in mobile edge computing for IoT devices," *IEEE Internet of Things Journal*, vol. 6, pp. 6142–6175, 2019.

[7] Y. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing: a key technology towards 5G," *ETSI (European Telecommunications Standards Institute)*, vol. 9, 2015.

[8] R. Solozabal, "Exploitation of mobile edge computing in 5G distributed mission-critical push-to-talk service deployment," *IEEE Access*, vol. 6, 2015.

[9] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned Internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, 2020.

[10] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, pp. 1628–1656, 2017.

[11] Y. Zhu, Y. Hu, and A. Schmeink, "Delay minimization offloading for interdependent tasks in energy-aware cooperative MEC networks," *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 15, pp. 1–6, 2019.

[12] C. You, K. Huang, H. Chae, and B. H. Kim, "IEEE transactions on wireless communications," *IEEE*, vol. 16, pp. 1397–1411, 2017.

[13] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, pp. 4924–4938, 2017.

[14] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, and A. WolmanVolley, "Automated data placement for geo-distributed cloud services sharad," *Microsoft Research*, vol. 16, 2010.

[15] X. Xu, B. Shen, X. Yin et al., "Edge server quantification and placement for offloading social media services in industrial cognitive IoV," *IEEE Transactions on Industrial Informatics*, vol. 16, 2020.

[16] X. Xu, S. Fu, L. Qi et al., "An IoT-Oriented data placement method with privacy preservation in cloud environment," *Journal of Network and Computer Applications*, vol. 124, pp. 148–157, 2018.

[17] B. Lin, F. Zhu, J. Zhang et al., "A time-driven data placement strategy for a scientific workflow combining edge computing and cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4254–4265, 2019.

[18] W. Li, P. Svard, J. Tordsson, and E. Elmroth, "A general approach to service deployment in cloud environments," 2012.

[19] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: state-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, 2015.

[20] N. Handigol, "Where is the debugger for my software-defined network?" 2015.

[21] E. Ahmed and M. H. Rehmani, "Mobile edge computing: opportunities, solutions, and challenges," *Future Generation Computer Systems*, vol. 23, 2016.

[22] H. Li, G. Shou, Y. Hu, and Z. Guo, "Mobile edge computing: progress and challenges," *Future Generation Computer Systems*, vol. 5, p. 3, 2016.

[23] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Aiolos: middleware for improving mobile application performance through cyber foraging," *Journal of Systems and Software*, vol. 85, no. 11, pp. 2629–2639, 2012.

[24] X. Zhang, S. Jeong, A. Kunjithapatham, and S. Gibbs, "Towards an elastic application model for augmenting computing capabilities of mobile platforms," 2010.

[25] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: dynamic resource allocation and parallel execution in the cloud for mobile code offloading," 2012.

[26] E. Koukoumidis, D. Lymberopoulos, K. Strauss, J. Liu, and D. Burger, "Pocket cloudlets," *ACM SIGARCH Computer Architecture News*, vol. 39, no. 1, pp. 171–184, 2011.

[27] A. Dou, V. Kalogeraki, D. Gunopulos, T. Mielikainen, and V. H. Tuulos, "Misco: a mapreduce framework for mobile systems," 2010.

[28] D. Kovachev, Y. Cao, and R. Klamma, "Augmenting pervasive environments with an xmpp-based mobile cloud middleware," in *Mobile Computing, Applications, and Services*Springer, Berlin, Germany, 2010.

[29] D. Kovachev, T. Yu, and R. Klamma, "Adaptive computation offloading from mobile devices into the cloud," 2012.

[30] S. Goyal and J. Carter, "A lightweight secure cyber foraging infrastructure for resource-constrained devices," 2004.

[31] D. Fesehaye, Y. Gao, K. Nahrstedt, and G. Wang, "Impact of cloudlets on interactive mobile cloud applications," 2012.

[32] K. Dolui and S. K. Datta, "Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing," 2017.

[33] M. Deng, H. Tian, and X. Lyu, "Adaptive sequential offloading game for multi-cell Mobile Edge Computing," 2016.

[34] T. Verbelen, T. Stevens, F. De Turck, and B. Dhoedt, "Graph partitioning algorithms for optimizing software deployment in mobile cloud computing," *Future Generation Computer Systems*, vol. 29, no. 2, pp. 451–459, 2013.

[35] L. Xiang, S. Ye, Y. Feng, B. Li, and B. Li, "Ready, Set, Go: coalesced offloading from mobile devices to the cloud," *IEEE*, vol. 7, p. 8, 2014.

[36] J. D. Gagliardi, T. S. Munger, and D. W. Ploesser, "Content Delivery network," *U.S. Patent*, vol. 8, 2012.

[37] J. Chandrakanth, P. Chollangi, and C. H. Lung, "Content distribution networks using software defined networks," *IEEE*, vol. 11, p. 30, 2015.

[38] S. Shahzadi, "Multi-access edge computing: open issues, challenges and future perspectives," *Journal of Cloud Computing*, vol. 12, p. 21, 2017.

[39] C. Vaidya and P. Khobragade, "Data security in cloud computing," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 12, 2015.

[40] X. Yu, Z. Tian, J. Qiu, and F. Jiang, "A data leakage prevention method based on the reduction of confidential and context terms for smart mobile devices," *Wireless Communications and Mobile Computing*, vol. 23, 2018.

[41] R. Rauscher and R. Acharya, "A network security architecture to reduce the risk of data leakage for health care organizations," 2014.

[42] X. Xu, D. Zhu, X. Yang, S. Wang, L. Qi, and W. Dou, "Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain," *ACM Transactions on Internet Technology*, vol. 23, 2020.

[43] P. Papadimitratos and Zygmun, "Secure data transmission in mobile ad hoc networks," *ACM Workshop on Wireless Security*, vol. 9, pp. 41–54, 2013.

[44] F. Y. Okay and S. Ozdemir, "A secure data aggregation protocol for fog computing based smart grids," *IEEE*, vol. 7, p. 7, 2018.

[45] R. Li, X. Dong, X. Gu, W. Zhou, and C. Wang, "Overview of the data security and privacy-preserving of mobile cloud services," *Journal on Communications*, vol. 12, 2013.

[46] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, "Joint optimization of offloading utility and privacy for edge computing enabled IoT," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2622–2629, 2020.

[47] E. Robert, "The mobile privacy-security knowledge gap model: understanding behaviors," *Destination Area: Integrated Security (IS)*, vol. 4, 2017.

[48] G. A. Mensah, C. O. Johnson, G. Addolorato et al., "Global burden of cardiovascular diseases and risk factors, 1990-2019: update from the GBD 2019 study," *Journal of the American College of Cardiology*, vol. 34, 2020.

[49] J. Zhang, Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: survey and open issues," *IEEE Access*, vol. 6, 2018.

[50] M. Elhoseny, "Secure medical data transmission model for IoT-based healthcare systems," *IEEE Access*, vol. 3, p. 21, 2018.

[51] L. Chen, Z. Liu, and Z. Wang, "Research on heterogeneous terminal security access technology in edge computing scenario," *2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, vol. 10, p. 7, 2019.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8843803, 11 pages
https://doi.org/10.1155/2020/8843803

Research Article

A Group Recommendation System of Network Document Resource Based on Knowledge Graph and LSTM in Edge Computing

Yuezhong Wu,[1,2] Qiang Liu,[1,2] Rongrong Chen,[3] Changyun Li,[1,2] and Ziran Peng[1]

[1] *College of Computer Science, Hunan University of Technology, Zhuzhou 412007, China*
[2] *Intelligent Information Perception and Processing Technology Hunan Province Key Laboratory, Zhuzhou 412007, China*
[3] *College of Business, Hunan University of Technology, Zhuzhou 412007, China*

Correspondence should be addressed to Qiang Liu; liuqiang@hut.edu.cn and Changyun Li; lichangyun@hut.edu.cn

Received 4 August 2020; Revised 12 October 2020; Accepted 11 November 2020; Published 2 December 2020

Academic Editor: Xiaolong Xu

The Internet has become one of the important channels for users to obtain information and knowledge. It is crucial to work out how to acquire personalized requirement of users accurately and effectively from huge amount of network document resources. Group recommendation is an information system for group participation in common activities that meets the common interests of all members in the group. This paper proposes a group recommendation system for network document resource exploration using the knowledge graph and LSTM in edge computing, which can solve the problem of information overload and resource trek effectively. An extensive system test has been carried out in the field of big data application in packaging industry. The experimental results show that the proposed system recommends network document resource more accurately and further improves recommendation quality using the knowledge graph and LSTM in edge computing. Therefore, it can meet the user's personalized resource need more effectively.

1. Introduction

With the popularity of the Internet, network resources have become people's first choice to find information. As a kind of special resources of the Internet, the rapid growth of network document resources makes the problem of "information overload" and "resource trek" increasingly serious, preventing people from collecting and obtaining information efficiently. For example, there will be more than 19 million query results when the keyword "recommendation system" is given in Baidu Library. Massive and excessive information will be presented at the same time, which makes it difficult for people to make correct and efficient choices and obtain the resources they really look for. As an essential means of information filtering, the recommendation system is one of the most effective methods to solve the current "information overload" and "resource trek" problems [1]. However, most existing recommendation systems support a single user. In recent years, with the development of social networks and online communities, under the environment of "human beings are social animals," users with similar interests form groups and participate in practical activities. Group recommendation systems have been successfully applied to learning [2], academic knowledge [3], audio and video services [4], travel [5], communications [6], and other fields. Group recommendation systems have gradually become one of the research hotspots in the field of recommendation systems. Methods, theories, and applications of group recommendation systems have been studied in depth abroad, while research on group recommendation systems has just begun in China. In a group recommendation system, the determination of recommendations depends on the selected preference fusion strategy.

With the deepening of the era of big data [7, 8], the application of deep learning [9] and the knowledge graph [10] in the recommendation system has been paid increasingly attention by academics and industry. For one thing, the research on the recommendation system based on deep learning has become a hot research topic. Deep learning applied to the mining of comment text corpus can effectively improve the recommendation accuracy. For another, the knowledge graph can better enrich and represent the semantics of resources and provide more comprehensive and relevant information. The emergence of the knowledge graph provides an effective way to design recommendation systems in big data environments. It can enhance the semantic accuracy of the data to further improve recommendation accuracy and can solve the data sparsity problem of recommendation technology as well.

However, the current personalization of the recommendation system is expressed by the behavior of the user interacting with the item as a feature, but the user's behavior actually occurs on the client. The recommendation system model wants to get the user's behavior characteristics. When the data on the end are sent to the server, there will be a delay problem. Due to the delay in real-time perception of user's behavior, the resources obtained by the user cannot match the changes in the user's interest timely. Edge computing [11] has real-time perception and real-time feedback, which can solve the problem of insufficient real-time perception and real-time feedback capabilities of the current client-server architecture recommendation system.

Based on the traditional recommendation technology, a group recommendation system for network document resource discovery based on the knowledge graph and LSTM in edge computing is proposed, which is able to work out the target information in accordance with users' needs proactively and solve the "information overload" and "resource trek" problem as well. The main contributions of this paper are as follows:

(1) A group recommendation system based on the knowledge graph and LSTM in edge computing is proposed. For processing data through LSTM in edge computing, the proposed system combines group recommendation, collaborative filtering-based recommendation, and content-based recommendation based on knowledge. The recommendation results are all individually adjustable, and they meet the real users' need accordingly between the group of similar interest and single user, which undoubtedly makes the accuracy of the recommendation better.

(2) Recommendation quality is improved. The proposed system takes advantage of group recommendation and the knowledge graph to make up for sparsity of recommendation and increase the recommendation precision rate and recall rate. Therefore, the proposed system provides practical value for personalized recommendation systems of network document resources.

(3) The system test has been accomplished in the field of big data application in the packaging industry. The experimental results suggest that the group recommendation results can meet the real user's need with higher efficiency.

The remainder of the paper covers the background and related work discussion (Section 2), the preliminaries of the group recommendation system, and detailed illustration (Section 3), the design of the group recommendation system, and detailed illustration (Section 4), the experiments and test results (Section 5), and the conclusions and future work (Section 6).

2. Related Work

2.1. Recommendation System. With the accelerating loading of the Internet information, serious "information overload" and "resource trek" problems have been emerging. The recommendation system has received wide attention as a solution in academia and business. The recommendation system is a subset of the information-filtering system that predicts the user's possible preferences and recommends to users, based on user preferences, habits, personalized needs, and characteristics of information or objects (such as movies, TV shows, music, books, news, photos, and web pages), and it helps users to make quick decisions and improves user satisfaction [12, 13]. In recent years, with the continuous development of the recommendation system, according to the different selection methods, there are some recommended algorithms: demographic-based recommendation [14], content-based recommendation [15], collaborative filtering-based recommendation [16], knowledge-based recommendation [17], model-based recommendation [18], association rule mining for recommendation [19], social-based recommendation [20], hybrid recommendation [21], group recommendation [22], and so on. With the increasing application forms and scenarios of the recommendation system, the research and application of the recommendation system face some important issues, such as cold start problem, user niche problem, and personalized recommendation interpretability problem. The existing research focuses on constructing a personalized recommendation service based on a data model that reflects the user's interest characteristics. Hu [23] proposed a recommendation algorithm based on user interest and the topic model to solve the problems of data sparsity, cold start, and user interest acquisition. Hu et al. [24] proposed an enhanced group recommendation method based on preference aggregation, incorporating simultaneously the advantages of the aforesaid two aggregation methods, and effectively improved recommendation accuracy. The authors [25–27] all proposed to satisfy the user's preference, rely on the user's own attributes to make recommendations based on utility, and apply them in the recommendation of papers, music, and electronic products. The goal is to maximize the user's interests, improve the accuracy, and ensure the quality of recommendation services.

2.2. Deep Learning. Deep learning-based recommendation methods can incorporate multisource heterogeneous data for recommendation, including explicit or implicit feedback data from users, user portrait and project content data, and user-generated content. Deep learning methods use multisource heterogeneous data as input and use an end-to-end model to automatically train prediction models, which can effectively integrate multisource heterogeneous data into the recommendation system, thereby alleviating the data sparseness and cold start in traditional recommendation system problems and improving the ability of the recommendation system. The application of deep learning to corpus mining is a research hotspot. After 2006, with the publication of Hinton, it was wildly sought after by scholars in the artificial intelligence world. This model is based on a neural network model, but it is more complex than a simple neural model, and the problems it deals with are more complex and diverse. Deep learning methods have been successfully used in many applications in the computer field, including speech recognition, speech search, natural language understanding, information retrieval, and robotics. Mokri et al. [28] applied the neural network model to have a high degree of relevance. It retrieved Slovak-related documents, processed keyword parts of speech, and greatly improved accuracy and recall. Based on the highly nonlinear characteristics of neural network algorithms, using the BP network to optimize the weight of each parameter in the entire neural network, constantly revising the weights, Xu et al. [29] constructed a personalized behavior based on users, the information retrieval model. Guezouli [30] used the correlation characteristics of neighbor nodes in the neural network to combine all documents into a neural network and retrieves the most relevant document according to query. Wang et al. [31] designed a data collection and preprocessing scheme based on deep learning, which adopted the semisupervised learning algorithm of data augmentation and label guessing. The recommendation system collects and processes the data by using deep learning to improve the accuracy of recommendation. Therefore, it has important research significance and practical value, and becomes one of the most active branches on the research recommendation system.

2.3. Knowledge Graph. The application of the knowledge graph is coherently born to enrich and represent the semantics of resources. It was proposed by Google in 2012 to describe the various entities or concepts that exist in the real world and the incidence relation between them. The knowledge graph is not a substitute for ontology. Ontology describes the data scheme of the knowledge graph, namely, for knowledge graph building, a data schema is equivalent to establishing its ontology. The knowledge graph based on ontology enriches and expands, and the expansion is mainly embodied in the entity level. The knowledge graph is more accurate to describe the incidence of various relationships in the real world. The knowledge graph is a great promoter of the semantic annotation of digital resources and promoting the efficient acquisition of knowledge and information. At present, Google, Sogou cubic, Baidu bosom, Microsoft Probase, etc. already preliminarily applied the knowledge graph system in the industry. Most of them are the general knowledge graph, which emphasizes the breadth of knowledge and includes more entities. It is difficult to have a complete and global ontology layer for unified management and mainly used in the search services business with no need for high accuracy requirements. There are some industry knowledge graphs which have high accuracy requirements, used for auxiliary complex decision support, the rich and strict data patterns, etc. The authors [32, 33] reviewed knowledge graph technology in academia. Hu [34] researched on the construction of the knowledge graph based on the application. Li et al. [35] proposed an automatic knowledge graph establishment method and established a knowledge graph of the packaging industry. Chang et al. [36] summarized the application of a knowledge graph in the recommendation system. To seek semantics support for searching, understanding, analyzing, and mining, Wu et al. [37] proposed a more convenient way which is based on the domain knowledge graph to annotate network documents automatically. Jiang et al. [38] focused on graph-based trust evaluation models in Online Social Networks (OSNs). The recommendation system based on the knowledge graph enhances the semantic information of the data by connecting users and users, users and items, and items and items to further improve the accuracy of recommendation. Therefore, it has important research significance and practical value, and gradually becomes one of the most active branches on the research recommendation system.

2.4. Edge Computing. In response to the difficulties faced by cloud computing, edge computing has been proposed as a new computing paradigm and has gradually become an emerging computing model that meets the needs of Internet of Everything applications. The edge devices in the edge computing model have computing and analysis capabilities and provide computing power support for application developers and service providers by performing calculations at the edge of the network. It uses a distributed computing architecture to sink major applications, services, and data storage to the edge of the network, thereby bringing computing closer to the source of data. It decomposes the large tasks originally processed at the central node into multiple smaller, more manageable subtasks, which are placed close to the data source or user service terminal to provide edge intelligent services nearby, thereby reducing the delay of network communication and service delivery, reducing cloud pressure, and generating faster network service response, to meet the industry's key requirements for real-time business, intelligent applications, security and privacy protection, and so on. The research of edge computing has received increasingly attention, which involves application fields such as smart education, smart manufacturing, smart transportation, and smart medical care, which has important research significance and practical value [39–45]. Jiang et al. [46] introduced the concepts and characteristics of cloud computing and fog computing, and compared the cooperations between cloud computing and fog computing. The advantage of edge computing is that edge nodes have the ability to "independently think," which makes some decisions and calculations no longer dependent on the cloud, and the end-side can give results in a more real-time and more strategic manner. Especially, with the advent of the 5G era, its low-latency feature greatly reduces the interaction time between the end and the cloud and is more conducive to us using end intelligence to achieve lower-cost decision-making and rapid response. The cloud and the end-side are more closely integrated, and the end-side can perceive the user's intention in seconds to make decisions. The recommendation system based on edge computing enhances service performance in the network low latency, real-time interaction, service stability, and security. Therefore, edge computing can promote the development of the recommendation system and become a research hotspot, which will have great application value in practical business.

3. Preliminaries

3.1. The Computation Model of Word Vector. TF-IDF is a very significant concept and method in the field of information retrieval and data mining. The figure of TF-IDF is inversely proportional to the time of the word that exists in the whole gathered document, and is proportional to the frequency that appears in the document. However, all the document sets is converged by all attribute characteristics of instances, including the basic attribute and the domain attribute. In the traditional TF-IDF model, it failed to reflect the contribution of the different attributes to instance word vectors. Hence, this paper advocates to calculate the word vector by the use of the upgraded TF-IDF model based on the contribution of the literature [37].

CTF is short for contribution of term frequency, which is defined as follows:

$$\mathrm{CTF}(w_i) = \frac{\sum_{j=1}^{n} C(w_{ij}) * W(\mathrm{Attr}_{ij})}{\sum_{j=1}^{m}\sum_{j=1}^{n} C(w_{ij}) * W(\mathrm{Attr}_{ij})}. \quad (1)$$

CIDF is defined as follows, which is short for contribution of inverse document frequency,

$$\mathrm{CIDF}(w_i) = \frac{\sum_{j=1}^{m}\sum_{j=1}^{n} W(\mathrm{Attr}_{ij})}{\sum_{j=1}^{n} W(\mathrm{Attr}_{ij}) + 1}. \quad (2)$$

In addition, this formula CTF(w_i) demonstrates the word frequency of the contribution for w_i, $C(w_{ij})$ represents the word frequency in the j attribute text for w_i, $W(\mathrm{Attr}_{ij})$ represents the weight of the j attribute, and the formula CIDF(w_i) demonstrates the inverse document frequency based on the contribution for w_i.

Calculate the figure of the upgraded TF-IDF:

$$W_i = \mathrm{CTF}(w_i) * \mathrm{CIDF}(w_i). \quad (3)$$

The network document refers to an article including information, new, paper, and so on. Its format can be structured in types such as TXT and XML. It can also be unstructured types such as WORD, PDF, and others. The network document is presented with an upgraded word model TF-IDF based on the contribution in this paper:

$$d = (w_{d1}, w_{d2}, \ldots, w_{dm}). \quad (4)$$

In this paper, the system has a user set, each user has own hobbies and interests. Interests are grouped by the subject. Users' interests are collected by both manual and automatic acquisition modes in this paper. Adding keywords by users themselves is the manual mode, while the automatic mode is that the system obtains keywords through processing user access records, achieving adaptive updates in the interactive process in edge computing, and putting these keywords into the word library. The user interest model is presented with an upgraded word model TF-IDF based on the contribution in this paper:

$$u = (w_{u1}, w_{u2}, \ldots, w_{un}). \quad (5)$$

3.2. LSTM. Long short-term memory (LSTM) [47] is an improved recurrent neural network (RNN). The block cell of LSTM is shown in Figure 1. Compared with traditional recurrent networks, LSTM has an additional unit state for storing long-distance information, which solves the problem of gradient dispersion caused by excessively long gradients; LSTM repeating modules have different structures, including four interactive layers and a special form of interaction; the specially designed gate structure in LSTM enables the model to decide to discard information, determine to update cells, and update cell status. Because of its design characteristics, LSTM is very suitable for modeling time-series data, such as text data. The LSTM model can better capture the long-distance dependencies because LSTM can learn what information to remember and what information to forget through the training process.

The memory unit module is composed of three "gate" structures: input gate, forget gate, and output gate, and a loop connection unit. The internal parameters of the LSTM unit structure can be expressed as follows: assume that at time t, the input of a memory unit module is x_t, the output is h_t, unit status is C_t, and then the forget gate, input gate, input conversion, unit status update, output gate, and hidden layer output of the memory unit module are shown by equations (6) to (11), respectively.

(1) Forget gate: it can choose to forget certain past information and decide what information should be discarded or retained:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right). \quad (6)$$

(2) Input gate: it can remember some information now and update unit status:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (7)$$

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C). \quad (8)$$

(3) Merge and update:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t. \quad (9)$$

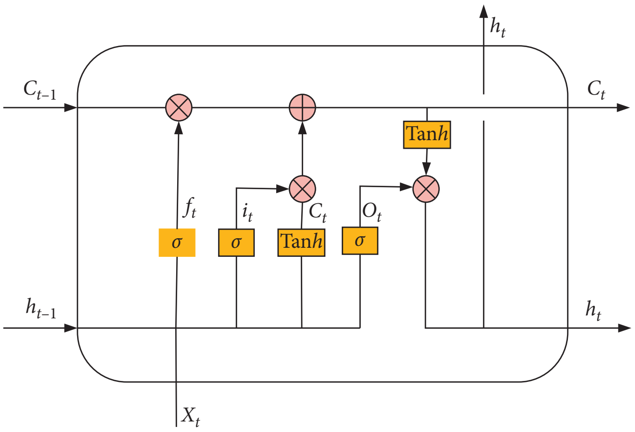


Figure 1: LSTM block cell.

(4) Output gate: it can determine the value of the next hidden state. The hidden state contains the related information of the previous input and can also be used for prediction:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (10)$$

$$h_t = o_t * \tanh(C_t), \quad (11)$$

where σ is the sigmoid function; tan$h$ is the hyperbolic tangent function; i_t, f_t, o_t, and $\tilde{C}_t$are the input of the input gate, forget gate, output gate, and input transformation to the unit; W_i, W_f, W_o, and W_c are weight matrices of input gate, forget gate, output gate, and input conversion corresponding to x_t and h_{t-1}; and b_i, b_f, b_o, and b_c are the offset vector of input gate, forget gate, output gate, and input conversion, respectively.

3.3. Knowledge Graph Construction Method. The framework of the knowledge graph construction method is shown in Figure 2. It includes the lifecycle of the domain knowledge graph, which mainly has five processes, namely, ontology definition, knowledge extraction, knowledge fusion, knowledge storage, and knowledge application, respectively. Each process has its own methods and tasks. For example, D2RQ is used to transform the atomic entity table and the atomic relation table into RDF in knowledge extraction; defined by the knowledge fusion rules to complete the knowledge fusion task while extracting knowledge with D2R and Wrappers, the tasks are such as entity merge, entity linking, and attribute merge.

In this paper, the authors obtain the semantic annotation knowledge graph. The semantic annotation helps the generation of sentence text and eliminates the ambiguity and ambiguity of natural language text. The entity in the knowledge graph can be used as a word segmentation dictionary. The semantics of entities, attributes, and relationships provide synonymy, inclusion, etc., and remove ambiguity and ambiguity, thus providing standard, concise, and comprehensive knowledge information.

4. The Design of Group Recommendation System

This paper designs a personalized group recommendation system of network document resources based on knowledge graph and LSTM in edge computing. Through the knowledge graph and LSTM in edge computing, the new meaning of the string is given, the document set is associated with the document feature, and the knowledge system related to the keyword is systematically made, so that the recommendation is superior in quality.

4.1. System Architecture. Based on the knowledge graph and LSTM, combining the content recommendation algorithm and collaborative filtering algorithm, this paper presents a group recommendation system of network document resources in edge computing, which has a five-data flow part comprising data collection, data mining, data fusion, data computing, and data application. Figure 3 shows the architecture of our proposed system.

It mainly includes the following parts:

(1) Data collection: data sources include user behavior data, user interest data, system historical data, resource evaluation data, and network data. These data are the basis for building knowledge maps and recommending. They need to be processed through the data mining part.

(2) Data mining: data cleaning and analysis on the collected data are performed. Corpus learning and entity naming process are carried out through LSTM. Then, the processed data are aggregated into the data fusion part.

(3) Data fusion: data from different data sources are processed for integration of heterogeneous data under the same framework specification and stored in different types of databases for use in the data calculation part.

(4) Data computing: the text classification process is run based on LSTM in edge computing, and the document set associated with document feature is got. Personalized recommendation results are obtained through user interest graph, topic association interest recommendation, semantic annotation-based content recommendation, and knowledge map-based group recommendation, and transmitted to the data application part.

(5) Data application: the recommended results are displayed to users according to the topic of network document resources. At the same time, relevant data are fed back to the data collection part.

4.2. The Description of Personalized Recommendation Algorithm. Performing corpus learning, entity naming, and text classification process through LSTM, the domain knowledge graph and the document set associated with document feature are ready. Then, go to recommendation.

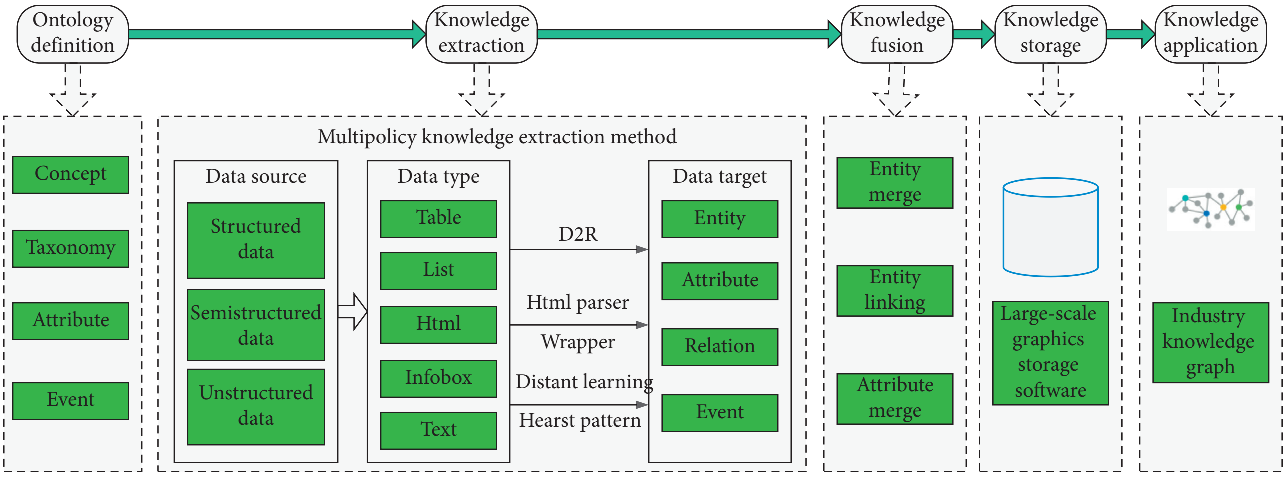


Figure 2: The framework of the knowledge graph construction method.

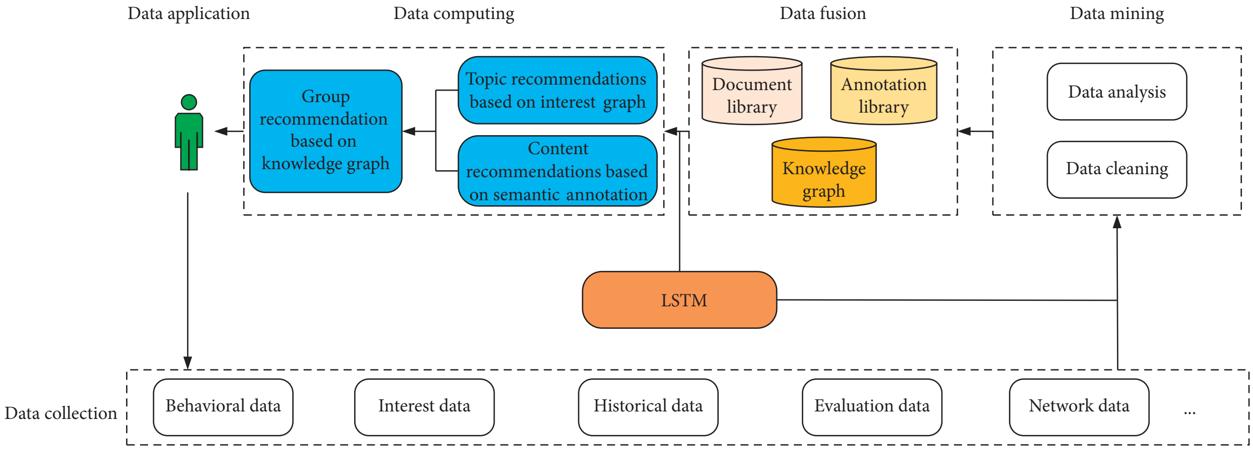


Figure 3: System architecture.

4.2.1. Topic Recommendations Based on Interest Graph. Through the user's interest graph, find other users associated with the user's interests, then combine the other users who have acted in the document what the target user has acted on, and form a similar interest user set U_1. At the same time, through the user's interest graph, the user's interest is extended to the topic layer, then perform the content-based recommendation, and remove the document what the target user has acted on, and obtain the corresponding document set L_1.

4.2.2. Content Recommendations Based on Semantic Annotation. In the process of constructing the domain knowledge graph, the documents and instances are semantically annotated to obtain the triplet < document, instance, and similarity > annotation library. Then, based on the user's attention graph instance, perform the content-based recommendation, remove the document what the target user has acted on, and obtain the corresponding document set L_2.

4.2.3. Group Recommendation Based on Knowledge Graph. *(1) Computing User Interest Similarity.* In the system, the user interest similarity is defined by *sim*, which is measured with the similarity between two user interest vectors q and d, seeing

$$sim(q,d) = \frac{\sum_{i=1}^{n} W_{i,q} W_{i,d}}{\sqrt{\sum_{i=1}^{n} W_{i,q}^{2}}\sqrt{\sum_{i=1}^{n} W_{i,d}^{2}}}, \tag{12}$$

where $w_{i,q}$ represents the weight of the interest keywords i in user q, $w_{i,d}$ represents the weight of the interest keywords i in user d, and n is the number of the keywords in the user interest set. The matrix of the user interest similarity is obtained by processing with cosine similarity calculation between two user interest vectors, see Table 1. The similar interest user set U_2 is obtained by computing the user interest similarity.

(2) Predicting User Document Behavior Evaluation. In the system, there is behavior evaluation between the user and the document. Six behavioral characteristics were selected as the users' interest in the document to participate in the prediction score, selecting the highest behavioral score. Implicit scoring principle [48] is used for reducing the degree of user participation in this paper. It marks 1 point when the user downloads the document; it marks 0.8 point when the user

Table 1: The matrix of two user interest similarity.

	u_1	u_2	u_3	u_4	u_5
u_1	1	0.2766	0	0.8620	0.1054
u_2	0.2766	1	0.3931	0.1144	0.2082
u_3	0	0.3931	1	0.6675	0.4932
u_4	0.8620	0.1144	0.6675	1	0.3704
u_5	0.1054	0.2082	0.4932	0.3704	1

Table 2: User document behavior evaluation.

	d_1	d_2	d_3	d_4	d_5
u_1	0.1	0.2	—	0.1	0.4
u_2	0.3	0.1	0.6	0.1	—
u_3	0.1	0.8	0.2	0.6	0.2
u_4	0.6	0.6	1	0.8	1
u_5	—	1	0.4	0.2	0.6

shares the document; it marks 0.6 point when the user comments the document; it marks 0.4 point when the user collects the document; it marks 0.2 point when the user clicks the document; it marks 0.1 point when the user only browses the document; it marks "/" when the user does not browse the document. All users and documents form a behavior evaluation matrix at the same time, see Table 2.

Based on K users who are similar to the target user's interest, find documents that K users like but the target user has not touched, predict the target user's interest in a document using equation (13), sort the documents according to the degree of interest, and get the document set L_3 finally:

$$P(u,i) = \sum_{v\in S(u,K)\cap N(i)} w_{uv} r_{vi}, \qquad (13)$$

where w_{uv} represents the interest similarity of two users u and v, r_{vi} represents the interest weight of the user v and the document i, $S(u,K)$ represents K users most similar to user u interests, and $N(i)$ represents having acted on document i.

(3) Performing Group Recommendation. There is a consensus function [49] between the user group and the document in the system. The consensus function quantifies the utility of the candidate item to the group from the degree of preference of the entire group to the project and the degree of preference difference between group members. The formalization of the group recommendation system is presented in this paper:

(1) Group prediction score: the predicted score $GP(G,i)$ of group G for document i is obtained by fusing the predicted score $P(u,i)$ of each user in the group:

$$GP(G,i) = \frac{1}{|G|} \sum_{u\in G} P(u,i) \qquad (14)$$

(2) Group divergence: the degree of divergence $dis(G,i)$ of the group to the document indicates the degree of difference in the prediction score of users of group G for document i:

$$di\,s(G,i) = \frac{1}{|G|} \sum_{u\in G} (P(u,i) - \text{mean}(G,i))^2, \qquad (15)$$

where mean(G,i) is the average of the prediction scores of users of group G for document i.

(3) Consensus function:

$$F(G,i) = w_1 \times GP(G,i) + w_2 \times (1 - di\,s(G,i)). \qquad (16)$$

Among them, w_1 and w_2, respectively, represent the weight of the group prediction score and the group disagreement in the consensus function, and $w_1 + w_2 = 1$, Algorithm 1

5. Experiment and Evaluation

In order to verify the feasibility of the proposed system and its services, we conducted experiments in the big data knowledge graph platform in the China packaging industry (URL: http://58.20.192.198:8090). The data in the experiments are collected from government information, business information, industry information, academic papers, global packaging patents, and other data resources, which add up to more than 3 million articles. We choose 28150 document resources for the experiment. Offline experiments aimed at 10 users and pretreated their web usage access logs.

5.1. Experiment Environment Configuration. The experiment environment configuration is as shown in Table 3. We build a knowledge base of packaging knowledge graph covering information, policies, conferences, standards, papers, patents, companies, products, universities, institutions, and experts. The instances in the knowledge graph are stored in MongoDB via key values. The data of semantic annotation library are stored in ES in triples. Network document resource is also stored in ES.

5.2. Constructing Packaging Knowledge Graph. As is shown in Figure 4, a packaging knowledge graph [35]is constructed. For example, the knowledge graph includes the following basic concepts, namely, "packaging knowledge point," "company," "product," "organization," "patent," "paper," and "event." Major relations include "has product," "upstream," "downstream," "has patent," and "executive."

5.3. Algorithm Evaluation. In this paper, we adopt an evaluation method to calculate the precision rate, recall rate, F-measure value, and real-time interaction response time T.

The precision rate calculation formula is as follows:

```
Input: domain knowledge graph $KG$, users set $U$, $G$, document set $docs$, < $document\ name, instance, similarity$ > triple list
Output: recommendation document set
Processing the document set $docs$ through LSTM
for $i = 1$ to $n$ do
  computing user interest similarity $sim(q,d)$, and obtain a similar interest user set $U_2$
  for each user in users associated with the user's interests in $KG$ do
    topic recommendations based on interest graph to obtain document set $L_1$ and obtain a similar interest user set $U_1$
  end
  for each individual $ins$ in <$document, instance, similarity$> triplet list do
    content recommendations based on semantic annotation to obtain document set $L_2$
  end
  for each document in $docs$ the user $u$ has not acted on do
    for each user in users of the intersection of $U_1$ and $U_2$ having acted on document $j$ do
      predicting user document behavior evaluation $P(u,j)$
      for each user in $G$ having acted on document $j$ do
        computing the user group and the document consensus function $F(G,j)$ to obtain document set $L_3$
      end
    end
  end
  do
    the intersection of $L_1$, $L_2$ and $L_3$, then sort by Top-$k$
  return recommendation document set
end
```

Algorithm 1: Algorithm for the group recommendation based on the knowledge graph and LSTM in edge computing.

Table 3: Experiment environment configuration.

Name	Versions
Operating system	CentOS6.5
Java runtime environment	JDK 1.8.0_141
Application server	Apache Tomcat 9.0.16
Mahout	Mahout 0.9
TensorFlow	TensorFlow 1.11.0
Program development platform	IntelliJ IDEA 13.1.2
Data bases	MySQL5.6, mongodb-linux-x86_64–3.4.10 and elasticsearch-5.4.2

$$P = \frac{\text{the number of documents searched that users are interested in}}{\text{the total number of the documents searched}}. \tag{17}$$

The recall rate calculation formula is as follows:

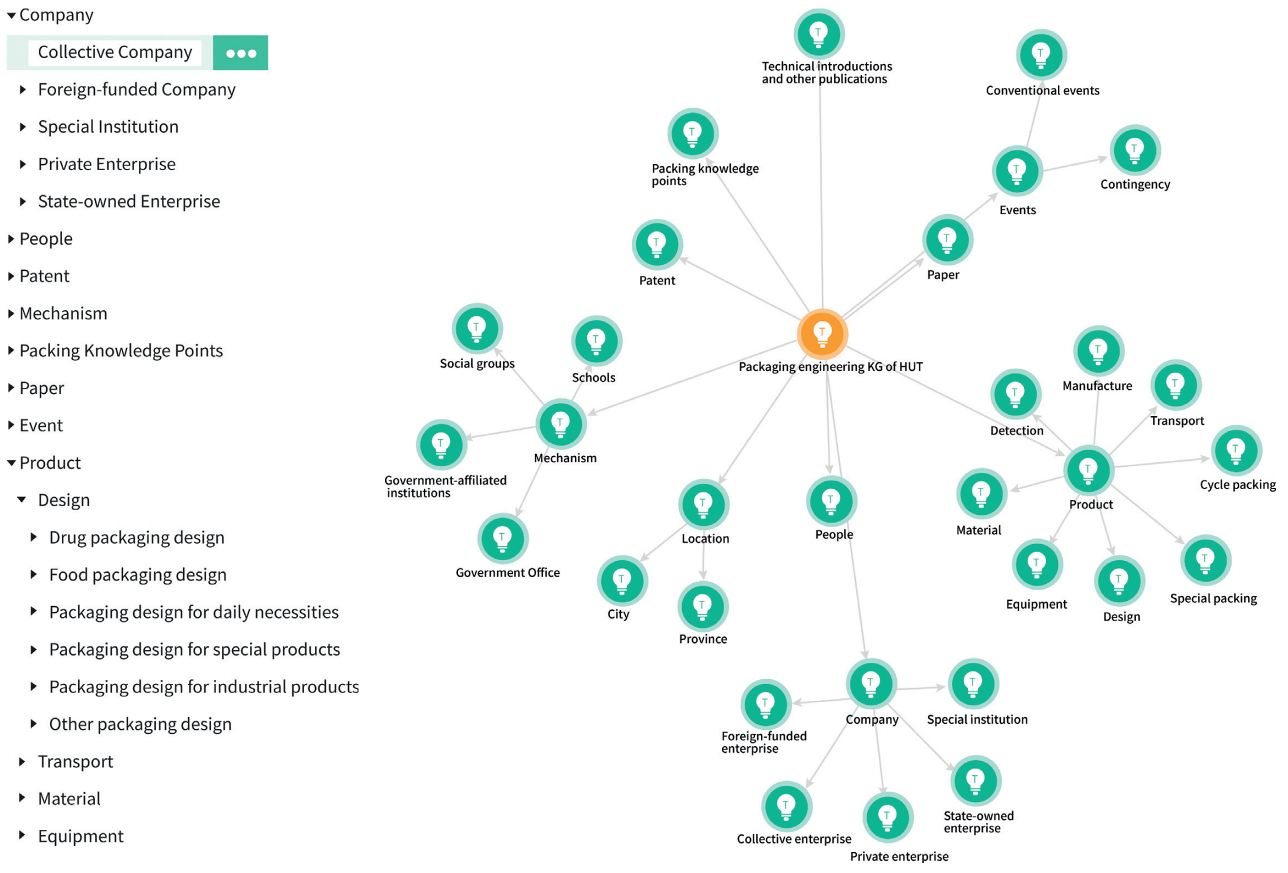
$$R = \frac{\text{the number of documents searched that users are interested in}}{\text{the total number of the documents related}}. \tag{18}$$

The F-measure value calculation formula is as follows:

$$F_1 = \frac{2 \times R \times P}{R + P}. \tag{19}$$

Through the system implementation, we put part of the data of the matrix of two user interest similarities and user document behavior evaluation in Table 1 and Table 2, and give recommendation results by using the traditional content-based recommendation, traditional collaborative filtering recommendation, personalized collaborative filtering recommendation based on knowledge graph [50], and proposed group recommendation based on knowledge graph and LSTM. The experimental results are as shown in Tables 4 and 5.

From the results of Table 4, we can see that the improved personalized group recommendation algorithm has higher precision rate, recall rate, and F-measure value among them, indicating that the use of domain knowledge graph and LSTM helps to enhance the semantic information of data and improve the quality of recommendation, and the group recommendation system effectively alleviates cold start problems. In conclusion, it is obvious that the



(a)

(b)

Figure 4: Packaging knowledge graph.

Table 4: Experimental results.

Algorithm	Precision rate	Recall rate	F_1 value
Traditional content-based recommendation	0.322	0.215	0.258
Traditional collaborative filtering recommendation	0.399	0.266	0.319
Personalized collaborative filtering recommendation based on knowledge graph	0.605	0.403	0.484
Proposed group recommendation based on knowledge graph and LSTM in edge computing	0.679	0.453	0.543

Table 5: Time test results.

Algorithm	Real-time interaction response time (ms)
Traditional content-based recommendation	217
Traditional collaborative filtering recommendation	231
Personalized collaborative filtering recommendation based on knowledge graph	390
Proposed group recommendation based on knowledge graph and LSTM in edge computing	95

personalized group recommendation system that we have proposed has higher accuracy as well as high availability in real systems.

From the results of Table 5, we can see that the improved personalized group recommendation algorithm has higher timely real-time interaction response time among them, indicating that the use of edge computing helps to enhance the quality of recommendation, because the end-side results can be presented in a more real-time and strategic manner. In conclusion, it is obvious that the personalized group recommendation system that we have proposed has higher interaction in real systems.

6. Conclusion

With the explosive growth of information on the Internet, the mining of massive multisource heterogeneous data is a key issue in the recommendation system. The emergence of the knowledge graph and deep learning brings a new opportunity for the integration processing of multisource heterogeneous data in the recommendation system. Therefore, the recommendation system based on the knowledge graph and LSTM in edge computing has become a new research field. In this paper, based on the packaging industry knowledge graph, the authors provide a technical implementation scheme for the group recommendation system of network document resources in edge computing, by joining the content-based recommendation and collaborative filtering-based recommendation algorithms. The proposed method is considering personalized demand between group and single user. The experimental results show that the proposed system improves the quality of recommendation.

The future research includes applying BiLSTM to mine and learn text eigenvector, optimizing the prediction algorithm based on deep learning model at the end-side in edge computing, improving the group recommendation algorithm, analyzing personalized recommendation reviews based on emotion weight, doing experiments in packaging evaluation corpus, and constructing a complete packaging big data recommendation system.

Data Availability

The data used to support the findings of this study are not applicable because the data interface cannot provide external access temporarily.

Disclosure

The work of packaging knowledge graph construction is published in Cyberspace Data and Intelligence and Cyber-Living Syndrome and Health. The authors have extended this research, and modified and optimized the algorithms.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program Funded Project of China under grant no. 2018YFB1700200, in part by the Hunan Provincial Key Research and Development Project of China under grant no. 2019GK2133, in part by the Scientific Research Project of Hunan Provincial Department of Education under grant nos. 19B147, 18K077, and 17C0479, in part by the Project of China Packaging Federation under funding support no. 17ZBLWT001KT010, and in part by the Intelligent Information Perception and Processing Technology Hunan Province Key Laboratory under grant no. 2017KF07.

References

[1] X. W. Meng, X. Hu, L. C. Wang, and Y. J. Zhang, "Mobile recommender systems and their applications," *Journal of Software*, vol. 24, no. 1, pp. 91–108, 2013.

[2] T. Xie, Q. Zheng, and W. Zhang, "A behavioral sequence analyzing framework for grouping students in an e-learning system," *Knowledge-Based Systems*, vol. 111, pp. 36–50, 2016.

[3] C.-H. Lai, "Applying knowledge flow mining to group recommendation methods for task-based groups," *Journal of the Association for Information Science and Technology*, vol. 66, no. 3, pp. 545–563, 2015.

[4] X. Zhou, L. Chen, Y. Zhang, and D. Qin, "Enhancing online video recommendation using social user interactions," *VLDB Journal*, vol. 26, no. 1, pp. 1–20, 2017.

[5] T. D. Pessemier, J. Dhondt, and L. Martens, "Hybrid group recommendations for a travel service," *Multimedia Tools & Applications*, vol. 76, no. 2, pp. 1–25, 2017.

[6] X. Y. Lin, Y. Yu, and Z. K. Mi, "A model of group mobility management based on user behavior prediction," *Telecommunications Science*, vol. 33, no. 8, pp. 94–99, 2017.

[7] A. McAfee and E. Brynjolfsson, "Big data: the management revolution," *Harvard Business Review*, vol. 90, no. 10, pp. 60–128, 2012.

[8] Nature, "Big data," 2012, http://www.nature.com/news/specials/bigdata/index.html(2012-10-02).

[9] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[10] S. Amit, *Introducing the Knowledge Graph*, America: Official Blog of Google, 2012.

[11] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[12] L. Chang, Y. T. Cao, W. P. Sun, W. T. Zhang, and J. T. Chen, "Review of tourism recommendation system research," *Computer Science*, vol. 44, no. 10, pp. 1–6, 2017.

[13] C. L. Huang, W. J. Jiang, J. Wu, and G. J. Wang, "Personalized review recommendation based on users' aspect sentiment," *ACM Transactions on Internet Technology (TOIT)*, vol. 20, no. 4, Article ID 42, 26 pages, 2020.

[14] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 393–408, 1999.

[15] A. Gunawardana and G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *Journal of Machine Learning Research*, vol. 10, no. 10, pp. 2935–2962, 2009.

[16] B. Ju, Y.-T. Qian, and M.-C. Ye, "Preference transfer model in collaborative filtering for implicit data," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no. 6, pp. 489–500, 2016.

[17] R. Burke, "Knowledge-Based recommender systems," *Encyclopedia of Library and Information Systems*, vol. 69, no. 32, pp. 180–200, 2000.

[18] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *Acm Transactions on Information Systems*, vol. 23, no. 1, pp. 103–145, 2005.

[19] J. B. Yuan and S. L. Ding, *Research and improvement on association rule algorithm based on FP-Growth*, vol. 7529, pp. 306–313, Springer, Berlin, Germany, 2012.

[20] X. W. Meng, S. D. Liu, Y. J. Zhang, and X. Hu, "Research on social recommender systems," *Journal of Software*, vol. 26, no. 6, pp. 1356–1372, 2015.

[21] Y. Wu, H. Huang, Q. Wu, A. Liu, and T. Wang, "A risk defense method based on microscopic state prediction with partial information observations in social networks," *Journal of Parallel and Distributed Computing*, vol. 131, pp. 189–199, 2019.

[22] I. García, S. Pajares, L. Sebastia, and E. Onaindia, "Preference elicitation techniques for group recommender systems," *Information Sciences*, vol. 189, no. 8, pp. 155–175, 2012.

[23] F. Y. Hu, *Research and Implementation of Hybrid Recommendation Algorithm Based on User Interest and Topic Model*, Beijing university of posts and telecommunications, Beijing, China, 2018.

[24] C. Hu, X. W. Meng, Y. J. Zhang, and Y. L. Du, "Enhanced group recommendation method based on preference aggregation," *Journal of Software*, vol. 29, no. 10, pp. 3164–3183, 2018.

[25] Y. Yin, D. Feng, and Z. Shi, "A personalized paper recommendation method based on utility," *Chinese Journal of Computers*, vol. 40, no. 12, pp. 2797–2811, 2017.

[26] H. S. Park, J. O. Yoo, and S. B. Cho, "A context-aware music recommendation system using fuzzy bayesian networks with utility theory," in *Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 970–979, Xi'an, China, 2006.

[27] N. Manouselis and C. Costopoulou, "marService: multiattribute utility recommendation for e-market," *International Journal of Computer Applications in Technology*, vol. 33, no. 2-3, pp. 176–189, 2008.

[28] I. Mokri and L. Skovajsova, "Neural network model of system for information retrieval from text documents in Slovak language," *Acta Electrotechnica et Informatica*, vol. 5, no. 1, p. 3, 2005.

[29] K. Y. Xu, S. Wang, S. Zhang, and G. Chang, "ANN-based personalized retrieval method," *Library and Information Service*, vol. 55, no. 2, pp. 59–63, 2011.

[30] L. Guezouli and A. Kadache, "Information retrieval model based on neural networks using neighborhood," in *Proceedings of the Information Technology and E-Services (ICITeS), 2012 International Conference*, pp. 1–5, IEEE, Sousse, Tunisia, 2012.

[31] T. Wang, Z. Cao, S. Wang et al., "Privacy-enhanced data collection based on deep learning for Internet of vehicles," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6663–6672, 2020.

[32] Q. Liu, Y. Li, H. Duan, Y. Liu, and Z. G. Qin, "Knowledge graph construction techniques," *Journal of Computer Research and Development*, vol. 53, no. 3, pp. 582–600, 2016.

[33] Z. L. Xu, Y. P. Sheng, L. R. He, and Y. F. Wang, "Review on knowledge graph techniques," *Journal of University of Electronic Science and Technology of China*, vol. 45, no. 4, pp. 589–606, 2016.

[34] F. H. Hu, *Chinese Knowledge Graph Construction Method Based on Multiple Data Sources*, East China University of Science and Technology, Shanghai, China, 2014.

[35] C. Y. Li, Y. Z. Wu, and F. H. Hu, "Establishment of packaging knowledge graph based on multiple data sources," *Revista de la Facultad de Ingeniería*, vol. 32, no. 14, pp. 231–236, 2017.

[36] L. Chang, W. T. Zhang, T. L. Gu, W. P. Sun, and C. Z. Bin, "Review of recommendation systems based on knowledge graph," *CAAI Transactions on Intelligent Systems*, vol. 14, no. 2, pp. 207–216, 2019.

[37] Y. Z. Wu, Z. H. Wang, S. H. Chen, G. J. Wang, and C. Y. Li, "Automatically semantic annotation of network document based on domain knowledge graph," in *Proceedings of the 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications*, pp. 715–721, Guangzhou, China, 2017.

[38] W. Jiang, G. Wang, M. Z. A. Bhuiyan, and J. Wu, "Understanding graph-based trust evaluation in online social networks," *ACM Computing Surveys*, vol. 49, no. 1, pp. 1–35, 2016.

[39] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020.

[40] W. Z. Khana, E. Ahmedb, S. Hakakb, I. Yaqoobc, and A. Ahmed, "Edge computing: a survey," *Future Generation Computer Systems*, vol. 97, no. 2, pp. 1–40, 2019.

[41] X. L. Xu, X. Zhang, X. H. Liu, J. L. Jiang, L. Y. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned Internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 10 pages, 2020.

[42] X. L. Xu, Q. Wu, L. Y. Qi, W. C. Dou, S. Tsai, and M. Z. A. Bhuiyan, "Trust-Aware service offloading for video surveillance in edge computing enabled Internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 10 pages, 2020.

[43] X. L. Xu, B. W. Shen, X. C. Yin et al., "Edge server quantification and placement for offloading social media services in industrial cognitive IoV," *IEEE Transactions on Industrial Informatics*, vol. 99, 9 pages, 2020.

[44] T. Wang, L. Qiu, A. K. Sangaiah, A. Liu, M. Z. A. Bhuiyan, and Y. Ma, "Edge-Computing-based trustworthy data collection model in the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4218–4227, 2020.

[45] T. Wang, H. Luo, X. X. Zeng, Z. Y. Yu, A. F. Liu, and A. K. Sangaiah, "Mobility based trust evaluation for heterogeneous electric vehicles network in smart cities," *IEEE Transactions on Intelligent Transportation Systems*, 10 pages, 2020.

[46] J. L. Jiang, Z. Li, Y. Tian, and N. Al-Nabhan, *A Review of Techniques and Methods for IoT Applications in Collaborative Cloud-Fog Environment. Security and Communication Networks*, 2020.

[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[48] S. Ryohei, K. Tetsuji, and Y. Hiroshi, "User behaviour modelling by abstracting low-level window transition logs," *International Journal of Computational Science and Engineering*, vol. 11, no. 3, pp. 249–258, 2015.

[49] S. B. Roy, S. Amer-Yahia, A. Chawla, G. Das, and C. Yu, "Space efficiency in group recommendation," *The VLDB Journal*, vol. 19, no. 6, pp. 877–900, 2010.

[50] Y. Wu, R. Chen, C. Li, and S. Chen, "A personalized collaborative filtering recommendation system of network document resource based on knowledge graph," *Communications in Computer and Information Science*, vol. 1137, pp. 94–106, 2019.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8861639, 14 pages
https://doi.org/10.1155/2020/8861639

Research Article

cHybriDroid: A Machine Learning-Based Hybrid Technique for Securing the Edge Computing

Afifa Maryam,[1] Usman Ahmed,[2] Muhammad Aleem,[3] Jerry Chun-Wei Lin,[2] Muhammad Arshad Islam,[3] and Muhammad Azhar Iqbal[4]

[1]*Department of Computer Science, Capital University of Science and Technology, Islamabad 44000, Pakistan*
[2]*Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen 5063, Norway*
[3]*National University of Computer and Emerging Sciences, Islamabad 44000, Pakistan*
[4]*School of Information Science and Technology (SIST), Southwest Jiaotong University, Chengdu 611756, China*

Correspondence should be addressed to Jerry Chun-Wei Lin; jerrylin@ieee.org

Received 13 August 2020; Revised 7 October 2020; Accepted 3 November 2020; Published 27 November 2020

Academic Editor: Gautam Srivastava

Smart phones are an integral component of the mobile edge computing (MEC) framework. Securing the data stored on mobile devices is very crucial for ensuring the smooth operations of cloud services. A growing number of malicious Android applications demand an in-depth investigation to dissect their malicious intent to design effective malware detection techniques. The contemporary state-of-the-art model suggests that hybrid features based on machine learning (ML) techniques could play a significant role in android malware detection. The selection of application's features plays a very crucial role to capture the appropriate behavioural patterns of malware instances for a useful classification of mobile applications. In this study, we propose a novel hybrid approach to detect android malware, wherein static features in conjunction with dynamic features of smart phone applications are employed. We collect these hybrid features using permissions, intents, and run-time features (such as information leakage, cryptography's exploitation, and network manipulations) to analyse the effectiveness of the employed techniques for malware detection. We conduct experiments using over 5,000 real-world applications. The outcomes of the study reveal that the proposed set of features has successfully detected malware threats with 97% F-measure results.

1. Introduction

Internet of things (IoT), along with edge computing, has revolutionized industrial processes with the help of mobile devices such as tablets, smartphones, smartwatches, and PDAs. Nowadays, mobile devices can adequately render advanced functionalities for efficient, reliable, and scalable cloud services that exploit mobile edge computing (MEC). Extensive usage of Android mobile devices attracts the number of malwares to do MEC services. An increasing number of security threats have emerged recently that is used to steal private user information, lead towards bank frauds, and other socioeconomic crimes [1]. To evade the damages caused by such threats, different malware detection systems [2–4] were presented. Android security solutions for vulnerability assessment and malware analysis can be divided into two main categories as: (1) static and (2) dynamic analysis approaches. In the static technique, the application code is analysed without executing it. The dynamic technique focuses on analysing applications during execution and monitors its interaction with the other system modules and networks [5–7]. However, majority of the existing malware analysis techniques do not consider both the permissions and intents to analyse Android malware.

In contrast to static analysis, most of the dynamic techniques [8, 9] only focus on analysing system and API calls. Existing dynamic malware analysis techniques do not focus on important dynamic features, such as data leakages, network connection manipulation, and enforcing special permissions. Using multiple dynamic features could strengthen the run-time analysis to detect a variety of malicious activities and application security threats. A comprehensive dynamic approach can detect most of the vulnerabilities and security threats at the cost of execution overhead. To efficiently cope with these issues, there should be a comprehensive malware analysis approach that exploits the lightweight static analysis for the already known malware and a comprehensive dynamic approach for the analysis of zero-day malware threats. In this work, we propose a comprehensive framework that incorporates both the static and dynamic analysis exploiting permissions and intents and considers important dynamic features such as data leakages, network connection manipulation, and enforcing special permissions. The major contributions of this research include the following:

(1) a novel machine learning-based framework to analyse Android applications using a hierarchical approach (applying both the static and dynamic analysis) to detect known and zero-day malware,

(2) a machine learning-based comprehensive static analysis model that incorporates both the application's permissions and intents,

(3) a dynamic analysis model that involves the investigation of system calls (such as network activity, files access, SMS activity, and call activity), external DexClass usage, cryptographic activity, run-time permissions enforcement, and rehashing to detect known and zero-day malware,

(4) hyper-tuning malware classifiers using the tree-based pipeline optimization technique to improve the accuracy for malware detection.

2. Literature Review

This section encompasses the critical analysis of existing state-of-the-art approaches related to malware analysis as shown in Table 1.

2.1. Malware Detection Using Static Analysis. Arora et al. [22] suggested a static approach to analyse permissions using the manifest file. A lightweight technique for malware detection was proposed, and its effectiveness was experimentally demonstrated using real Android malware samples. It extracted the permissions from the manifest file and compared them with a predefined keyword list. The designed model considered only one aspect of vulnerability but ignored other aspects, for example, intents and API calls, among others. Another study [10] considered intents (both the explicit and implicit) as semantically rich features to encode the malicious intentions of malware, especially when the intents are used in combination with permissions. The proposed system performed encoding and extracted explicit and implicit intents, intent filters, and permissions. Almin and Chatterjee [5] utilized the k-means clustering algorithm to classify applications which exploited the permission authorization to do malicious activity. The comparison of the research with famous antivirus solutions indicated that the proposed technique was able to detect the malware that remains undetected by most of the antivirus software. MalDozer [18] is a system relying on artificial neural network that took an input of the raw sequences of API method calls with the same order as they showed up in the .dex file for android malware detection and their family recognition. During the training, MalDozer can automatically recognize malicious patterns using only the sequences of raw method calls in the assembly code. A framework [17] using several features that reflects multidimensional characteristics of the Android applications useful for malware detection is proposed. The authors choose a multimodal deep neural network to select the features with different characteristics. They focused on static features such as Opcode, API, permissions, component, and environmental and string features. Experiments were conducted using the data set from VirusShare and Malgenome project. The proposed system attained a good accuracy of up to 98%. Though they studied many static features, the authors use dynamic features useful to detect zero-day and obfuscated malware. Wang et al. [16] recommended a deep learning-based hybrid model using autoencoder (i.e., DAE) and convolutional neural network (CNN) to improve the accuracy of malware detection. Reconstruction of the multiple features of android application performed and multiple CNN were employed for effective malware detection. To boost feature extraction proficiency, several pretraining procedures were accomplished, and customized combination of the deep autoencoder and CNN model (i.e., DAE-CNN) was employed that various learned ranges of patterns in a short time. The empirical test was performed on a data set comprises 23,000 Android applications with the attained 99.8% accuracy.

2.2. Malware Detection Using Dynamic Analysis. The dynamic analysis technique is based on observing the application behavior during execution. In 2012, Google introduced a dynamic analysis-based security infrastructure named Bouncer by Wang et al. [16] for the Android platform. According to Google officials [16], every application that is uploaded on the Google play store is first simulated on Google Cloud infrastructure (using software named Bouncer). The Bouncer aims at guarding the Google Play store against malware threats.

Canfora et al. [8] introduced a detection method to identify malware attacks by employing system calls. Authors assumed that malicious behaviors were implemented by a sequence of system calls. The study employed a machine learning classifier SVM [23] to identify the specific sequence of system calls associated with malware. Authors used the sequence to identify the new malware families. Though the results of this research work produced a promising accuracy of up to 97%, more features like API calls and network statistics should be explored for a comprehensive dynamic analysis and a higher detection rate. A technique, named IntelliDroid is introduced by Wong and Lie [4], to capture the malicious activities during run time of an application. The IntelliDroid recorded instances of specific API calls. The inputs generated by the proposed system triggered different events to monitor application behavior. In [11], the authors suggested an API sequence analysis-based dynamic

Table 1: A summary of related work.

References	Methodology				Used feature		
	Static	Dynamic	Hybrid	ML-based	Static	Dynamic	Data set
Feizollah et al. [10]	✗	✗	✗	✗	Permission	✓	Custom | Drebin
Almin et al. [5]	✓	✗	✗	✗	Intents	✓	Custom
Canfora et al. [8]	✗	✓	✗	✓	✗	System calls	Custom | Drebin
Wong et al. [4]	✗	✓	✗	✗	✗	Malware tracking through input Genera-	Custom | Drebin
Youngjoon et al. [11]	✗	✓	✗	✗	✗	API calls	Custom
Alzaylaee et al. [2]	✗	✓	✗	✗	✗	API calls	Malgenome data set
Zhao et al. [12]	✗	✓	✓	✓	Permissions	General dynamic activities trigged by	Custom
Dash et al. [13]	✓	✓	✗	✓	✗	System calls, Decoded binder communication, abstracted behavioural patterns	Custom
Xu et al. [14]	✗	✓	✓	✓	Collect attack tree path	Graph kernels	Custom
Yuan et al. [15]	✓	✓	✓	✓	Permissions, sensitive API	DexClass, receive net service start	Custom
Wang et al. [16]	✓	✗	✗	✓	Permissions, API calls, hardware features, code patterns	✗	Custom
Kim et al. [17]	✓	✗	✗	✓	Opcode, API, permissions, component, and environmental and string features	✗	Custom | malgenome data set
Karbab et al. [18]	✓	✓	✗	✓	API method calls	✗	Drebin | malgenome | virushare | contagio minidump
Arshad S et al. [19]	✓	✓	✓	✓	Hardware components requested per missions, application components, and API calls.	System calls	Drebin
Hou et al. [20]	✗	✓	✗	✓	Linux kernel system calls	✗	Custom
Pektas and Acarman [21]	✓	✓	✗	✓	Permissions and hidden payload	API calls, installed services, network connections	Virushare

mechanism. To monitor a new program, the hooking process (part of the implemented tool) monitors and tracks the API call sequences of programs. After extracting the API call sequences, the proposed system is compared with the API call sequence reference database. If matched, an alert about the potential malware is generated.

Alzaylaee et al. [2] proposed a system named DynaLog to extract many features (such as logging of high-level behavior and API calls). The extracted features were further analysed to detect malicious applications. The DynaLog took advantage of existing open-source tools such as Droidbox [24] that can detect a wide range of Android malware. The DynaLog is basically based on the Monkey tool [25] provided by Google for testing Android applications. The applications which were unable to run in the emulated environment remain unchecked by the proposed system. Moreover, the DynaLog was incapable of recording events from the native code within Android applications.

2.3. Hybrid Malware Analysis Techniques. A hybrid malware analysis technique combines the features from both the static and dynamic approaches to detect the wide range of Android security threats. Zhao et al. [12] proposed a hybrid malware analysis technique named AMDetector that employs a modified attack tree model [26] for malware analysis. The static part of the proposed technique detects possible attacks and employs this knowledge to classify applications into benign and malware classes. The application behavior triggered by different code components during run time is the part of the proposed dynamic analysis. The organized rules (with attack trees) rendered the good code coverage to the prototype model. The major drawback of the proposed system was the manual formation of rules and time-costly dynamic analysis.

Bläsing et al. [27] suggested the Application Sandbox (Sandbox) system, which is capable to identify malicious applications using the hybrid analysis. The static part of the proposed analyzer extracted the classes (i.e., .dex files) and decompiled these files into human-readable format. Furthermore, the code is scanned for suspicious patterns. The proposed system recorded the low-level details of system interactions during the application execution within the sandbox environment. The sandbox environment ensured the security of analysing system and safety of data of the underobservation device. The dynamic part of the proposed technique employs the Monkey tool [25] to observe the behavior of an application by producing random events. One of the limitations observed within the system was its incapability to detect unknown or new types of malware.

SAMADroid [19] represents a hybrid malware detection model that combined the benefits of three different levels: (1) static and dynamic analysis; (2) host, which is local and remote, and (3) machine learning. Static analysis was performed on remote host considering the features belong to hardware components, requested permissions, application components, and API calls. The dynamic analysis was performed on local host using system calls that helped in the detection of malware patterns. Experimental results show that SAMADroid achieves up to 98% malware detection accuracy. Thus, the inspection of the applications is statistical. However, the employed dynamic analysis is only for the system calls related analysis. The employed dynamic analysis of system calls is already a well-worked area [10], and many malwares easily bypass system calls inspections [28] using code obfuscation techniques. Therefore, there is a need to check the other dynamic features like network activity, API calls, and executable codes.

A technique to discover all the flow paths of most engaging APIs in a program using static analysis was proposed [29]. They preferred static analysis because dynamic analysis is sometimes unable to extract all the important APIs completely. This technique is then named DroidDomTree. The strategy that they opted dependent on the study of dominant API is called during static analysis of an application. These dominant API calls are also known as (semantic signatures), and mining the dominance tree of these semantic signatures is used to detect malware. Furthermore, in the dominance tree, authors assigned weights to individual nodes for effective feature selection. This weighting arrangement supported to choose imperative modules that helped further in feature selection and malware detection. The DroidDomTree detection rate ranged between 98.1% and 99.3%. This study proposed the DL-Droid, a dynamic analysis-based Android malware detection scheme by using deep learning to find malicious patterns in a specific application. Authors enhanced their techniques through a state-based input generation method for improved code coverage. DL-Droid examined the accomplishment of the stateful input generation method using random input generation as a relational baseline. They obtained higher accuracies with these stateful approaches. This study highlighted the significance of enhanced input generation for Android malware detection systems during dynamic analysis. The authors conducted experiments using real devices and achieved a detection rate of 97.8% with dynamic features [24].

Chaulagain et al. [30] suggested a deep learning-based hybrid classifier for the safety screening of Android-based applications. The proposed approach takes advantages of automated feature engineering and the combines benefits of static and dynamic analysis. This research collects different artifacts during static and dynamic analysis and trains the deep learner to get independent models. These separate models combined to create a hybrid classifier that helped in vetting decision. The suggested vetting system has proved efficient against imbalance data and has achieved 99% accuracy. Pektas and Acarman [21] presented a hybrid feature-based classification system that statically analysed the requested permissions and the hidden payload while dynamic features such as API calls, installed services, and network connections were considered for malware detection. Different well-known machine learning algorithms were applied to evaluate the accuracy level in the classification of different classifiers using a data set of 3,339 samples. Authors attained the testing accuracy of up to 92% on the employed Android applications. Though the proposed static analysis technique exploits the permissions and payload features, it ignored the close relationship of intents with permissions. Most of the time, considering only permissions to identify the malware is not adequate [10].

Table 1 shows the summary of related work about methodology and important features that most of the researchers employed for static or dynamic analysis. As shown in Table 1, most of the researchers have concatenated either on static or dynamic analysis and ignored an important aspect of application vulnerabilities that can be exploited in both static and dynamic analysis. A few researchers considered hybrid analysis. However, most of them ignored the intents and permissions relationship, which is a crucial aspect of Android applications. Moreover, most of the researchers have not exploited important system calls (such as network activity, file access, SMS, and call activity), usage of external DexClass, data leaks, cryptographic activity, run-time permissions, and rehashing activity during the execution of applications. The critical analysis of narrated state-of-the-art approaches has led us to formulate the following research questions

(i) **Q1:** which of the static features (e.g., permissions along with certain intents patterns) play a vital role in Android malware detection?

(ii) **Q2:** which combination of the dynamic features such as system calls (i.e., network activity, file access, SMS, and call activity), usage of external DexClass, data leaks, cryptographic activity, run-time permissions, and detection of rehashing activity is important for Android malware identification?

(iii) **Q3:** how can malware detection rate be improved by employing hybrid analysis and machine learning-based classification?

To address these research questions, we propose a hybrid machine learning-based malware detection framework called *HybriDroid* for Android platform.

3. Proposed Hybrid Malware Analysis

To analyse the impact of hybridization, we propose two machine learning-based hybrid malware analyzers, respectively, named *HybriDroid* and ***cHybriDroid***. The *HybriDroid* framework exploits static as well as dynamic features for malware analysis using a hierarchical mechanism. First, the applications are analysed solely using the static features, and then the dynamic features are employed to examine the suspicious (the applications marked as clean by the static analysis) applications. Moreover, to investigate the impact of combined analysis (using both the static and dynamic features), we propose **cHybriDroid** framework.

3.1. HybriDroid Architecture. This section describes the overall methodology of the proposed Android malware analysis framework, that is, *HybriDroid* (shown in Figure 1). The proposed hybrid approach is comprised of a hierarchical system based on two phases: (1) static and (2) dynamic phases (as depicted in Figure 1). In the static analysis phase, the APK files of applications are first dissembled into XML and Java files. After that, the XML files are examined to extract the application related to permissions and intents.

These features are then supplied to the proposed machine learning-based static analyzer. By employing the provided static features, the machine learning-based analyzer categorizes an application like malware or suspicious. To further examine the suspicious applications, the dynamic analysis phase is initiated. The applications classified as suspicious are then provided to the dynamic analyzer for analysing run-time behaviors.

For dynamic analysis, first of all, each application is executed in the emulated environment (using DroidBox [24] emulation tool) to log the observed dynamic features (such as system calls, usage of external DexClass, data leaks, cryptographic activity, and detection of rehashing activity). The dynamic features are then provided to the machine learning-based dynamic analyzer for the classification purpose. The machine learning-based dynamic analyzer classifies these suspicious applications as benign or malware. The applications classified as malware are added to the malware data set while the applications declared as benign are added to the clean applications data set.

3.2. cHybriDroid Architecture. To investigate the impact of combined analysis (using both the static and dynamic features), we propose a ***cHybriDroid*** framework (as shown in Figure 2). The **cHybriDroid** examines the Android applications using both the static and dynamic features simultaneously (see the architecture of **cHybriDroid** in Figure 2). For each Android application, both the static and dynamic features are extracted and provided to the machine learning-based analyzer for classification (as malware or benign). To extract the static features (i.e., intents and permissions), the application is disassembled into APK and manifest files. Moreover, the application is executed in the virtual environment (interactively by tapping and using sample inputs), and the dynamic features are logged. Afterwards, the static (i.e., intents and permissions) and dynamic (such as data leakage, network usage, and use of DexClass) features are provided for the developed **cHybriDroid** to analyse the application (as depicted in Figure 2).

3.3. Classifier Training for HybriDroid and cHybriDroid. Figure 3(a) depicts the complete training process of the proposed *HybriDroid* malware analyzer. The training data set comprises 50% benign (i.e., clean Android applications) and 50% malware (as mentioned in Table 2). As the *HybriDroid* mechanism is based on the hierarchical model, therefore, both the static and dynamic machine learning analyzers are trained separately. To train the static analyzer, an Android application is disassembled into Java and XML files (sample shown in Figure 4) in order to extract the feature vectors related to permission and intents. The disassembled Java, XML, and manifest files are used to obtain static features such as intents and permissions. These intents and permissions are then compared with each application in the data set. If the application intent or permission matches with the extracted permissions, the value of that intent or permission is set to 1; otherwise, it is set to 0. Similarly, a feature vector based on 407 distinct values is formed. These feature vectors along the application category or label (i.e., malware or benign) are provided to the static machine learning analyzer. Similarly, for the training of the dynamic analyzer (in the *HybriDroid* framework), 50% of the benign and 50% of the malware applications-based training data set was executed in a virtual environment (i.e., DroidBox [24]). A total of 15 distinct dynamic features are collected and provided along with the application category (i.e., malware or benign) to the dynamic analyzer (*HybriDroid*). Additionally, *K-fold cross-validation* method is used along with grid search mechanism that is employed for hyperparameter tuning (as shown in Table 3).

Figure 3(b) shows the training of **cHybriDroid** that employs single machine learning-based analyzer trained using both the static and dynamic features simultaneously. For each Android application, the static and the dynamic features are extracted and supplied along with the application category (i.e., benign or malware) to the **cHybriDroid**'s combined analyzer. The combined analyzer is trained using 432 distinct feature vectors based on the static and dynamic aspects of the application.

4. Experimental Result

The experiments are performed on a personal computer. Detailed specifications of the machine are illustrated in Table 4. To evaluate the proposed frameworks, *HybriDroid* and **cHybriDroid**, we employed five machine learning classifiers, respectively, are *Random Forest* (RF), *K star* (K∗), *Naive Bayes* (NB), *Support Vector Machine* (SVM), and J48 decision tree [12–14, 27, 31]. Moreover, TPOT [28] technique is also used that chooses the right machine learning model and the best hyperparameter for that model.

4.1. Data Set. The benign or clean applications in the data set are collected from the Google play store [16], and a third-party app store called Apkpure [16] is shown in Table 5. For

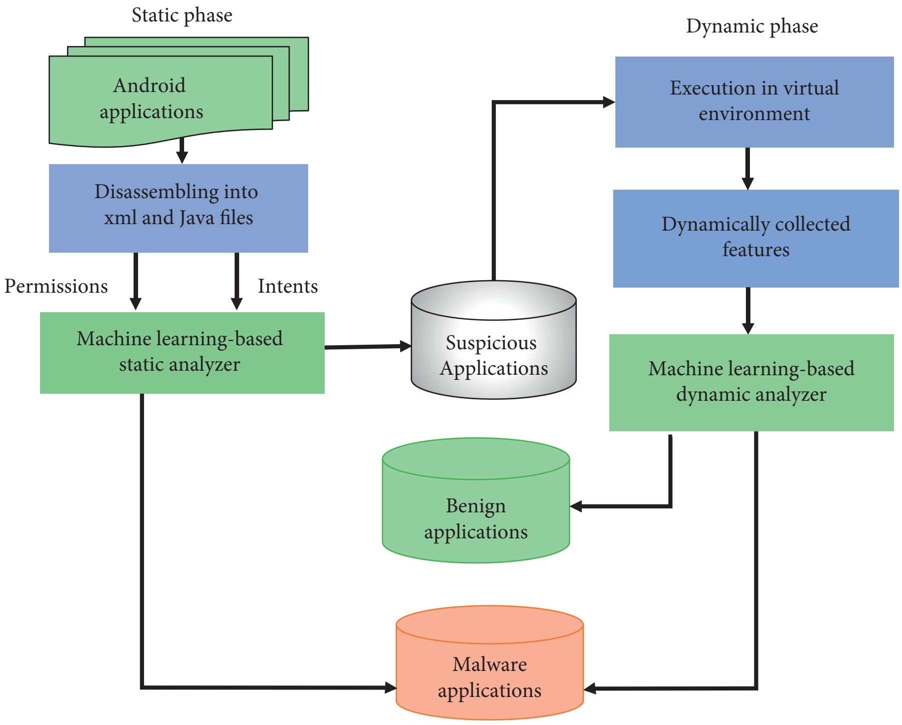

Static phase
Android applications
Disassembling into xml and Java files
Permissions
Intents
Machine learning-based static analyzer
Suspicious Applications
Dynamic phase
Execution in virtual environment
Dynamically collected features
Machine learning-based dynamic analyzer
Benign applications
Malware applications

Figure 1: Architecture of *HybriDroid*.

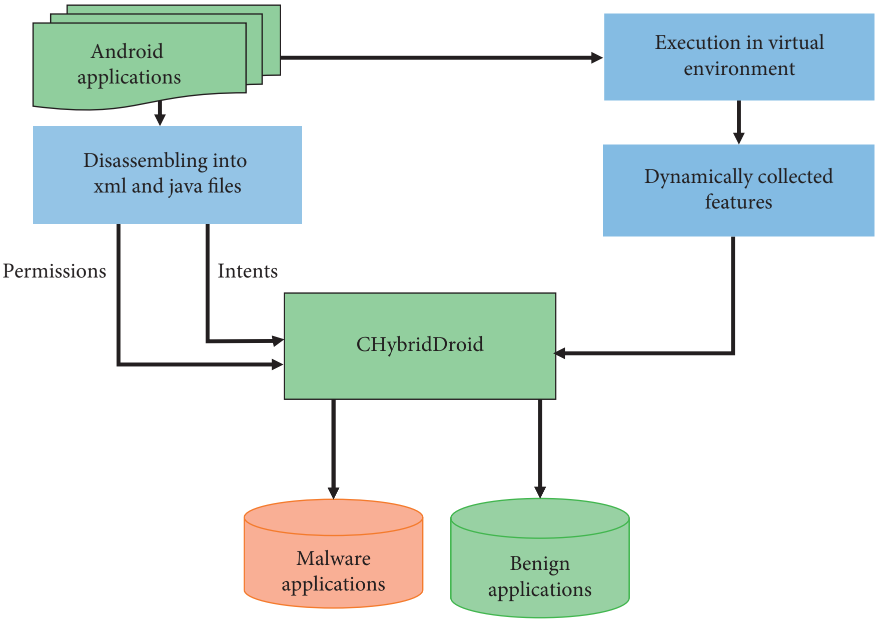

Android applications
Execution in virtual environment
Disassembling into xml and java files
Dynamically collected features
Permissions
Intents
CHybridDroid
Malware applications
Benign applications

Figure 2: Architecture of cHybriDroid.

malware samples, we acquired benchmark Drebin [3] data set that consists of 5,560 malwares from 179 different families and some of them are shown in Table 6. Drebin is extensively used throughout research works on Android malware detection. The Drebin data set consists of malware applications obtained from various Android markets, different antivirus engines, malware forums, security blogs, and Android *malgenome* project [16].

4.2. Feature Selection. The permissions are one of the important static features which must be examined carefully to safeguard from the potential security threats. In addition to the permissions, intents within Android applications are another important aspect requiring careful analysis. Intents are part of the complex messaging model of Android system, which facilitates execution of the different applications, services, and operating system functions. Different activities,

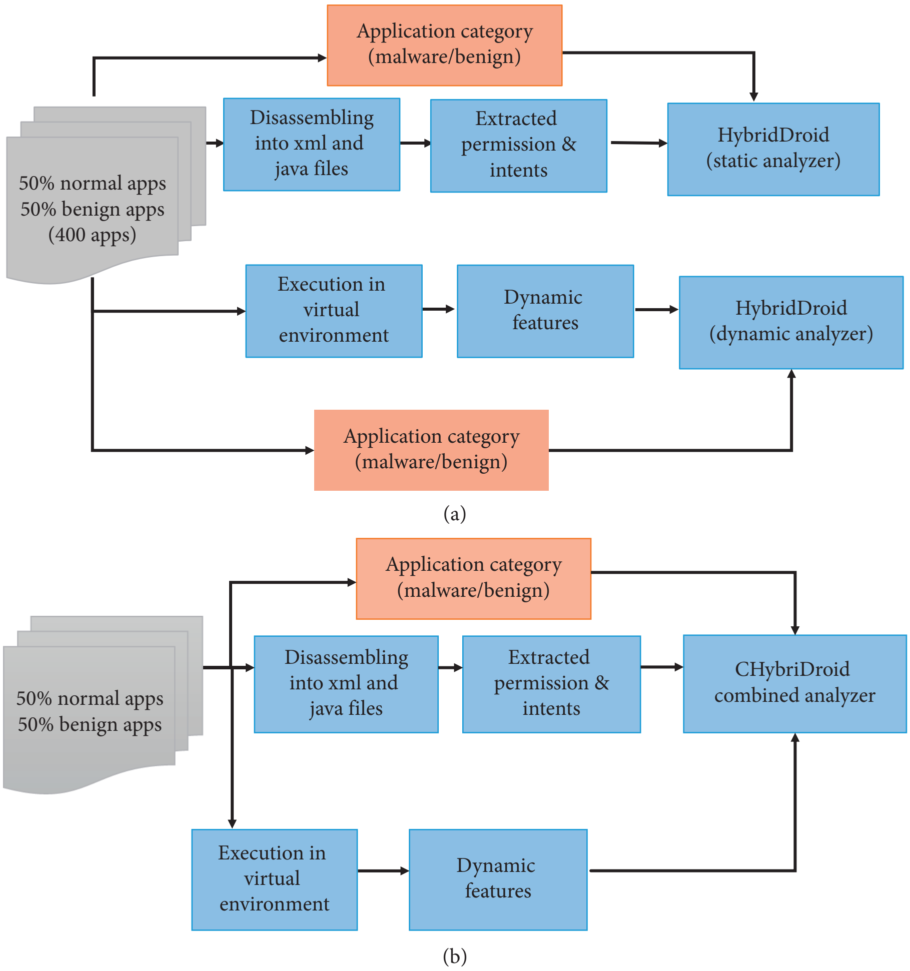


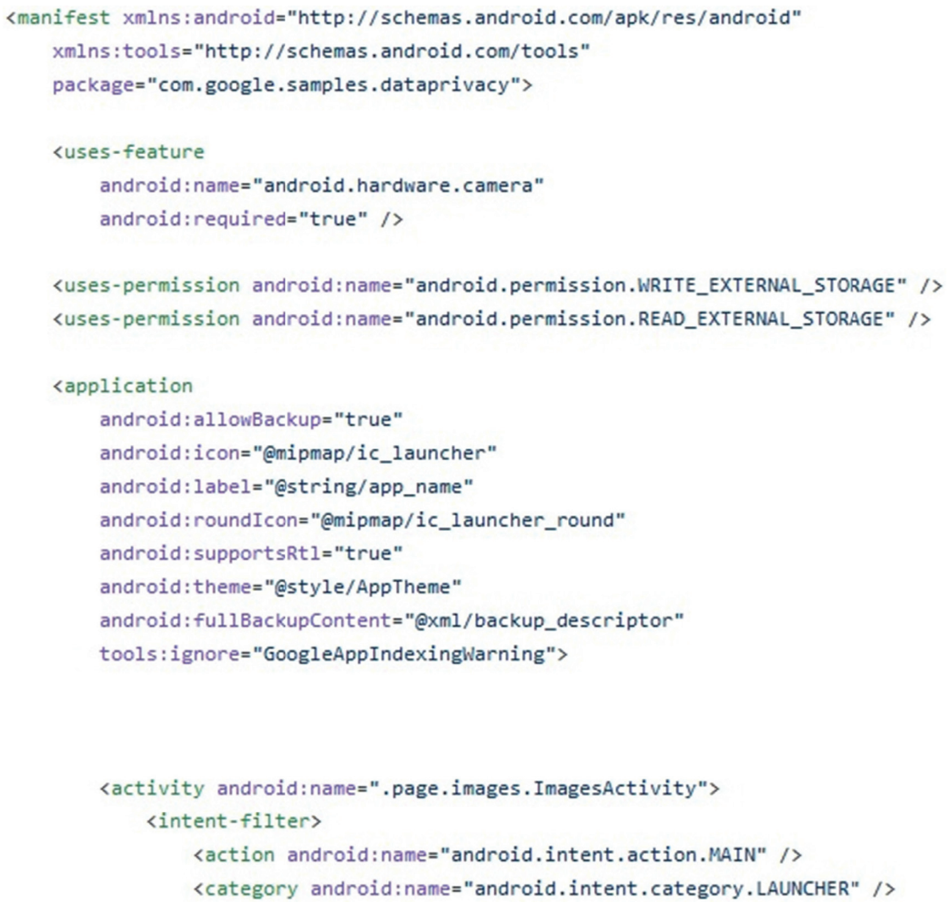
Figure 3: Training methodology. (a) Training of static and dynamic machine learning analyzers for *HybriDroid*. (b) Training of the combined analyzer for cHybriDroid.

Table 2: Data set details.

Application type	Number of applications	Applications categories
Benign	2500	28 different categories
Malware	2500	178 different families

broadcast receivers, and some services used intents for their activation and record their type of intent using intent filters in the manifest file. Some of the recent studies [10, 32] have shown that the intents and permissions are often exploited (such as intent spoofing and permission collusion) by the malware. Thus, their critical examination is necessary to detect malicious activities. Table 7 shows the features collected (using DroidBox tool) during the dynamic analysis step of the proposed methodology. These features are the result of the execution events generated during the execution of applications (within a virtual environment). From Table 8, it is evident that the internet is the most employed (i.e., 20%) permission by the applications (by both the malware and benign). Other permissions that are the part of the most requested permission set in malware applications belong to sending and writing SMS, having a collective percentage of 14. Moreover, accessing approximate and exact locations through *ACCESS_FINE_LOCATION* and *ACCESS_COARSE_LOCATION* permissions is employed by the 11% malware applications.

4.3. Feature Ranking. The motivation behind using a reduced feature set (for the employed predictive models) is to eliminate redundant data, reduce overfitting issues, improve classification accuracy, and decrease the training time of the algorithm. The dynamic analysis results in a large number of features; therefore, it was necessary to use only the important features for the machine learning model. For this purpose, we employ the information gain method [16] that finds certain patterns of the features in the employed applications of the data set. Each feature is assigned with a certain score highlighting the effectiveness of the feature in classification. The *InfoGain* is a well-known feature selection algorithm that records the changes in the entropy of the information class before and after the observation [3]. The formula to measure the information gain is shown as



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.google.samples.dataprivacy">

    <uses-feature
        android:name="android.hardware.camera"
        android:required="true" />

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        android:fullBackupContent="@xml/backup_descriptor"
        tools:ignore="GoogleAppIndexingWarning">



        <activity android:name=".page.images.ImagesActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
```

Figure 4: Sample of the manifest .xml file.

Table 3: Grid search setting.

Classifier	Candidate	Parameters
Decision tree	462	{‘max_features’: [‘auto’-‘sqrt’-‘log2’]-‘min_samples_split’: [2-3-4-5-6-7-8-9-10-11-12-13-14-15]-‘min_samples_leaf’: [1-2-3-4-5-6-7-8-9-10-11]-‘random_state’: [123]}
Random forest	288	{‘Bootstrap’: [True]-‘max_depth’: [80-90-100-110]-‘max_features’: [2-3]-‘min_samples_leaf’: [3-4-5]-‘min_samples_split’: [8-10-12]-‘n_estimators’:[100-200-300–1000]}
SVM	14	{‘C’: [6-7-8-9-10-11-12]- ‘kernel’: [‘linear’-‘rbf’]}
Kstar	192	{‘n_neighbors’: [5-6-7-8-9-10]-‘leaf_size’:[1-2-3-5]-‘weights’:[‘uniform’-‘distance’]-‘algorithm’:[‘auto’-‘ball_tree’-‘kd_tree’-‘brute’]-‘n_jobs’:[−1]}
Naive Bayes	—	—
Tpot	Generative model	—

Table 4: Experimental setup.

Processor	Intel core (TM) i7-4720HQ 2.60 GHz
Memory	16 GB
Operating system	Ubuntu 16.04LTS
Machine learning tool	Weka 3.6

$$\text{infoGain}(P, F) = \text{Entropy}(P) - \sum_{v \in V(F)} \frac{|P_v|}{|P|} \cdot \text{Entropy}(P_v), \tag{1}$$

where P indicates the set representing the pattern, $|P|$ is the number of samples in P, v is the value of the feature F, (P, v) is the value of feature F, and Pv is the subset of P (where feature F has value v). Before the observation of features entropy, the class is defined and shown as

$$\text{Entropy}(P) = \sum_{c \in C} \frac{|P_c|}{|P|} \cdot \log_2 \frac{|P_c|}{|P|}, \tag{2}$$

where C indicates the class set and Pc represents the subset of P belonging to class c. Information gain is considered as a simple and fast ranking method that yields the most suitable features, which are helpful in identifying application class (in our case malware or benign). Using InfoGain, 172 important static features (comprising permissions and intents) out of a total of 407 features are selected. The top 10 features are

Table 5: Nonmalware application details.

S.NO.	Applications categories
1	Health & fitness
2	Art & design
3	Beauty
4	Business
5	Communication
6	Education
7	Event
8	House & home
9	Sports
10	Productivity
11	Photography
12	Camera
13	Finance
14	Auto & vehicles
15	Travel and local
16	Food & drink
17	Lifestyle
18	Video players & editors
19	Weather
20	Social
21	Shopping
22	Tools
23	Parenting
24	News & magazines
25	Music & audio
26	Medical
27	Entertainment
28	Music & audio

Table 6: Malware application details.

S.NO.	Malware family
1	Plankton
2	DroidKungFu
3	GinMaster
4	FakeDoc
5	FakeInstaller
6	Opfake
7	BaseBridge
8	Nisev
9	Adrd
10	$K_{\min}$
11	Geinimi
12	DroidDream
13	FakeRun
14	Iconosys
15	SmsWatcher
16	UpdtKiller
17	Gappusin
18	Proreso
19	Mobsquz
20	Cosha
21	SpyMob
22	Coogos
23	Updtbot
24	Ackposts
25	Fatakr
26	Vidro
27	Booster
28	EWalls

ranked and shown in Table 9. These results show that *Send_SMS* and *Receive_SMS* static features have attained the highest rank value compared with the other static features.

For intents, the *receiver* has been ranked highest among the intent category. The top ranked dynamic features with the rank score are shown in Table 10. As shown in Table 10, *sendsms* is the top dynamic feature that has the highest potential to reveal the category of an Android application (i.e., as malware or benign). *Sendsms* dynamic feature represents information leakage via network, SMS, or any file-based activity. *Cryptousage*, *sendsms*, *enfperm*, and *sendnet* are the other top-ranked features which retain maximum information (i.e., attained higher rank value), and this shows the significance of these features for malware analysis. In this research, we use the top five (out of a total of 15) ranked dynamic features.

The *dataleaks* dynamic feature retains the maximum information when *InfoGain* is applied (as shown in Table 10). This information is necessary for accurate malware classification. Similarly, the *READ_SMS* from the intent category has the highest potential to accurately classify malware compared with the other employed features. The *android.provider.Telephony.SMS_RECEIVED* in the permission's category is among the top 10 highest-ranked permission (as shown in Table 11). In this research, we selected the top 20 (422) hybrid features. The full feature ranking and information gain are mentioned at https://bit.ly/2GduUEt.

4.4. Result Discussion. In Table 12, results related to cross-validation grid search experiment are presented. When feature selection is not performed then, TPOT produces the highest 0.91 F-measure. However, the Naive Bayes produces 0.98 precision, and TPOT produces 0.91 recall. In Table 12, when feature selection is performed, the TPOT F-measure is decreased from 0.91 to 0.87. Random forest produced the best result of 0.88 F-measure and 0.88 precision. The reduced result of the model indicated that removed features have minimal impact on the performance of the classifiers. In Table 12, the cross-validation grid search experiment related data based on dynamic features with and without feature selection is presented. When feature selection is not performed then, TPOT produces the highest 0.94 F-measure, which is 0.03% improved compared with the static features mentioned in Table 13. However, the Naive Bayes produces 0.99 precision and support vector machine produces 0.92 recall. In Table 12, when feature selection is employed, the TPOT F-measure is decreased from 0.94 to 0.91. The TPOT produced the best result of 0.91 F-measure and 0.88 precision while reducing the number of features from 15 to 5 (with a drop of F-measure 0.03). Table 13 shows the best classifier for dynamic features based analysis. In Table 12, cross-validation grid search experiment is conducted on the hybrid features with (20 selected features) and without (total 422 features) feature selection. When feature selection is not performed then the Naive Bayes produces the highest F-measure (i.e., 0.99) which is

Table 7: Applications features for dynamic analysis.

No.	Feature	Description
1	DexClass	Actions of loading external dex function
2	Opennet	Facilitate the connection for network
3	Service start	Log the services in operation
4	Close net	Close network connection
5	Send net	Data transmit/sent to the network
6	Recvnet	Data received from the network
7	Data leaks	Detect leakage of information on the phone including messages, e-mail, password, contacts, IMEI, GPS information phone number, and so on
8	Accessed files	File accesses
9	Fda ccess	Read and write operations of file and directory
10	Send sms	Send SMS
11	Phone call	Phone calls made
12	Cryptousage	Detect the cryptographic functions and what key is used when encrypting and decrypting data
13	Recvaction	APKs function invoked as a receiver
14	Enfperm	Enforce special permission to activity, broadcast receiver, and service
15	Hashes	The hash value of APK file

Table 8: Most used (percentage of occurrence) permissions and intents in the applications.

Frequent permissions	Percentage	Frequently asked intents	Percentage (%)
Internet	20	Action.main	28
Read_phone_state	15	Category.launcher	24
Access_ network_state	13	Boot_completed	14
Write_external_storage	7	Category.default	8
Write_sms	5	Sms_received	8
Send_sms	9	Phone_state	5
Receive_boot_completed	11	Category_home	4
Wake_lock	9	New_outgoing_call	3
Access_fine_location	6	ACTION.VIEW	3
Access_coarse_location	5	Category.browsable	3

Table 9: Static features (ranked using info gain).

Ranked	Importance	Features name
1	0.18147	SEND_SMS
2	0.16383	com.Google.android.c2dm.intent.RECEIVE
3	0.16296	com.Google.android.c2dm.permission.RECEIVE
4	0.14353	com.android.vending.INSTALL_REFERRER
5	0.13254	READ_PHONE_STATE0
6	0.12882	0com.Google.firebase.INSTANCE_ID_EVENT
7	0.12363	READ_EXTERNAL_STORAGE
8	0.12227	ACCESS_NETWORK_STATE
9	0.12148	c1dm.intent.REGISTRATION
10	0.11792	C2D_MESSAGE
11	0.10674	category.BROWSABLE
12	0.10086	android.intent.action.VIEW
13	0.09721	RECEIVE_SMS
14	0.08396	READ_SMS
15	0.07374	GET_ACCOUNTS
16	0.07059	Telephony.SMS_RECEIVED
17	0.06276	GET_TASKS
18	0.06225	com.android.vending.BILLING
19	0.0551	android.intent.action.SEND
20	0.0546	READ_LOGS/writeLogs

0.05% improved compared with dynamic features and 0.08% improved result compared with the static features mentioned in Table 13, respectively. The Naive Bayes produces the precision of 1.00 and the recall of 0.99. In Table 14, when feature selection is performed, the TPOT F-measure results in 0.97 and the Naive Bayes F-measure is decreased from 0.99 to 0.96. The TPOT produced the best results, that is, 0.97 F-measure, 1.00 precision, and

Table 10: Dynamic features (obtained using InfoGain).

Rank	Importance	Feature name
1	0.2654	Sendsms
2	0.2425	Dataleaks
3	0.1872	Cryptousage
4	0.0913	Enfperm
5	0	Accessedfiles
6	0	Servicestart
7	0	Recvnet
8	0	Sendnet
9	0	Phonecalls
10	0	Closenet
11	0	Opennet
12	0	Hashes
13	0	DexClass
14	0	Recvsaction
15	0	Fdaccess

0.94 recall, while reducing the features from 422 to 20, with a drop of F-measure up to 0.02.

Table 13 showed each fold result of the TPOT (without feature selection) and random forest (with feature selection). Since random forest is trained on different samples of the data which reduces variance, it obtained better performance. Moreover, random forest used a random subset of features which also helps to reduce overfitting. The dynamic features based TPOT technique is shown in Table 13 depicting the most performing classifiers (with and without employing feature selection). The reason that the **extratree** classifier obtained the improved results compared with the other classifier is that the random value is selected for feature consideration. The random split for the extra trees helps to create more diversified trees and less *splitters*. Table 13 shows the best classifier using the hybrid features. In the hybrid feature, Naive Bayes classifier resulted in the best classifier without feature selection and the TPOT-based technique results in best classifier with reduced features. The Naive Bayes is a probabilistic based classifier, so it does not require any selection of tune parameter. However, TPOT needed hypertuning, where we used the evolutionary algorithm to optimize the parameter. The tune parameters for TPOT model are *StackingEstimator (estimator = LogisticRegression (C* = 0.1, *dual = True, penalty = "l2")), GaussianNB ())*. The reason that TPOT technique obtained the improved results compared with the other classifiers is that it uses a stack generation technique to improve its performance. The metalearner that outputs Gaussian classifier makes the final prediction. The results presented in Table 13 show that the TPOT and Naive Bayes outperformed the other machine learning models and are more effective in malware detection. The attained F-measure value for the TPOT model indicates the notable performance of the model. It is evident that, for the TPOT model, the true positive rate is observed fairly high and the false positive rate is extremely low. Therefore, we employ the TPOT classification technique for our proposed **cHybriDroid** framework.

4.5. Prediction Model Overhead.

The **cHybriDroid** is trained offline. The overhead of using **cHybriDroid** predictor includes the selective feature extraction and making the predictions. The overhead of feature extraction is negligible (approximated 1s in total) as a feature is extracted at compile time. The prediction model training is performed once, and it is a one-time cost. The training and testing time for both models are mentioned in Table 15. In summary, the overhead of the prediction model is negligible, that is, two seconds for one application.

Table 11: Hybrid features (information gain).

Rank	Importance	Features name
1	0.364	Dataleaks
2	0.312	SEND_SMS
3	0.305	Sendsms
4	0.298	Servicestart
5	0.294	Opennet
6	0.285	READ_SMS
7	0.285	RECEIVE_SMS
8	0.272	ACCESS_COARSE_LOCATION
9	0.272	android.provider.Telephony.SMS_RECEIVED
10	0.259	Sendnet
11	0.259	Recvnet
12	0.252	ACCESS_FINE_LOCATION
13	0.249	Cryptousage
14	0.248	READ_PHONE_STATE0
15	0.23	ACCESS_NETWORK_STATE
16	0.226	WRITE_SMS
17	0.22	Enfperm
18	0.22	CAMERA
19	0.217	action.BOOT_COMPLETED
20	0.214	Internet

Using the hybrid analysis approach, we experimented with real malware and benign Android applications. Our study showed that using both the static and dynamic application features result in a commendable malware detection accuracy. With the feature ranking mechanism, we further optimized the two proposed hybrid methodologies in terms of performance and accuracy. The reduced number and employing only the important features results in good detection performance and accuracy. For hybrid malware analysis, we adopted two strategies: (1) *HybriDroid* and (2) **cHybriDroid**. The *HybriDroid* methodology was typically designed to perform a hybrid malware analysis (employing both the static and dynamic or run-time features) using a hierarchical mechanism. At the same time, the **cHybriDroid** mechanism was employed to analyse the effectiveness of malware detection when the static and dynamic features are analysed simultaneously. Our results exhibit a higher malware detection accuracy for the *HybriDroid* with a 97% F-measure as mentioned in Table 16. We found that the TPOT [28] was the top-performing machine learning model (for **cHybriDroid**) as compared with the other employed models. To attain a better performance insight, we noted the *False Positive Rate* (FPR) and *True Positive Rate* (TPR) for the **cHybriDroid** classifier. The results revealed that the TPOT [28] machine learning model attained the highest performance up to 96% TPR. Similarly, the r^2 value for the TPOT machine learning model also specifies the potential of TPOT to detect malware. Overall, the malware detection accuracy of the hierarchical hybrid approach (i.e.,

Table 12: The summary of results with and without feature selection.

	(a) Static features results			(b) Dynamic features results			(c) Hybrid feature results		
	F-meas.	Prec.	Recall	F-meas.	Prec.	Recall	F-meas.	Prec.	Recall
WO Feat. Sel.									
SVM	0.90	0.90	0.91	0.90	0.90	**0.92**	0.93	1.00	0.89
Decision tree	0.83	0.84	0.84	0.84	0.85	0.84	0.88	0.96	0.83
Random forest	0.89	0.85	0.83	0.89	0.86	0.84	0.96	0.98	0.93
K-star	0.84	0.91	0.8	0.85	0.92	0.80	0.57	1.00	0.42
Naive Bayes	0.85	**0.98**	0.75	0.85	**0.99**	0.75	**0.99**	**1.00**	**0.99**
TPOT	**0.91**	0.92	**0.91**	**0.94**	0.98	0.90	0.99	1.00	0.99
With feat. sel.									
SVM	0.86	0.86	0.88	0.90	0.94	0.87	0.95	1.00	0.91
Decision tree	0.87	0.86	0.89	0.91	0.95	0.87	0.91	0.97	0.87
Random forest	**0.88**	**0.88**	0.89	0.90	0.94	0.87	0.95	0.96	0.95
K-star	0.84	0.77	**0.94**	0.83	0.83	0.87	0.84	1.00	0.74
Naive Bayes	0.79	0.68	**0.94**	0.83	**0.97**	0.73	0.96	1.00	0.93
TPOT	0.87	0.87	0.88	**0.91**	0.94	**0.89**	**0.97**	**1.00**	**0.94**

Table 13: The selected best model in terms of cross-validation score.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$	$k = 9$	$k = 10$	Average
					(a) Static features						
TPOT, without feature selection											
F-measure	0.92	0.82	0.82	0.97	0.93	0.86	0.97	0.92	1.00	0.93	0.91
Precision	1.00	0.75	0.75	1.00	0.90	1.00	1.00	0.95	1.00	0.90	0.93
Recall	0.85	0.90	0.90	0.95	0.95	0.75	0.95	0.90	1.00	0.95	0.91
Random forest, with feature selection											
F-measure	0.92	0.74	0.86	0.93	0.85	0.83	0.97	0.93	0.97	0.84	0.89
Precision	0.95	0.65	0.79	0.90	0.84	0.93	1.00	0.90	0.95	0.89	0.88
Recall	0.90	0.85	0.95	0.95	0.85	0.75	0.95	0.95	0.95	0.80	0.89
					(b) Dynamic features						
TPOT, without feature selection											
F-measure	0.90	0.90	0.97	0.85	0.79	0.97	1.00	1.00	1.00	1.00	0.94
Precision	1.00	0.93	1.00	0.91	1.00	1.00	1.00	1.00	1.00	1.00	0.98
Recall	0.87	0.93	0.73	0.67	0.80	1.00	1.00	1.00	1.00	1.00	0.90
TPOT, with feature selection											
F-measure	0.89	0.90	0.93	0.67	0.84	0.93	0.97	1.00	1.00	1.00	0.91
Precision	1.00	0.93	1.00	0.75	0.81	0.93	1.00	1.00	1.00	1.00	0.94
Recall	0.80	0.87	0.87	0.60	0.87	0.93	0.93	1.00	1.00	1.00	0.89
					(c) Hybrid features						
Native Bayes, with feature selection											
F-measure	0.93	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
Precision	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Recall	0.88	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99
TPOT, with feature selection											
F-measure	0.93	1.00	1.00	0.92	0.92	0.92	1.00	1.00	1.00	1.00	0.97
Precision	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Recall	0.88	1.00	1.00	0.86	0.86	0.86	1.00	1.00	1.00	1.00	0.94

Table 14: Research answers.

Q1	Selected 172 out of 407 based on InfoGain method
Q2	Selected 5 out 15 based on InfoGain method
Q3	Hybrid analysis increased the F-measure score of 5% with and without feature selection

Table 15: Training and testing time.

Model	Training time (seconds)	Testing time (seconds)
TPOT	0.09	0.007

Table 16: Hybrid features comparison.

Classifier	TPR	FNR	R^2
TPOT	0.96	0.04	0.91
SVM	0.91	0.09	0.82
Decision tree	0.87	0.13	0.64
Random forest	0.87	0.13	0.73
Kstar	0.65	0.35	0.27
Naive Bayes	0.96	0.04	0.91

cHybriDroid) was marginally better than the combined hybrid approach, that is, *HybriDroid*.

4.6. Analysis. As seen from Table 13, the static, hybrid, and dynamic model achieve the high F-measure score. We train the analysis tool on a comprehensive data set and use the optimized parameters for machine learning. Within the proposed security mechanism, we firstly do the static analysis part mainly comprising the manifest file, including permission tags and application intents. The reason for the static analysis is that malware can be tested on the submission of the application before the execution of the application. If the model probability is low, the mechanism should apply the dynamic classification model and detect it under control environment. The dynamic method takes rigorous testing, so it will cost execution time. If the model is uncertain again, then the proposed method will apply to the hybrid model. In this way, we test the application with three different models. Table 14 answers the research question mentioned in Section 1. The method can be adopted for the ransomware and adversarial attacks. The method can be applied in a huge size data set. We can train such kind of ensemble machine learning analyzer on discussed features to detect the ransomware application and classify them into families.

5. Conclusion and Future Work

Nowadays, Android is deemed as the renowned OS for mobile devices. Subsequently, the Android platform attracts several malware experts to gather huge economic and social benefits. To mitigate malware activities, different malware detection systems have been proposed. However, the deficiencies in these systems have led us to propose a novel machine learning-based hybrid malware detection framework that employs several important static and dynamic features. Furthermore, the study has also analysed the role of different machine learning classifiers for malware detection. This study highlights that, in the development of a robust machine learning-based malware detection system, the selection of features from the data set is one of the significant steps. Feature selection depends upon the analysis method through which they are extracted. It is the analysis technique that determines the compatibility of features with the classification algorithm. In the experiments, we attain 97% F-measure, and the trained classifier shows a tremendous efficiency with an r^2 value of 0.91. The TPR is also high, that is, 0.96, while the FPR is very low, that is, 0.04. For the future work, we intend to incorporate code coverage, memory utilization, and network statistics aspects of the executing applications (for dynamic analysis). Moreover, the classifiers will be trained to subclassify the malware into families.

Data Availability

The datasets used in the study were taken from previously published studies (Google-play store [16]; Drebin [3]).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The research was partially supported by Western Norway University of Applied Sciences, Norway.

References

[1] P. Calciati, K. Kuznetsov, A. Gorla, and A. Zeller, “Automatically granted permissions in android apps,” in *Proceedings of the International Conference on Mining Software Repositories*, pp. 114–124, Montreal, Canada, May 2020.

[2] M. K. Alzaylaee, S. Y. Yerima, and S. S. Dynalog, “An automated dynamic analysis framework for characterizing android applications,” 2016, http://arxiv.org/abs/1607.08166.

[3] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, and K. Rieck, “DREBIN: effective and explainable detection of android malware in your pocket,” in *Proceedings of the Annual Network and Distributed System Security Symposium*, pp. 1–15, San Diego, CA, USA, March 2014.

[4] M. Y. Wong and D. Lie, “IntelliDroid: a targeted input generator for the dynamic analysis of Android Malware,” in *Proceedings of the Annual Network and Distributed System Security Symposium*, University of Toronto, Toronto, Canada, 2016.

[5] S. B. Almin and M. Chatterjee, “A novel approach to detect android malware,” *Procedia Computer Science*, vol. 45, pp. 407–417, 2015.

[6] I. A. Dogru and M. Önder, “Appperm analyzer: malware detection system based on android permissions and permission groups,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 30, no. 3, pp. 427–450, 2020.

[7] S. Liang and X. Du, “Permission-combination-based scheme for android mobile malware detection,” in *Proceedings of the IEEE International Conference on Communications*, pp. 2301–2306, Sydney, NSW, Australia, June 2014.

[8] G. Canfora, E. Medvet, F. Mercaldo, and C. A. Visaggio, “Detecting android malware using sequences of system calls,” in *Proceedings of the International Workshop on Software Development Lifecycle for Mobile*, pp. 13–20, Bergamo, Italy, August 2015.

[9] S. J. Hussain, U. Ahmed, H. Liaquat, S. Mir, N. Z. Jhanjhi, and M. Humayun, “IMIAD: intelligent malware identification for android platform,” in *Proceedings of the International Conference on Computer and Information Sciences*, pp. 1–6, Sakaka, Saudi Arabia, April 2019.

[10] F. Ali, N. B. Anuar, R. Salleh, G. Suarez-Tangil, and S. Furnell, “AndroDialysis: analysis of android intent effectiveness in

malware detection," *Computers & Security*, vol. 65, pp. 121–134, 2017.

[11] K. Youngjoon, E. Kim, and H. K. Kim, "A novel approach to detect malware based on API call sequence analysis," *International Journal of Distributed Sensor Networks*, vol. 11, no. 9, p. 659101, 2015.

[12] S. Zhao, X. Li, G. Xu, L. Zhang, and Z. Feng, "Attack tree based android malware detection with hybrid analysis," in *Proceedings of the IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 380–387, Beijing, China, September 2014.

[13] S. K. Dash, G. Suarez-Tangil, S. J. Khan et al., "Classifying android malware based on runtime behavior," in *Proceedings of the IEEE Security and Privacy Workshops*, pp. 252–261, San Diego, CA, USA, May 2016.

[14] L. Xu, P. Z. Dong, M. A. Alvarez, J. A. Morales, X. Ma, and J. Cavazos, "Dynamic android malware classification using graph-based representations," in *Proceedings of the IEEE International Conference on Cyber Security and Cloud Computing*, pp. 220–231, New York, NY, USA, October 2016.

[15] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-sec: deep learning in android malware detection," in *Proceedings of the ACM SIGCOMM Conference*, pp. 371-372, Beijing, China, August 2014.

[16] W. Wang, M. Zhao, and J. Wang, "Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, pp. 3035–3043, 2019.

[17] T. Kim, B. Kang, M. Rho, S. Sezer, and E. G. Im, "A multimodal deep learning method for android malware detection using various features," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 773–788, 2019.

[18] E. B. Karbab, M. Debbabi, A. Derhab, and D. M. Maldozer, "Automatic framework for android malware detection using deep learning," *Digital Investigation*, vol. 24, no. S48–S59, 2018.

[19] S. Arshad, M. A. Shah, A. Wahid, A. Mehmood, H. Song, and H. Yu, "Samadroid: a novel 3-level hybrid malware detection model for android operating system," *IEEE Access*, vol. 6, pp. 4321–4339, 2018.

[20] S. Hou, A. Saas, L. Chen, and Y. Ye, "Deep4maldroid: a deep learning framework for android malware detection based on linux kernel system call graphs," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence - Workshops*, Omaha, NE, USA, October 2016.

[21] A. Pektas and T. Acarman, "Ensemble machine learning approach for android malware classification using hybrid features," in *Proceedings of the International Conference on Computer Recognition Systems*, Wroclaw, Poland, May 2017.

[22] A. Arora, S. K. Peddoju, and M. C. PermPair, "Android malware detection using permission pairs," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1968–1982, 2020.

[23] J. Cui, L. Wang, X. Zhao, and H. Zhang, "Towards predictive analysis of android vulnerability using statistical codes and machine learning for iot applications," *Computer Communications*, vol. 155, pp. 125–131, 2020.

[24] M. K. Alzaylaee, S. Y. Yerima, and S. S. Dl-droid, "Deep learning based android malware detection using real devices," *Computers and Security*, vol. 89, 2020.

[25] N. Wongwiwatchai, P. Pongkham, and K. Sripanidkulchai, "Comprehensive detection of vulnerable personal information leaks in android applications," in *Proceedings of the IEEE Conference on Computer Communications Workshops*, Toronto, ON, Canada, July 2020.

[26] S. Valluripally, A. Gulhane, R. Mitra, K. A. Hoque, and C. Prasad, "Attack trees for security and privacy in social virtual reality learning environments," in *Proceedings of the IEEE Annual Consumer Communications & Networking Conference*, Las Vegas, NV, USA, January 2020.

[27] T. Bläsing, L. Batyuk, S. Aubrey-Derrick, S. A. Çamtepe, and A. Sahin, "An android application sandbox system for suspicious software detection," in *Proceedings of the International Conference on Malicious and Unwanted Software*.

[28] R. S. Olson, B. Nathan, R. J. Urbanowicz, and J. H. Moore, "Evaluation of a tree-based pipeline optimization tool for automating data science," in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*, pp. 485–492, Denver, CO, USA, July 2016.

[29] S. Alam, S. A. Alharbi, and S. Yildirim, "Mining nested flow of dominant APIs for detecting android malware," *Computer Networks*, vol. 167, p. 107026, 2020.

[30] D. Chaulagain, P. Poudel, P. Pathak et al., "Hybrid analysis of android apps for security vetting using deep learning," in *Proceedings of the IEEE Conference on Communications and Network Security*, Avignon, France, June 2020.

[31] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, ""Andromaly": a behavioral malware detection framework for android devices," *Journal of Intelligent Information Systems*, vol. 38, no. 1, pp. 161–190, 2012.

[32] F. Idrees and M. Rajarajan, "Investigating the android intents and permissions for malware detection," in *Proceedings of the IEEE International Conference on Wireless and Mobile Computing*, pp. 354–358, Larnaca, Cyprus, October 2014.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8857564, 6 pages
https://doi.org/10.1155/2020/8857564

Research Article

A Prediction Approach for Video Hits in Mobile Edge Computing Environment

Xiulei Liu,[1,2] **Shoulu Hou**,[1,2] **Qiang Tong**,[1,2] **Xuhong Liu**,[1,2] **Zhihui Qin**,[1,2] **and Junyang Yu**[3]

[1]*Beijing Advanced Innovation Center for Materials Genome Engineering, Beijing Information Science and Technology University, Beijing 100101, China*
[2]*Laboratory of Data Science and Information Studies, Beijing Information Science and Technology University, Beijing 100101, China*
[3]*Software School, Henan University, Kaifeng 475001, China*

Correspondence should be addressed to Junyang Yu; jyyu@henu.edu.cn

Received 13 August 2020; Revised 12 September 2020; Accepted 27 October 2020; Published 17 November 2020

Academic Editor: Xiaolong Xu

Smart device users spend most of the fragmentation time in the entertainment applications such as videos and films. The migration and reconstruction of video copies can improve the storage efficiency in distributed mobile edge computing, and the prediction of video hits is the premise for migrating video copies. This paper proposes a new prediction approach for video hits based on the combination of correlation analysis and wavelet neural network (WNN). This is achieved by establishing a video index quantification system and analyzing the correlation between the video to be predicted and already online videos. Then, the similar videos are selected as the influencing factors of video hits. Compared with the autoregressive integrated moving average (ARIMA) and gray prediction, the proposed approach has a higher prediction accuracy and a broader application scope.

1. Introduction

At present, smart device users spend more than 70% of the fragmentation time in the entertainment applications such as videos and films. The video content providers (e.g., Netflix) desire to know the future video view counts of all their videos, especially the new ones, to provide a better experience for consumers. In the era of the 5G-based mobile edge computing, the explosive increase of video resources requires placing multiple copies to the edge of the network for better performance [1–3]. The distributed model brings many problems for video service providers, such as how to keep the storage efficiency and improve the energy efficiency of storage [4, 5]. The key to solve these problems is to effectively manage data copies and data nodes [6–8]. The current popular cloud storage platforms generally use a static storage mechanism, that is, setting the number of copies before placing them, such as Google File System [9], Hadoop Distributed File System (HDFS) [10], and Amazon Dynamo [11]. The static placement of copies is easy to implement, but it may lead to unbalanced access. For example, it is found that 90.26% of the data in Yahoo's Hadoop Cluster can only be accessed within two days after constructing, 89.61% of the data from the last access to deletion do not exceed 10 days, and 40% of the data have a dormant period (not accessed) for more than 20 days [12].

The current research [13] shows that the migration and reconstruction of video copies is an effective means to improve the storage efficiency, and the prediction of the number of video hits is the prerequisite for migrating video copies. There are many data prediction and recommendation approaches [14, 15]; however, they do not consider the fine-grained granularity of every video copy and cannot provide the required information for migration and reconstruction of video copies. Additionally, the videos occupy a high proportion of storage space and have more rich attributes, and the trend of video hits is influenced by various factors, which are difficult to predict accurately. Based on the combination of correlation analysis and wavelet neural network (WNN), this paper proposes a new prediction approach for video hits by analyzing the correlation between the video to be predicted and already online videos, and selecting the similar videos as the influencing factors.

2. Related Work

2.1. Feature Analysis of Video Copies. A video copy covers a variety of attributes, such as news, teaching, and viewing, and it is the most representative type of data copies which are stored on the cloud. The existing hits prediction are coarse-grained (see Figure 1(a)), and they do not consider the fine-grained hits of every copy and cannot provide information for migration and reconstruction of video copies. The prediction approach that provides the hits of video copies is an urgent problem.

Figure 1(a) shows the trend of a long-timescale (in months) from March 2014 to February 2015. The reason for the smooth rising during the first few months is that "Transformers 4" released on June 2014 stimulates the previous series and even related science fiction series. From the short-timescale (in weeks) counterpart in Figure 1(b), it is found that the video hits have a certain periodicity and autocorrelation. This means that the video hits are affected by the hits of other similar videos. Therefore, the following are the general directions to follow to predict video hits: the short learning time of time series data and the small number of available parameters.

2.2. Existing Prediction Approaches. With more requirement of new entertainment, various prediction approaches for guaranteeing quality of service (QoS) have attracted increasing attention in recent years [16, 17]. Zhang et al. [18] proposed a covering-based quality prediction method for web services via neighborhood-aware matrix factorization. Qi et al. [15] proposed a novel privacy-aware data fusion and prediction approach for smart city industrial environment, which is based on the classic locality-sensitive hashing technique. Zhang et al. [19] proposed a distributed edge QoS prediction model with privacy-preserving for the edge computing networks. However, these works are from macroscopic point and cannot apply to the hits prediction of video copies which requires a micro perspective.

The autoregressive integrated moving average (ARIMA) model is based on the autocorrelation of time series, which is characterized by the fact that the first and the time series are broad and stable. Additionally, if the value of the individual data does not fluctuate up and down in the sequence mean value, ARIMA can do smooth processing of the original data through differential way. In this way, even if the data have a certain degree of fluctuation, the prediction accuracy can be achieved through the smooth processing. Therefore, ARIMA is suitable for prediction of the data with the flat trend and feature of linear wide stationary processes. It is often used in network traffic prediction [20, 21], while the traffic of each data copy has a more fine-grained granularity.

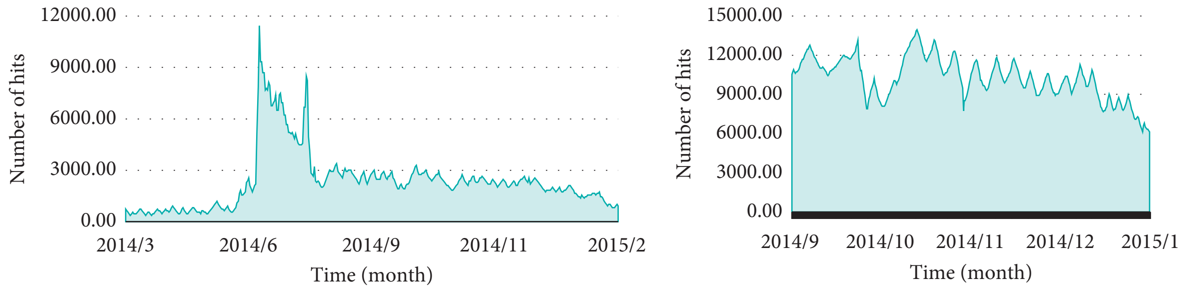
Gray prediction is a knowledge acquisition method with incomplete or uncertain data [22]. It processes the original data through the analysis of the difference degree of the changes in the system factors and predicts the future data by establishing the gray differential prediction model based on a small amount of information. Gray prediction model applies to the situation that the video copy is just launched into the market, which has little original data. These two methods seem to be feasible, but because of the video copy of the fluctuations in many factors, we analyze this video from the factors.

This paper considers the fact that the number of hits for each video is closely related to its similar videos. This is achieved by analyzing the principal components of the sources of the video viewing modes. For example, a systematic quality correlation model was proposed that described three different types of quality correlations between services [23]. Logically, there are some relevant video recommendations after watching a certain video on the web, and the recommendation and impact of the associated videos are the principal components of video-on-demand. By investigating and analyzing the Storm website, after a video is watched, the average viewing rates of the top three similar videos recommended by Storm are 47%, 30%, and 25%, respectively. The average value of the recommendation results comes from 200 videos which are randomly sampled from the Storm website. This means that when the total number of viewers remains stable, it is possible to infer the number of potential viewers of a certain video in the future, according to the current number of viewers of its similar videos. There exists delay during the process, and it is essential for intelligent analysis. This paper applies the wavelet neural network to predict future video hits. The wavelet-based neural network replaces the activation function of hidden nodes with wavelet function. In recent years, several researches applied the wavelet neural network model to predict network traffic [24], but the hits prediction of video copies is a more micro perspective. The key to solve this problem is to inversely deduce the videos that are used as prediction-relevant parameters, according to the video to be predicted.

2.3. Selection of Associated Videos. The choice of related video in this paper is divided into two steps: the first step is to select 12 videos that are similar to the video to be predicted; the second step further reduces the parameter dimension, from the similarity of the video selecting the highest degree of relevance of the four.

2.3.1. Similar Video Selection. Similar video selection is in the same type of videos through the establishment of vector space model (VSM) to calculate and select the level of the video, the degree of the audience, the type of video, the age of the audience, the influence of the producers, etc., and then refine and quantify the score between 0 and 100, as shown in Table 1.

Let m_1 and m_2 be the video objects to be compared, and $f_1, f_2, \ldots\ldots, f_n$ be the attributes of every object. The similarity



(a) (b)

Figure 1: The hits trend of "Transformers 3." (a) The trend of "Transformers 3" from 2014.3 to 2015.2. (b) The trend of "Transformers" from 2014.9 to 2015.2.

Table 1: Video attribute rating scale.

	Production company	Actor level	The audience high school≤	Bachelor degree	≥Master's degree	Ages 20≤	20–30	30–50	≥50	Video rating
m_1	80	90	30	60	10	32	56	6	6	72
m_2	80	90	20	76	14	16	70	9	5	80
m_3	70	80	10	70	20	17	65	12	6	83
...	...	...	...	...	...	...	...	...	...	...

between m_1 and m_2 can be calculated by using the attributes of video objects. The values of the video attribute f_i corresponding to m_1 and m_2 are denoted by a_i and b_i, respectively. By using support vector machine (SVM), the similarity between m_1 and m_2, denoted by $\operatorname{Sim}(m_1, m_2)$, is calculated as follows:

$$\operatorname{Sim}(m_1, m_2) = \frac{m_1 \cdot m_2}{\|m_1\| \times \|m_2\|} = \frac{\sum_i (a_i \times b_i)}{\sqrt{\sum_i a_i^2 \times \sum_i b_i^2}}. \tag{1}$$

From equation (1), we can calculate the most similar 12 videos, in order to further reduce the dimension, and from the 12 videos to select the most relevant video as a prediction parameter.

Table 2: Comparison on the video hits.

$x1$	$x2$	$x3$	$x4$	$x5$	$x6$
48101	26773	34708	849620	228982	5319
49052	27698	46100	824476	231698	4018
49702	27389	60668	822595	253113	4282
52146	26822	61768	743477	256318	4341
......	......	......	......	......	......
29694	86084	58727	236572	104088	3480

2.3.2. Correlation Degree Analysis. Table 2 shows the hits of 6 sample videos within 10 days, where $x1$ is from 2014.10.26 to 2014.11.24, and the data of $x2, \ldots, x5$ cover the time span from 2014.10.20 to 2014.11.18. $x1$ is the video to be predicted, which can be obtained by the hits of other videos in the last week.

The calculation of the association degree can be divided into the following steps.

Step 1: standardize the data in Table 2, where x, y represent longitudinal data and horizontal data, respectively. The sample mean of j-th video is $\overline{Y}_j = (1/m)\sum_{i=1}^{m} Y_{ij}$ $(j = 1, 2, \ldots\ldots, n)$; through it, we can calculate the sample variance of j-th video: $S_j^2 = (1/m-1)\sum_{i=1}^{m}(Y_{ij} - \overline{Y}_j)^2$ $(j = 1, 2, \ldots\ldots, n)$; standardized form is $X_{ij} = (Y_{ij} - \overline{Y})/S_j$. After standardized calculation, we have

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2n} \\ \vdots & \vdots & & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{mn} \end{bmatrix}. \tag{2}$$

Step 2: calculate the correlation matrix$R = (r_{ij})_{m\times m}$, $r_{ij} = \sum_{k=1}^{n} x_{ki}x_{kj}/(n-1)$, $(i, j = 1, 2\ldots, m)$; then we can analyze the correlation between the videos after getting the correlation matrix, as shown in Table 3, in which the x1 is the video that needs to be predicted; $x2$, $x3$, $x4$, $x5$, $x6$ is the similar video that needs to be analyzed for the degree of change.

From 12 similar videos, select the correlation matrix and $x1$ closest to the four videos used as predictive video; the above matrix is the first batch to be compared with video $x1$; it is clear that $x4$, $x5$, and $x1$ are highly correlated with clicking playback. In this way, select the videos $x3$–$x5$ and use the current number of hits to predict the hits of video $x1$.

Table 3: Correlation matrix of videos.

	$x1$	$x2$	$x3$	$x4$	$x5$	$x6$
$x1$	1.0000	−0.4893	0.3744	0.8808	0.9149	0.3654
$x2$	−0.4893	1.0000	−0.0046	−0.3993	−0.4509	−0.1175
$x3$	0.3744	−0.0046	1.0000	0.0990	0.3364	0.3366
$x4$	0.8808	−0.3993	0.0990	1.0000	0.9322	0.0689
$x5$	0.9149	−0.4509	0.3364	0.9322	1.0000	0.0785
$x6$	0.3654	−0.1175	0.3366	0.0689	0.0785	1.0000

2.4. Wavelet Neural Network. Neural networks are often used to predict and analyze non-linear time series. In theory, neural network prediction accuracy can be achieved arbitrarily. But in practical application, it may meet many difficulties, such as the delineation of the structure of the neural network, training and learning process being too slow, and being trapped in local second-best in the optimization process. WNN combines the characteristics of wavelet analysis and neural network. It replaces the neurons in the neural network with wavelet neurons, replaces the Sigmoid in the neural network with wavelet functions (as activation function), and establishes the relationship between wavelet transform and network coefficients through affine transformation. The WNN has the following characteristics:

(1) High prediction accuracy due to strong learning ability and excellent function approximation ability.

(2) The fact that local optimum can be avoided as the shift factor and scaling factor of the network are determined in advance.

(3) High learning and training speed as wavelet function as an activation function is easy to implement.

The WNN has three layers: input layer, hidden layer, and output layer, as shown in Figure 2. The output layer adopts the linear output. The neurons of the input layer are (x_1, x_2, ..., x_m), the hidden layer has K neurons, and the output layer has N neurons. In this case, since the number of associated videos is 4, there are 4 related parameters as the basis for prediction and the input neuron selection is 4. In the set of daily visits, it is found that the visits are recorded in the unit of day; the visit cycle is 7 days. Then, the number of neurons used in the middle of the hidden layer is set to 7, and the output layer is 7. That is why this case uses 4-7-7 structure neural network. $h_k(x)$ is the wavelet basis function, instead of the previous activation function Sigmoid.

Let w_{ij} denote the weights of neurons from layer $p-1$ to layer p, a_r^m denote the r-th input of neuron i in layer p, φ_p denote the transfer function of layer p, and b_r^p denote the corresponding output of the layer.

$$a_r^p = \sum_i w_{ij}^p b_r^{(p-1)}, \tag{3}$$

$$b_r^p = \varphi_p\left(a_r^m\right). \tag{4}$$

Since the case is designed in a three-tier network, $b_r^1 = x_{r-j+1}$, $b_r^3 = \widehat{x}_{r+1}$ are the transfer functions in the hidden layer. This case is called "Morlet wavelet," which is given by

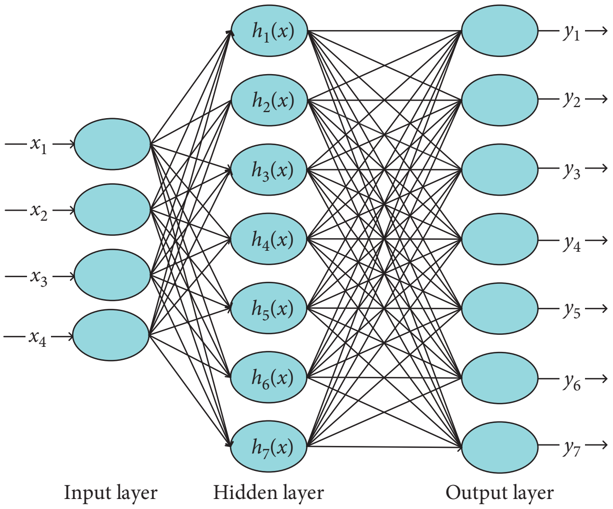


Figure 2: The hierarchy diagram of WNN.

$$\varphi(\mu) = \cos(1.75\mu)e^{\left(-\mu^2/2\right)}. \tag{5}$$

Substituting (5) into (3) and (4), we get

$$a_r^2 = \sum_{i=1}^{m} w_{ij}^2 x_{r=j+1}, \tag{6}$$

$$b_r^2 = \varphi_2\left(\frac{a_r^2 - b_j}{a_j}\right). \tag{7}$$

According to (6) and (7), we can conclude that the video prediction on time series is

$$x_{r+1} = \sum_{i=1}^{K} b_r^2 w_j^3 = \sum_{j=1}^{K} \varphi_2\left(\frac{a_r^2 - b_j}{a_j}\right). \tag{8}$$

Given the input and output samples of group, the error function can be expressed as

$$E = \sum_{P=1}^{P} E^P = \frac{1}{2P}\sum_{P=1}^{P}\sum_{n=1}^{N}\left(d_n^P - y_n^P\right), \tag{9}$$

where d_n^P is the expected output of the n-th node and y_n^P is the actual output of the n-th node. w_k, b_k, and a_k are constantly adjusted to minimize the error. When the error is less than the given value, the program will end, and an appropriate prediction value can be calculated.

For a series of videos that have similar and stable attributes, the proposed approach can achieve reasonable prediction results. However, for certain videos that have a worse or better reputation than other videos in the series, the proposed approach may produce a biased prediction result. This is because a potentially unbalanced reputation has affected the final results.

2.5. Performance Evaluation. As the actual video copy, if the prediction granularity is too low, such as one day, it does not make much sense, because it is impossible for the video copy

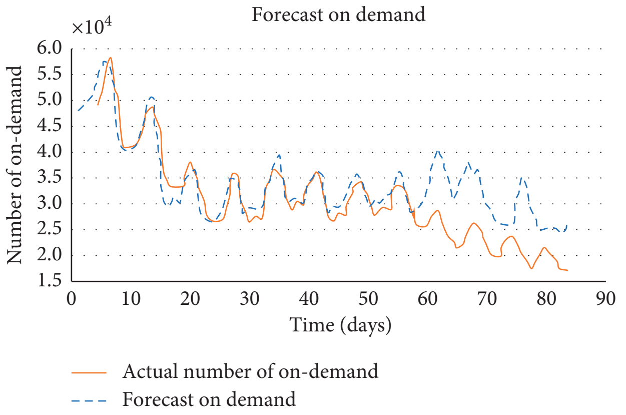


Figure 3: Prediction of ARIMA approach (56–84 is the actual prediction).

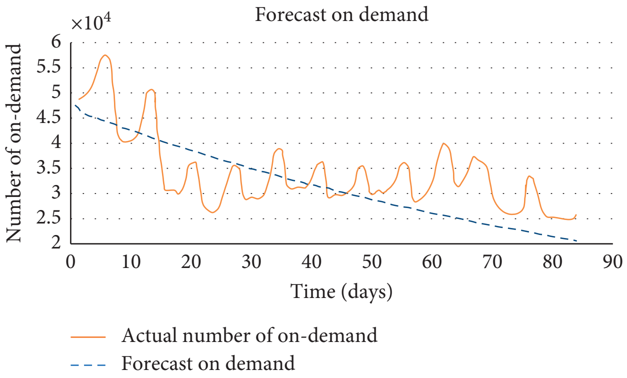


Figure 4: Prediction of gray approach (56–84 is the actual prediction).

to undergo frequent evaluation migration with a 24-hour time unit. If the prediction granularity is too long, such as in the unit of month, the weekly trends of a video copy cannot be timely captured. So, this experiment is expected to have a length of weeks and data collection is based on days, the estimated collection of 84 days of data. The number of targets to be tested is from 2014.10.26 to 2015.1.17, and the video data as the influencing factor are from 2014.10.20 to 2015.1.11, the first 56 days of the two sides are used as training data, and the last 28 days of data are used as the test data. The experiment runs on 64-bit Windows 7 Professional with 10 GB of RAM and 2.1 GHz Intel Core i3 processor and uses MATLAB R2014a to conduct the simulation and numerical analyses. The neural network is designed to be 4-7-7 structure, and the output is 7-node data.

From Figure 3, it can be seen that ARIMA through differential algorithm fits the time series data well before prediction, but it is not accurate in grasping the trend of the data. Figure 4 shows that the gray prediction can predict the general trend of the data but has a large error range. In terms of the change trend and prediction accuracy, the performance of the proposed approach exhibits the optimal results, as shown in Figure 5.

If it is predicted after a week, it will become a training set and then forecast the way of the next week. The time granularity of each prediction is not the same, which is 7 days, 14 days, and 28 days, respectively. As can be seen from Table 4, if the time granularity for prediction is 1 week or 4 weeks, the proposed approach is the most accurate and far higher than those of ARIMA and gray. If the time granularity for prediction is two weeks, the accuracies of the three prediction methods are comparable. In the proposed approach, the histories of hits of the similar video are used as the influencing factors. Therefore, it can be inferred that the prediction approach based on the combination of the correlation analysis and WNN has better practical effect than the alternatives.

When applying this method to several other types of videos, the result is similar to the above case. The average prediction accuracy of the proposed approach is 10% higher than that of ARIMA and 5–7% higher than that of gray prediction.

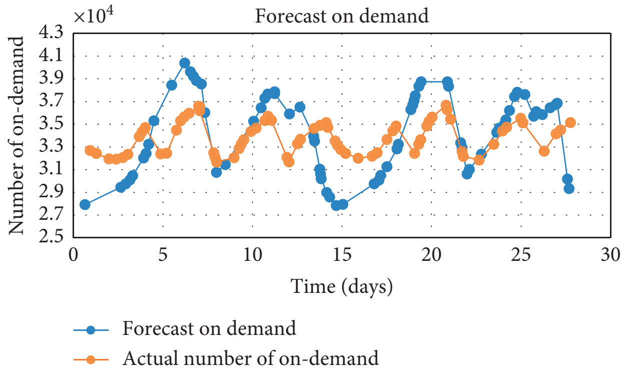


Figure 5: Prediction of the proposed approach.

Table 4: Comparison of the error rates of different approaches.

	Forecast for 7 days (%)	14 days (%)	28 days (%)
ARIMA	16.7	18.01	24.81
Gray	18.28	16.11	20.60
WNN	13.85	16.49	14.11

From the above analysis, each prediction approach has its own advantages and disadvantages. ARIMA has certain advantages if the video enters a certain life cycle, which have the relatively flat change trend and the feature of linear wide stationary processes. But if the change trend of hits is gentle, it has little significance for migrating video copies.

3. Conclusions

This paper proposes a new prediction approach for video hits based on the combination of correlation analysis and WNN. This is achieved by establishing the video index quantification system and analyzing the correlation between the video to be predicted and already online videos. Then, the similar videos are selected as the influencing factors of video hits. Compared with the ARIMA and gray prediction, the proposed approach has a higher prediction accuracy and a broader application scope.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Xiulei Liu and Junyang Yu proposed the overall idea of this manuscript. Shoulu Hou and Xuhong Liu designed the experimental plan. Qiang Tong and Zhihui Qin wrote and revised the manuscript. All authors have contributed to this research work. All authors have read and approved the final manuscript.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (No. 2018YFC0830202); supported in part by Qin Xin Talents Cultivation Program, Beijing Information Science and Technology University (2020); supported in part by Construction Project of Innovative Scientific Research Platform for Edge Computing (No. 2020KYNH105); and supported in part by the Opening Foundation of State Key Laboratory of Digital Publishing Technology.

References

[1] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[2] Y. Chen, N. Zhang, Y. Zhang, and X. Chen, "Dynamic computation offloading in edge computing for internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4242–4251, 2019.

[3] J. Huang, J. Liang, and S. Ali, "A simulation-based optimization approach for reliability-aware service composition in edge computing," *IEEE Access*, vol. 8, pp. 50355–50366, 2020.

[4] Y. Chen, Y. Zhang, Y. Wu, L. Qi, X. Chen, and X. Shen, "Joint task scheduling and energy management for heterogeneous mobile edge computing with hybrid energy supply," *IEEE Internet of Things Journal*, vol. 7, no. 9, p. 8419, 2020.

[5] J. Huang, C. Zhang, and J. Zhang, "A multi-queue approach of energy efficient task scheduling for sensor hubs," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 242–247, 2020.

[6] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2020.

[7] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, "Community-diversified influence maximization in social networks," *Information Systems*, vol. 92, pp. 1–12, 2020.

[8] X. Xu, S. Fu, W. Li, F. Dai, H. Gao, and V. Chang, "Multi-objective data placement for workflow management in cloud infrastructure using NSGA-II," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, p. 605, 2020.

[9] G. Ghemawat, H. Gobioff and S. T. Leung, The Google file system," in *Proceedings of the Nineteenth ACM symposium on Operating systems principles*, pp. 29–43, Landing NY USA, October 2003.

[10] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Proceedings of the 2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*, pp. 1–10, IEEE, Tahoe, NV, USA, May 2010.

[11] G. DeCandia, D. Hastorun, M. Jampani et al., "Dynamo," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6, pp. 205–220, 2007.

[12] R. T. Kaushik and M. Bhandarkar, "Greenhdfs: towards an energy-conserving, storage-efficient, hybrid hadoop computer cluster," in *Proceedings of the USENIX Annual Technical Conference*, vol. 109, p. 34, Boston, MA, USA, June 2010.

[13] A. Higai, A. Takefusa, and H. Nakada, "A study of effective replica reconstruction schemes at node deletion for HDFS," in *Proceedings of the 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 512–521, IEEE, Chicago, IL, USA, May 2014.

[14] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou, and Q. Ni, "Data-driven web apis recommendation for building web applications," *IEEE Transactions on Big Data*, 2020.

[15] L. Qi, C. Hu, X. Zhang et al., "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.

[16] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Transactions on Cloud Computing*, 2019.

[17] J. Huang, S. Li, and Y. Chen, "Revenue-optimal task scheduling and resource management for IoT batch jobs in mobile edge computing," *Peer-to-Peer Networking and Applications*, vol. 13, pp. 1776–1787, 2020.

[18] Y. Zhang, K. Wang, Q. He et al., "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Transactions on Services Computing*, 2019.

[19] Y. Zhang, J. Pan, L. Qi, and Q. He, "Privacy-preserving quality prediction for edge-based IoT services," *Future Generation Computer Systems*, vol. 114, pp. 336–348, 2021.

[20] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: an experimental comparison of time-series analysis and supervised learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 871–882, 2013.

[21] Y. Yu, J. Wang, M. Song, and J. Song, "Network traffic prediction and result analysis based on seasonal ARIMA and correlation coefficient,"vol. 1, pp. 980–983, in *Proceedings of the 2010 International Conference on Intelligent System Design and Engineering Application*, vol. 1, pp. 980–983, IEEE, Changsha, China, October 2010.

[22] S. Liu, M. Hu, F. Jeffrey, and Y. Yang, "Progress of grey system models," *Transactions of Nanjing University of Aeronautics and Astronautics*, vol. 29, no. 2, pp. 103–111, 2012.

[23] Y. Zhang, G. Cui, S. Deng, F. Chen, Y. Wang, and Q. He, "Efficient query of quality correlation for service composition," *IEEE Transactions on Services Computing*, 2018.

[24] N. M. Hindoriya, S. N. Singh, and S. K. Singh, "An adaptive wavelet neural network-based energy price forecasting in electricity markets," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1423–1432, 2008.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8822990, 13 pages
https://doi.org/10.1155/2020/8822990

Research Article

Deploying GIS Services into the Edge: A Study from Performance Evaluation and Optimization Viewpoint

Jianbing Zhang, Bowen Ma, and Jiwei Huang

Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum, Beijing 102249, China

Correspondence should be addressed to Jiwei Huang; huangjw@cup.edu.cn

Received 13 August 2020; Revised 27 September 2020; Accepted 16 October 2020; Published 31 October 2020

Academic Editor: Xiaolong Xu

Geographic information system (GIS) is an integrated collection of computer software and data used to view and manage information about geographic places, analyze spatial relationships, and model spatial processes. With the growing popularity and wide application of GIS in reality, performance has become a critical requirement, especially for mobile GIS services. To attack this challenge, this paper tries to optimize the performance of GIS services by deploying them into edge computing architecture which is an emerging computational model that enables efficient offloading of service requests to edge servers for reducing the communication latency between end-users and GIS servers deployed in the cloud. Stochastic models for describing the dynamics of GIS services with edge computing architecture are presented, and their corresponding quantitative analyses of performance attributes are provided. Furthermore, an optimization problem is formulated for service deployment in such architecture, and a heuristic approach to obtain the near-optimal performance is designed. Simulation experiments based on real-life GIS performance data are conducted to validate the effectiveness of the approach presented in this paper.

1. Introduction

Geographic information system (GIS) has been a hot technique for providing the tools for capturing, storing, analyzing, and displaying spatial data [1]. In order to provision GIS services with high Quality of Service (QoS), performance of the system is a critical issue [2]. In recent years, there have been several research works dedicating to optimizing the performance of GIS services from different aspects [2–4].

Edge computing is an emerging technique of optimizing computing systems by performing data processing at the edge of the network near the source of the original data [5]. It pushes applications, data, and services away from centralized points (i.e., the cloud) to the logical extremes of a network, and thus, the communication latency for processing user requests can be significantly reduced [6, 7], as well as fault-tolerance [8], privacy [9–12], and security [13] being enhanced. With edge computing architecture, the performance as well as scalability of GIS systems can be dramatically enhanced [14].

Although there have been some research studies focusing on improving the QoS of GIS services by applying edge computing techniques, few of them paid attention to the performance evaluation issue. There lacks of analytical approaches for evaluating as well as optimizing the performance of GIS systems which is able to quantitatively indicating the impact after deploying GIS services into the systems with edge computing paradigm. It is quite a challenging work to capture the dynamics of the GIS systems, especially after constructing them with edge computing architecture, since the introduction of the edge layer makes it quite complicated for task scheduling and request processing. Furthermore, whether to dispatch the request to the near-end edge servers or far-end cloud servers for obtaining the optimal QoS remains largely unexplored.

In this paper, we make an attempt at filling this gap by presenting a performance evaluation and optimization study of the GIS services deployed in the edge computing architecture. A theoretical model for capturing the dynamics of the edge computing systems running GIS services is presented, and its corresponding quantitative analysis is conducted. With the analytical results, an optimization problem is formulated and a service deployment scheme is designed for obtaining the near-optimal performance of GIS services. With performance data generated from real-world GIS systems, simulation experiments are conducted to validate the effectiveness of the approach.

The remainder of this paper is organized as follows. In Section 2, we discuss the related work most pertinent to this paper. In Section 3, we present a theoretical model for formulating the GIS systems with edge computing architecture, and provide quantitative analysis of the model. In Section 4, we formulate an optimization problem and design a performance optimization approach. In Section 5, we conduct real-life data based experiments to validate the efficacy of our scheme. Finally, we conclude the paper in Section 6.

2. Related Work

2.1. Performance Evaluation. A straightforward approach of performance evaluation is to obtain the performance metrics by direct measurement. Due to the dynamics of the system and environments, a series of experimental measurements are commonly required and statistical techniques are applied for handling the original measurement data. Truong and Karan [15] designed a mobile application of performance measurement and studied the impact of performance and data quality for mobile edge cloud systems. Morabito et al. [16] constructed a real testbed to evaluate the container-based solutions in IoT environment at the network edge, and analyzed the power and resource consumption for performance evaluation. Chen and Kunz [17] combined measurement and emulation and designed a network emulator for performance evaluation of optimal protocols. Qi et al. [18] collected data from 18,478 real-world APIs and 6,146 real-world apps, and designed a data-driven approach for web service recommendation. Baptista et al. [19] deployed a web-based GIS and used two datasets as the benchmark to evaluate the performance of several optimization techniques in Web GIS.

Although the measurement-based approaches are effective in performance evaluation, their overhead is so expensive that sometimes especially in the design phase of a computing system, one may not be able to afford implementing all the feasible schemes for comparison in reality [20]. Therefore, an alternative type of approaches has emerged, which applied theoretical models to formulate a system and then provide quantitative analysis by solving the models. With significantly lower overhead, the model-based approaches are able to evaluate the performance of the schemes before their implementations, making them increasingly popular in system design and improvement. Wang et al. [21] applied queueing theory to formulate an edge computing system, based on which a near-optimal offloading scheme for the Internet of Vehicles was designed. Ni et al. [22] generalized Petri net models and conducted performance evaluation of resource allocation strategies in edge computing environments. Li et al. [23] presented a performance estimation approach using M/M/k queueing model in Internet of Things (IoT) environments, which further helped to explore the optimal QoS-aware service composition scheme.

2.2. Performance Optimization. The performance optimization is commonly based on the evaluation results and thus used to optimize the performance of a system by designing new policies, selecting the best candidate, or enhancing the existing ones. One popular way is to collect the performance data of the policies by either measurement-based approaches or model-based approaches and search for the optimal one. Sometimes due to the extremely large search space, such search-based optimization approaches may meet with search-space explosion problems, and thus how to search for the optimal solution with high efficiency has become a hot topic. Mebrek et al. [24] considered the QoS and energy consumption in edge computing for IoT, formulated a constrained optimization problem, and designed an evolutionary algorithm-based approach for searching the feasible solutions. Wu et al. [25] designed a service composition scheme for mobile edge computing systems by combining simulated annealing and genetic algorithm. Zhang et al. [26] used neural network models for search-based optimization and designed a proactive video push scheme for reducing bandwidth consumption in hybrid CDN-P2P VoD Systems. Xu et al. [27] designed a multiobjective evolutionary algorithm based on decomposition for adaptive computation offloading for edge computing in 5G-envisioned Internet of Connected Vehicles (IoCV).

Another feasible way is to build a mathematical model illustrating the relationships between the system parameters and the performance metrics, based on which optimization problems can be formulated and optimal policies can be obtained. Zhang et al. [28] presented a graph-based model for service composition and designed an optimization approach of service composition with QoS correlations. Mao et al. [29] formulated the resource management as a Markov decision process, and further applied deep reinforcement learning to construct an optimization algorithm. Chen et al. [30] applied queueing theory to capture the dynamics in the mobile edge computing environment, formulated a stochastic optimization problem and designed an energy-efficient task offloading and frequency scaling scheme for mobile devices.

2.3. Summary. Although there have been several cutting-edge research works dedicating to performance evaluation and optimization for edge computing systems, this topic remains largely unexplored in geographic information systems. Since it has been shown by the existing literature that edge computing is able to improve the performance of the computing systems, especially for real-time services, we believe that a comprehensive study on the performance evaluation and optimization of GIS services deployed in edge computing architecture will have theoretical reference and practical value for the design, management, and improvement of geographic information systems.

Previously, we have conducted some research works on the topic of model-based performance evaluation and optimization in edge computing service systems. We have applied queueing network model to the performance evaluation of IoT services deployed in edge computing paradigm [31], and further put forward a simulation-based optimization approach of efficient service selection [32]. With queueing theory, we also proposed a multiqueue approach of energy-efficient task scheduling for sensor hubs in IoT using Lyapunov optimization technique [33]. In [34], we investigated the task scheduling and resource management problem and designed an equivalent linear programming problem which could be efficiently and elegantly solved at polynomial computational complexity. In addition, we have explored generalized stochastic Petri net models for model-based performance evaluation and search-based optimization for both performance and reliability metrics [35]. However, the performance modeling, analysis, and optimization meet with new challenges in the background of GIS, due to the characteristics of different task arrivals and service procedures. This paper is our first attempt at studying the model-based evaluation and optimization issue for GIS services.

3. Analytical Model for Performance Evaluation

In this section, we apply queueing theory to construct an analytical model for performance evaluation of GIS services in edge computing paradigm. We firstly present the atomic queueing model of a GIS server and then propose a queueing network model for evaluating the overall performance of an edge computing system. The quantitative analyses of the performance metrics are also presented by solving the models mathematically. The main notations and definitions which will be used in the following discussions are provided in Table 1.

3.1. Queueing Model of a GIS Server. An atomic service represents a type of relationship-based interactions or activities between the service provider and the service consumer to achieve a certain business goal or solution objective [36]. In a GIS system, there are a number of atomic services that can provide different functionalities. For example, users upload requests to view satellite pictures of a certain area, sensors upload the temperature, humidity, and other data of a certain area in real time, and servers analyze and process a large amount of existing data. Due to the difference in the amount of calculation, some services with a small amount of calculation can be usually completed on the local devices, while some services with heavy computational workload should be deployed on more powerful edge servers.

The dynamic behavior of atomic services includes the following three basic parts. First, the request arrives at the service node and completes specific tasks according to their needs. These requests can be simple requests from users, routine sensing tasks on sensors, or complex data analysis in data centers. Second, because the resources on the service node are not unlimited, requests sometimes have to wait in the queue until the service is available. If the current queue is empty, the incoming request will be processed by the service immediately without waiting in line. Third, after the request is processed, it leaves the system.

In a real-life GIS service system, a single server can handle a number of different types of services, and the capacity of each queue should be finite. Thus, we consider a multiqueue, finite-capacity, and single-server queueing model, where each queue specifically deals with tasks of the same priority.

It has been shown that the task arrivals above the session level in distributed systems can be basically formulated by Poisson distribution [37]. And according to the known data, we can figure out that the service rate of GIS system obeys the general distribution. Therefore, we formulate a GIS server by a q-M/G/1/Ki queueing model [38].

We consider a scenario consisting of a set $\mathbb{Q}$ of q ($|\mathbb{Q}| = q$) queues. Each queue q_i, where $i \in \mathbb{Q} = \{1, 2, \ldots, q\}$ specifically deals with tasks with the same priority, is connected to the same server. Usually, tasks arrive to q_i according to the i.i.d. Poisson process with rate λ_i and are processed by the server under a general independent service rate μ_i. The order in which the server accesses the queue is determined by the queue selection rule (QSR) or the queue scheduler. To facilitate our analysis, we define the state of the multiqueue model as a q-tuple array $x = [n_1, n_2, \ldots, n_q]$, where $n_i \in [0, K_i]$ represents the number of tasks in q_i at the current moment.

With this description, we can clearly describe the current occupation of each queue with the state vector x. Furthermore, we have to introduce a secondary variable s to describe the queue currently being serviced. In this sense, another form of $[x; s] \in R^{q+1}, s \in \{1, 2, \ldots, q\}$, can give a more compact representation. Figure 1 illustrates an example of a queueing model where $x = [3, 0, 2; 1]$.

Since the service time follows the general distribution, the memoryless feature of state evolution in traditional Markovian queueing models does not hold. To facilitate the analysis, we choose our observation time for the moments when the task has just completed its service procedure. At these points, the Markovian attribute is retained and the arrival and service processes are restarted. For the sake of distinction, $[x; s^*]$ (s with a superscript $*$) is used to emphasize the observation of time as the state of the moment of departure. It should be noted that the corresponding state probabilities of $[x; s]$ and $[x; s^*]$ are denoted as $p_{x;s}$ and $\pi_{x;s^*}$, respectively.

3.1.1. Queue Transition Probability (QTP). Considering the state $[x; s^*]$, the state transitions to this state can be either (i) from any arbitrary states $[x; r^*]$ or (ii) from the null state $[0, 0, \ldots, 0; r^*]$. And the QTP is different in these two cases. In Case (i), the QTP is related to the queue selection rule (QSR). For example, in the case of the QSR is FCFS (first-come-first-served), the corresponding queue transfer probability is

$$\beta_{r \longrightarrow s}^{\mathrm{FCFS}} = \frac{n_s}{\sum_{j=1}^{q} n_j}. \tag{1}$$

Table 1: Notations and definitions.

Notations	Definitions
t_i	The i-th terminal
T	Number of terminals $\mathcal{T}$
h_j	The j-th type of tasks
H	Number of applications $\mathscr{H}$
q_j	Size of offloading request for h_j
s_j	Size of offloading response for h_j
c_j	Amount of h_j's computation
$p_{i,j}$	Probability for t_i to generate h_j
$h_{i,j}$	Task generated by t_i for h_j
$\alpha_{i,j,0}$	Probability that $h_{i,j}$ is executed by t_i
$\alpha_{i,j,1}$	Probability that $h_{i,j}$ is offloaded form t_i to edge server
λ_i	Task generation rate of t_i
$h_{i,j}^{T}$	Task $h_{i,j}$ which is executed by t_i
$\lambda_{i,j}^{T}$	Task arrival rate of $h_{i,j}^{T}$
$h_{i,j}^{\text{Edge}}$	Task $h_{i,j}$ which is offloaded form t_i to edge server
$\lambda_{i,j}^{\text{Edge}}$	Task arrival rate of $h_{i,j}^{\text{Edge}}$
μ_i	Service rate of t_i
μ^{Edge}	Service rate of the edge server
$\mu_{i,j}^{T}$	Service rate of each task $h_{i,j}^{T}$
$\mu_{i,j}^{\text{Edge}}$	The service rate of each task $h_{i,j}^{\text{Edge}}$
$L_{i,j}^{T}$	Average queue length of $h_{i,j}^{T}$
$L_{i,j}^{\text{Edge}}$	Average queue length of $h_{i,j}^{\text{Edge}}$
$t_{i,j}^{T}$	Average response time of $h_{i,j}^{T}$
$t_{i,j}^{\text{Edge}}$	Average response time of $h_{i,j}^{\text{Edge}}$
$r_i^{T\longrightarrow\text{Edge}}$	Uplink transmission rate
$t_{i,j}^{T\longrightarrow\text{Edge}}$	Transmission delay from t_i to edge server
$r_i^{\text{Edge}\longrightarrow T}$	Downlink transmission rate
$t_{i,j}^{\text{Edge}\longrightarrow T}$	Transmission delay from edge server to t_i
ξ_i	Energy consumption for completing each computational unit of t_i
$e_{i,j}^{T}$	Energy consumption caused by executing $h_{i,j}^{T}$
ξ^{Edge}	Energy consumption for completing each computational unit in edge server
$e_{i,j}^{\text{Edge}}$	Energy consumption caused by executing $h_{i,j}^{\text{Edge}}$
ω_i	Transmission energy consumption of t_i
$e_{i,j}^{T\longrightarrow\text{Edge}}$	Energy consumption from t_i to edge server
ω^{Edge}	Transmission energy consumption of edge
$e_{i,j}^{\text{Edge}\longrightarrow T}$	Energy consumption from edge server to t_i
$t_{i,j}$	Total time consumption for executing $h_{i,j}$
$e_{i,j}$	Total energy consumption for executing $h_{i,j}$
τ	Balance factor between energy and time

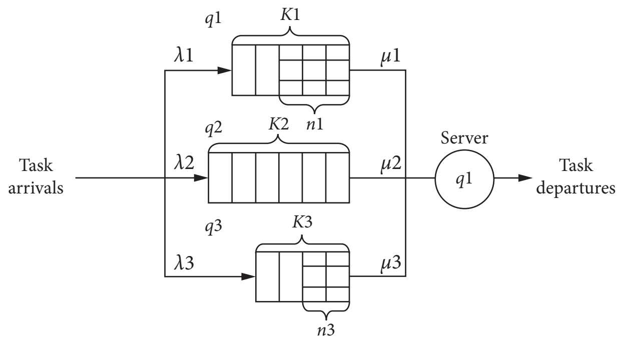


Figure 1: 3-M/G/1/Ki model when $K1 = 5$, $K2 = 6$, $K3 = 4$, $n1 = 3$, $n2 = 0$, $n3 = 2$, and $q1$ is being serviced.

However, in Case (ii), the QTP depends only on task arrival rates, which is represented as

$$\beta_{r\longrightarrow s} = \frac{\lambda_s}{\sum_{j=1}^{q} \lambda_j}. \tag{2}$$

In equation (2), the QSR is ignored since the QTP is merely related to the task arrival rates in Case (ii). For convenience, we do not need to label QSR unless it must be used.

3.1.2. Task Arrival Probability (TAP). The TAP of k arrival tasks during the service interval in the M/G/1/∞ model is represented as

$$\alpha_k = \int_0^{\infty} \frac{(\lambda t)^k}{k!} e^{-\lambda t} b(t)\mathrm{d}t, \quad 0 \le k < \infty, \tag{3}$$

where $b(t)$ is the probability density function (PDF) of the service time. When we solve the multiqueue model, the extension of α_k to multiqueue TAP is easily represented as

$$\begin{aligned}\alpha_{l1,l2,\ldots,l_q;s} &= \int_0^{\infty} \prod_{j=1}^{q} \left(\frac{\left(\lambda_j t\right)^{l_j}}{l_j!} e^{-\lambda_j t} \right) b_s(t)\mathrm{d}t, \quad 0 \le l_i < \infty, \\ &= \frac{1}{\prod_{i=1}^{q} l_i!} \int_0^{\infty} \left(\prod_{i=1}^{q} \left(\lambda_j t\right)^{l_j} \right) e^{-\sum_{m=1}^{q} \lambda_m t} b_s(t)\mathrm{d}t,\end{aligned} \tag{4}$$

where $\alpha_{l1,l2,\ldots,l_q;s}$ is expressed as the joint probability with l_k arrival tasks in q_k for $\forall k$ during the service interval of q_s, and $b_s(t)$ is the corresponding probability density function of the queue model. More specifically, the limited capacity of each queue should be taken into account. In the case of q-M/G/1/Ki, the formula in equation (4) needs to be modified properly further. Thus, since there are already n_i tasks in q_i, the maximum number of tasks allowed by q_i is $K_i - n_i$. And then the TAP can be expressed as $\sum_{m_i=K_i-n_i}^{\infty} \alpha_{l_1,\ldots,m_i,\ldots,l_q;s}$. Furthermore, assuming that the queues Q_{k+1} to Q_q are completely filled with tasks, $\alpha_{l_1,l_2,\ldots,l_q;s}$ is formulated as follows:

$$\sum_{m_{k+1}=K_{k+1}-n_{k+1}}^{\infty} \cdots \sum_{m_q=K_q-n_q}^{\infty} \alpha_{l_1,\ldots,l_k,m_{k+1},\ldots,m_q;s}. \tag{5}$$

3.1.3. State Transition Equations (STEs). After we have solved the QTP and TAP, the state probability $\pi_{x;s^*}$ of $[x; s^*]$ can be satisfied as the following STE to govern the dynamic of the queueing system:

$$\begin{aligned}\pi_{x;s^*} &= \sum_{r=1}^{q} \pi_{0,\ldots,0;r^*} \beta_{0\longrightarrow s} \alpha_{n1,\ldots,n_q;s} \\ &\quad + \sum_{r=1}^{q} \left[\sum_{l_1=0}^{n_1} \cdots \sum_{l_q=0}^{n_q} \pi_{n_1-l_1,\ldots,n_q-l_q;r^*} \times \beta_{r\longrightarrow s} \alpha_{l_1,\ldots,l_q;s} \right].\end{aligned} \tag{6}$$

In equation (6), the first term in the right-hand side is the probability from the null state to $[x; s^*]$, while the second term is the probability from $[x; r^*]$ to $[x; s^*]$. Based on the above formulation, the STEs composed of all feasible states can be expressed more concisely as a matrix-vector form:

$$\begin{aligned} A_{\pi}\pi &= 0, \\ \sum_{i=0}^{N_{q,\pi}} \pi_i &= 1, \end{aligned} \tag{7}$$

where $N_{q,\pi}$ is the number of all feasible states, π is the aggregation of $\pi_{x;s^*}$, and the state transition matrix $A_{\pi} \in R^{N_{q,\pi \times q,\pi}}$ consists of multiplications of QTP and TAP.

3.1.4. State Balance Equations (SBEs). Based on the QTP, TAP, and π to set up the SBEs, the state probability $p_{x;s^*}$ of $[x; s]$ is easily to be solved. According to the fact that the task flows must be conserved in the equilibrium status, SBEs can be expressed in the following equation:

$$\mathrm{IA}_{x;s} + \mathrm{ID}_{x;s}^{\mathrm{EMC}} = \mathrm{OA}_{x;s} + \mathrm{OD}_{x;s}^{\mathrm{EMC}}. \tag{8}$$

The arrival process, including $\mathrm{IA}_{x;s}$ and $\mathrm{OA}_{x;s}$, is owing to new arrival task to the queue system, for example, from $[x_r^-; s]$ to $[x; s]$ or from $[x; s]$ to $[x_r^+; s]$, which results in an increment of task during the service interval of q_s. Similarly, the departure process, including $\mathrm{ID}_{x;s}^{\mathrm{EMC}}$ and $\mathrm{OD}_{x;s}^{\mathrm{EMC}}$, is owing to departure task from the queue system, for example, from $[x; r^*]$ to $[x; s]$ or from $[x; s]$ to $[x_s^-; r]$, which results in a decrement of task in the departure instant of q_s. Thus, all of the state probabilities $p_{x;s}$ can be obtained.

(i) Null state $(x_z = [0, 0, \ldots, 0])$ probability $p_{x_z;0}$:

$$p_{x_z;0} = \frac{\lambda_{\mathrm{eff}}}{\sum_{i=1}^{q} \lambda_i} \sum_{i=1}^{q} \pi_{x_z;i*} \beta_{i\longrightarrow 0}. \tag{9}$$

(ii) Full-loaded state $(x_F = [K_1, K_2, \ldots, K_q])$ probability $p_{x_F;s}, s \ne 0$:

$$p_{x_F;s} = \mu_s^{-1} \lambda_{\mathrm{eff}} \psi_s - \sum_{n_1 \ne K_1 \text{ or} \ldots \text{or } n_q \ne K_q} P_{n_1,n_2,\ldots,n_q;s}, \tag{10}$$

where $\psi_s = \sum_{\forall n_i} \pi_{n_1,\ldots,n_s,\ldots,n_q;s^*}$.

(iii) Arbitrary state probability $p_{x;s}$ where $s \ne 0$:

$$p_{x;s} = \frac{\sum_{i=1}^{q} \lambda_i p_{x_i^-;s} + \lambda_{\mathrm{eff}} \sum_{i=1}^{q} \left(\beta_{i\longrightarrow s} \pi_{x;i^*} - \beta_{s\longrightarrow i} \pi_{x_s^-;i^*} \right)}{\sum_{i-1}^{q} \lambda_i}. \tag{11}$$

And then several performance measures can be obtained. For example, the average queue length L_s can be calculated by

$$L_s = \sum_{m=0}^{K_s} m P_{m;s} - \lambda_{\mathrm{eff}} \mu_s^{-1}, \tag{12}$$

where $P_{m;s}$ is the probability that there are m tasks in q_s and can be expressed by

$$P_{m;s} = \sum_{i=1}^{q} \sum_{n_s=m} P_{n_1,\ldots,n_s,\ldots,n_q;i}, \quad m > 0. \tag{13}$$

In equation (13), $\lambda_{\mathrm{eff}} = \lambda(1 - P_{k_s;s})$. In particular, $P_{k_s;s}$ is the probability when q_s is completely filled with tasks.

3.2. Queueing Network Model of an Edge Computing System. With the rapid development of the Internet and its applications, the single server cannot meet the needs of the vast majority of users, which is now replaced by a two-tier or even multitier group of server architecture. Therefore, we introduce edge server into the GIS system to provide higher quality of service.

All the users and sensors and other individuals who can send requests are called terminals. In the GIS system, the edge server can overwrite all the tasks request of the terminals. We define t_i as the i-th ($i \in \mathcal{T} = \{1, 2, \ldots, T\}$) terminal covered by the edge server E.

A terminal can run multiple applications concurrently, and each application may contain many different tasks. We use a set $\mathcal{H}$ ($|\mathcal{H}| = H$) to include all types of these tasks of all terminals in $\mathcal{T}$, and h_j ($j \in \mathcal{H} = \{1, 2, \ldots, H\}$) is expressed as the j-th type of tasks.

Each h_j is profiled by 3-tuple array $[q_j, s_j, c_j]$, which is characterized by the following: (i) q_j, the size of the task offloading request (including h_j's necessary description and parameters) for h_j sent by a terminal to the edge server; (ii) s_j, the size of the task offloading response (including h_j's execution result) for h_j received by a terminal from the edge server; (iii) c_j, the amount of h_j's computation.

t_i has a probability $p_{i,j}\,(p_{i,j} \in [0,1], \sum_{j \in H} p_{i,j} = 1)$ to generate h_j during its running period. And then we can use $h_{i,j}$ to express h_j generated by t_i. The total task generation rate of t_i is defined as λ_i.

There are two ways to completing $h_{i,j}$, i.e., (i) executing it locally, or (ii) offloading it remotely. On one hand, if $h_{i,j}$ is executed by t_i locally, time and energy consumption may be taken due to the low computing capability of m_i. On the other hand, if $h_{i,j}$ is offloaded to the edge server, it may suffer time and energy costs associated with the data transfer between t_i and the edge server although meanwhile it may benefit from edge server's powerful computing resources. Such tradeoff will be carefully balanced by an approach for obtaining global optimality which will be discussed in the next section.

We define $\alpha = \left\{\alpha_{i,j,k} \,|\, i \in T, j \in H, k = 1 \,\|\, k = 0\right\}$ as the selection probability to express the probability that terminal selects whether to execute the task locally or offload it to the edge server. For $h_{i,j}$, the value of $\alpha_{i,j}$ represents (i) the probability that $h_{i,j}$ is offloaded from t_i to the edge server, if $k = 1$; or (ii) the probability that $h_{i,j}$ is executed by t_i, if $k = 0$. And we have $\alpha_{i,j,0} + \alpha_{i,j,1} = 1$.

So far, we have been able to model the tasks generated by each terminal using the q-M/G/1/Ki model. For convenience, we define the task $h_{i,j}$ which is executed by t_i as $h_{i,j}^T$ and the task $h_{i,j}$ which is offloaded to the edge server as $h_{i,j}^{\text{Edge}}$. So, the task arrival rates $\lambda_{i,j}^T$ of $h_{i,j}^T$ can be expressed as

$$\lambda_{i,j}^T = \lambda_i p_{i,j} \alpha_{i,j,0}, \quad i \in \mathcal{T},\ j \in \mathcal{H}. \tag{14}$$

Similarly, the task arrival rates $\lambda_{i,j}^{\text{Edge}}$ of $h_{i,j}^{\text{Edge}}$ can be expressed as

$$\lambda_{i,j}^{\text{Edge}} = \sum_{i=0}^{T} \lambda_i p_{i,j} \alpha_{i,j,1}, \quad i \in \mathcal{T},\ j \in \mathcal{H}. \tag{15}$$

Then, we assume that the service rate for the terminal t_i is μ_i and the service rate for the edge server E is μ^{edge}. With μ_i μ^{edge} and the amount of h_j's computation c_j, the service rate $\mu_{i,j}^T$ of each task $h_{i,j}^T$ is easily obtained as

$$\mu_{i,j}^T = \frac{\mu_i}{c_j}. \tag{16}$$

Similarly, the service rate $\mu_{i,j}^{\text{Edge}}$ of each task $h_{i,j}^{\text{Edge}}$ is given by

$$\mu_{i,j}^{\text{Edge}} = \frac{\mu^{\text{edge}}}{c_j}. \tag{17}$$

Note that $\mu_i - \sum_{j \in \mathcal{H}} \lambda_i p_{i,j} \alpha_{i,j,0} c_j > 0, i \in \mathcal{T}$ and $\mu^{\text{Edge}} - \sum_{i=0}^{T} \sum_{j \in \mathcal{H}} \lambda_i p_{i,j} \alpha_{i,j,1} c_j > 0, i \in \mathcal{T}$ are the hard constraint, which means the service rate must be greater than the task arrival rate to make sure the queue is stable.

The average queue lengths of $h_{i,j}^T$ and $h_{i,j}^{\text{Edge}}$, represented by $L_{i,j}^T$ and $L_{i,j}^{\text{Edge}}$, respectively, can be obtained. Therefore, with Little's law, the average response time of $h_{i,j}^T$ and $h_{i,j}^{\text{Edge}}$ can be obtained from the following expression:

$$\begin{aligned} t_{i,j}^T &= \frac{L_{i,j}^T}{\lambda_{i,j}^T}, \\ t_{i,j}^{\text{Edge}} &= \frac{L_{i,j}^{\text{Edge}}}{\lambda_{i,j}^{\text{Edge}}}. \end{aligned} \tag{18}$$

In addition, the size of tasks' sending and receiving delays are so tiny that they can be ignored. And the time consumption caused by tasks to be offloaded on both terminals and edge server should be paid attention to. We define $r_i^{T \longrightarrow \text{Edge}}$ as the uplink data transmission rate from t_i to the edge server. Then, the transmission delay from t_i to the edge server can be given by

$$t_{i,j}^{T \longrightarrow \text{Edge}} = \frac{\lambda_i p_{i,j} q_j}{r_i^{T \longrightarrow \text{Edge}}}. \tag{19}$$

Similarly, the downlink data transmission rate from the edge server to t_i is denoted by $r_i^{\text{Edge} \longrightarrow T}$ delay:

$$t_{i,j}^{\text{Edge} \longrightarrow T} = \frac{\lambda_i p_{i,j} s_j}{r_i^{\text{Edge} \longrightarrow T}}. \tag{20}$$

4. Performance Optimization Approach for Service Deployment

4.1. Problem Formulation

4.1.1. Energy Consumption Analysis. In recent years, energy consumption has become a research hotpot in edge computing [39–41]. How to provide better services to meet the quality of service needs of users, while reducing the energy consumption of the systems and the operating cost of services, is one of the most important issues. It is different from [41], and we consider not only the energy consumption of mobile terminals, but also the energy consumption of edge server. In the GIS system, the energy consumption includes two aspects, i.e., task execution and task transmission.

We define the energy consumption caused by executing $h_{i,j}$ at t_i and caused by executing $h_{i,j}$ at the edge server as $e_{i,j}^T$ and $e_{i,j}^{\text{Edge}}$, respectively. And they can be expressed as follows:

$$\begin{aligned} e^{T}_{i,j} &= \xi_i \lambda_i p_{i,j} c_j, \\ e^{\text{Edge}}_{i,j} &= \xi^{\text{edge}} \lambda_i p_{i,j} c_j, \end{aligned} \tag{21}$$

where ξ_i and ξ^{Edge} are the energy consumed for each calculation at t_i and at the edge server, respectively.

Considering the energy consumption in the uplink data transmission process from t_i to the edge server, the energy consumption of t_i for the transmission is

$$e^{T\longrightarrow\text{Edge}}_{i,j} = \omega_i t^{T\longrightarrow\text{Edge}}_{i,j}, \tag{22}$$

where ω_i is the transmission energy consumption per unit time of t_i.

Similarly, the energy consumption of the edge server for the transmission is

$$e^{\text{Edge}\longrightarrow T}_{i,j} = \omega^{\text{Edge}} t^{\text{Edge}\longrightarrow T}_{i,j}, \tag{23}$$

where ω^{Edge} is the transmission energy consumption per unit time of the edge server.

The energy consumption used by the t_i to receive an offloading response is very low that it can be ignored. So far, we have got the tasks' response time and the energy consumption of task execution and transmission.

4.1.2. Utility Function. With the help of time and energy consumption of each part, we can build the corresponding utility function.

The total time consumed in executing the task includes two aspects: (i) the time consumption of terminal executing tasks, and (ii) the time consumption of the edge server executing tasks. In Case (i), the time consumption is caused by executing $h^{T}_{i,j}$ at t_i, that is, $t^{T}_{i,j}$. In Case (ii), the time consumption is caused by transmitting the offloading request of $h^{\text{Edge}}_{i,j}$ from t_i to the edge server, executing $h^{\text{Edge}}_{i,j}$ at the edge server and transmitting the offloading request of $h^{\text{Edge}}_{i,j}$ from the edge server to t_i, that is, $t^{T\longrightarrow\text{Edge}}_{i,j} + t^{\text{Edge}}_{i,j} + t^{\text{Edge}\longrightarrow T}_{i,j}$.

In summary, the total time consumption for executing h_{i_j} is easily obtained:

$$t_{i,j} = \alpha_{i,j,0} t^{T}_{i,j} + \alpha_{i,j,1}\left(t^{T\longrightarrow\text{Edge}}_{i,j} + t^{\text{Edge}}_{i,j} + t^{\text{Edge}\longrightarrow T}_{i,j}\right). \tag{24}$$

The total energy consumed in executing the task includes two aspects: (i) the energy consumption of terminal executing tasks and (ii) the energy consumption of the edge server executing tasks. In Case (i), the energy consumption is caused by executing $h^{T}_{i,j}$ at t_i, that is, $e^{T}_{i,j}$. In Case (ii), the energy consumption is caused by transmitting the offloading request of $h^{\text{Edge}}_{i,j}$ from t_i to the edge server, executing $h^{\text{Edge}}_{i,j}$ at the edge server and transmitting the offloading request of $h^{\text{Edge}}_{i,j}$ from the edge server to t_i, that is, $e^{T\longrightarrow\text{Edge}}_{i,j} + e^{\text{Edge}}_{i,j} + e^{\text{Edge}\longrightarrow T}_{i,j}$.

In summary, the total energy consumption for executing $h_{i,j}$ is easily obtained:

$$e_{i,j} = \alpha_{i,j,0} e^{T}_{i,j} + \alpha_{i,j,1}\left(e^{T\longrightarrow\text{Edge}}_{i,j} + e^{\text{Edge}}_{i,j} + e^{\text{Edge}\longrightarrow T}_{i,j}\right). \tag{25}$$

In general, total time consumption and total energy consumption in the GIS system can be easily obtained as $\sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{H}} t_{i,j}$ and $\sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{H}} e_{i,j}$, respectively.

Therefore, the utility function can be built to evaluate the overall benefit of the GIS system. We normalize the energy consumption and time consumption, and thus the utility function is defined as follows:

$$f = \tau \frac{\widetilde{t} - \sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{H}} t_{i,j}}{\widetilde{t}} + (1-\tau)\frac{\widetilde{e} - \sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{H}} e_{i,j}}{\widetilde{e}}, \tag{26}$$

where $\tau \in [0,1]$ is the balance factor between energy consumption and time consumption and $\widetilde{t} = \sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{H}} t^{T}_{i,j}$ and $\widetilde{e} = \sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{H}} e^{T}_{i,j}$ are the total time consumption and total energy consumption when the all tasks are executed in terminal without offloading, respectively. We should note that the closer τ is to 1, the more weight we put on time consumption. On the contrary, the closer τ is to 0, the more attention we pay on energy consumption. Therefore, τ should be set properly by the system manager to balance the tradeoff between performance and energy consumption according to the requirements in real-life scenarios.

4.1.3. Optimization Problem Formulation. With all the analytical results presented in the above sections, we formulate an optimization problem in GIS systems as follows:

$$\max \quad f, \tag{27}$$

$$\text{s.t.} \quad \mu^{\text{Edge}} - \sum_{i=0}^{T} \sum_{j\in\mathcal{H}} \lambda_i p_{i,j} \alpha_{i,j,1} c_j > 0, \quad i \in \mathcal{T}, \tag{28}$$

$$\mu_i - \sum_{j\in\mathcal{H}} \lambda_i p_{i,j} \alpha_{i,j,0} c_j > 0, \quad i \in \mathcal{T}, \tag{29}$$

$$\alpha_{i,j,k} \in [0,1], \quad i \in \mathcal{T},\ j \in \mathcal{H},\ k = 0\|k = 1\|, \tag{30}$$

$$\alpha_{i,j,0} + \alpha_{i,j,1} = 1, \quad i \in \mathcal{T}, j \in \mathcal{H}, \tag{31}$$

where constraints (28) and (29) are the hard constraint of the GIS system of each terminal and the edge server, which is used to make the queue system stable, respectively. And constraint (30) is the value range of $\alpha_{i,j,k}, i \in \mathcal{T}, j \in \mathcal{H}, k = 0\|k = 1$. Constraint (31) limits the total probability of offloading to the edge server, and local execution of each task is 1:

$$
\begin{aligned}
f &= \tau \frac{\widetilde{t}- = \sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{H}} t_{i,j}}{\widetilde{t}} + (1-\tau)\frac{\widetilde{e}- = \sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{H}} e_{i,j}}{\widetilde{e}}, \\
&= 1 + \frac{1-\tau}{\widetilde{e}}\left(-\sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{H}} \alpha_{i,j,0} e_{i,j}^{T} + \sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{H}} \left(1-\alpha_{i,j,0}\right)\left(e_{i,j}^{T\longrightarrow \text{Edge}} + e_{i,j}^{\text{Edge}} + e_{i,j}^{\text{Edge}\longrightarrow T}\right)\right) \\
&\quad + \frac{\tau}{\widetilde{t}}\left(\sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{H}} \alpha_{i,j,0} t_{i,j}^{T} + \sum_{i\in\mathcal{T}}\sum_{j\in\mathcal{H}} \left(1-\alpha_{i,j,0}\right)\left(t_{i,j}^{T\longrightarrow \text{Edge}} + t_{i,j}^{\text{Edge}} + t_{i,j}^{\text{Edge}\longrightarrow T}\right)\right).
\end{aligned}
\tag{32}
$$

4.2. Optimization Approach. Due to the complexity of the utility function, we propose a heuristic algorithm based on differential evolution (DE) algorithm [42, 43] which has good convergence properties with few control variables.

DE is a parallel direct search method which utilizes NP D-dimensional parameter vectors, expressed as

$$\underline{x}_{i,G}, \quad i = 1, 2, \ldots, \mathrm{NP}, \tag{33}$$

as a population for each generation G, where NP is the number of individuals in the population and does not change during the optimization process, i represents the i-th individual in the population, D is the dimension of the decision space. In a GIS system, if there are T terminals and H applications, the value of D is $T \times H$. And all the individuals in the population for the generation G is represented as $\underline{x}_G$. The j-th dimension variable of i-th individual in the population for the generation G is defined as

$$\underline{x}_{i,G}^{j}, \quad i = 1, 2, \ldots, \mathrm{NP},\ j = 1, 2, \ldots, D. \tag{34}$$

The DE algorithm includes the following four parts.

4.2.1. Initialization. As shown in Algorithm 1, if the system is unbeknown, the initial population should be chosen randomly.

4.2.2. Mutation. The core idea of DE is a new scheme for generating trial parameter vectors, which is called as mutation. DE generates new parameter vectors by using parameter F to add the weighted difference vector between two individuals to a third individual. For each vector $\underline{x}_{i,G}$ $(i = 0, 1, 2, \ldots, \mathrm{NP}-1)$, a perturbed vector $\underline{v}_{i,G+1}$ is generated according to Algorithm 2, with $r_1, r_2, r_3 \in [0, \mathrm{NP}-1]$, $i \neq r_1 \neq r_2 \neq r_3$. $F \in (0, 2)$ is a real and constant factor, which controls the amplification of the differential variation $(\underline{x}_{r_2,G} - \underline{x}_{r_3,G})$.

4.2.3. Crossover. In order to improve the diversity of the perturbed parameter vectors, crossover is introduced. To this end, the vector

$$\underline{u}_{i,G+1}, \tag{35}$$

with

$$
\underline{u}_{i,G+1}^{j} = \begin{cases} \underline{v}_{i,G+1}^{j}, & \text{for } j = \langle n\rangle_D, \langle n+1\rangle_D, \ldots, \langle n+L-1\rangle_D, \\ x_{i,G}^{j}, & \text{otherwise}, \end{cases}
\tag{36}
$$

is formed. The acute brackets $\langle\cdot\rangle_D$ denote the modulo function with modulus D. The starting index, $n \in [0, D-1]$, in equation (36) is a randomly chosen integer. The integer L, which represents the number of parameters that are going to be exchanged, is drawn from $[1, D]$ with the probability $Pr(L = v) = (CR)^{v-1}$, where $CR \in [0, 1]$ is the crossover probability. The random decisions for both n and L are made anew for each process of crossover. The crossover procedures are presented by Algorithm 3.

4.2.4. Selection. In order to decide whether the new vector $\underline{u}_{i,G+1}$ can become an individual in the population of generation $G+1$, it will be compared to $\underline{x}_{i,G}$. If vector $\underline{u}_{i,G+1}$ yields a larger objective function value which is the utility function in equation (34) than $\underline{x}_{i,G}$, $\underline{x}_{i,G+1}$ is set to $\underline{u}_{i,G+1}$, otherwise $\underline{x}_{i,G+1}$ retains $\underline{x}_{i,G}$. In addition, the optimal parameter vector $\underline{x}_{\text{best},G}$ is recorded for every generation G in order to keep track of the progress that is made during the optimization process. The selection scheme is formally presented in Algorithm 4.

Based on the following four parts, Algorithm 5 gives the main program of DE algorithm, which provides an approach on how to deploy the GIS services in the edge computing system. The near-optimal solutions for maximizing the utility function while satisfying the constraints can be obtained in an efficient way.

5. Evaluation

5.1. Experimental Setup. We conduct experiments based on the data collected from a real-world GIS system which has been deployed in reality providing real-time street view mapping services. The services are a kind of virtual reality service that provides end-users a 360-degree view panorama of the cities, streets, and other details. All the original data of the mapping services have been collected from real world by cars equipped with 3-dimensional laser scanners, global navigation satellite systems (GNSS), inertial measurement units (IMU), and panoramic cameras. Such original data have been stored in cloud data centers and processed by GIS servers. Upon the arrival of a task for users requesting a mapping service at a certain location, the task is firstly analyzed and initialized, and is divided into several subtasks

```
for $i = 1$ to NP do
    for $j = 1$ to $D$ do
        $\underline{x}^{j}_{i,G} = \underline{x}^{j}_{\min} + \text{rand}(0,1) \times (\underline{x}^{j}_{\max} - \underline{x}^{j}_{\min})$
    end for
end for
```

ALGORITHM 1: Initialization ($\underline{x}_G$).

```
$r_1, r_2, r_3 \in [0, NP - 1]$
$i \neq r_1 \neq r_2 \neq r_3$
$F \in (0, 2)$
$\underline{v}_{i,G+1} = \underline{x}_{r_1,G} + F \times (\underline{x}_{r_2,G} - \underline{x}_{r_3,G})$
```

ALGORITHM 2: Mutation ($\underline{x}_{i,G}$).

```
$n = \text{rand}[0, D]$ integer
$L = 1$
while (rand() < CR and $L < D$) do
    $L = L + 1$
end while
for $i = n$ to $n + D$ do
    $j = i\%D$
    if $i \geq n$ and $i \leq n + L - 1$ then
        $\underline{u}^{j}_{i,G+1} = \underline{v}^{j}_{i,G+1}$
    else
        $\underline{u}^{j}_{i,G+1} = \underline{x}^{j}_{i,G}$
    end if
end for
```

ALGORITHM 3: Crossover ($\underline{v}_{i,G+1}, \underline{x}_{i,G}$).

```
if $f(\underline{u}_{i,G+1}) > f(\underline{x}_{i,G})$ then
    $\underline{x}_{i,G+1} \longleftarrow \underline{u}_{i,G+1}$
else
    $\underline{x}_{i,G+1} \longleftarrow \underline{x}_{i,G}$
end if
if $f(\underline{x}_{i,G+1}) > f(\underline{x}_{\text{best},G+1})$ then
    $\underline{x}_{\text{best},G+1} \longleftarrow \underline{x}_{i,G+1}$
end if
```

ALGORITHM 4: Selection ($\underline{u}_{i,G+1}, \underline{x}_{i,G}, \underline{x}_{best,G+1}$)

```
$G \longleftarrow 1$
$Initialization\ (\underline{x}_G)$
while $G < T$ do
    $\underline{x}_{\text{best},G+1} = \underline{x}_{1,G}$
    for $i = 1$ to NP do
        $\underline{v}_{i,G+1} = Mutation\ (\underline{x}_{i,G})$
        $\underline{u}_{i,G+1} = Crossover\ (\underline{v}_{i,G+1}, \underline{x}_{i,G})$
        $\underline{x}_{i,G+1} = Selection\ (\underline{u}_{i,G+1}, \underline{x}_{i,G}, \underline{x}_{best,G+1})$
    end for
    $G \longleftarrow G + 1$
end while
```

ALGORITHM 5: Differential evolution of service deployment.

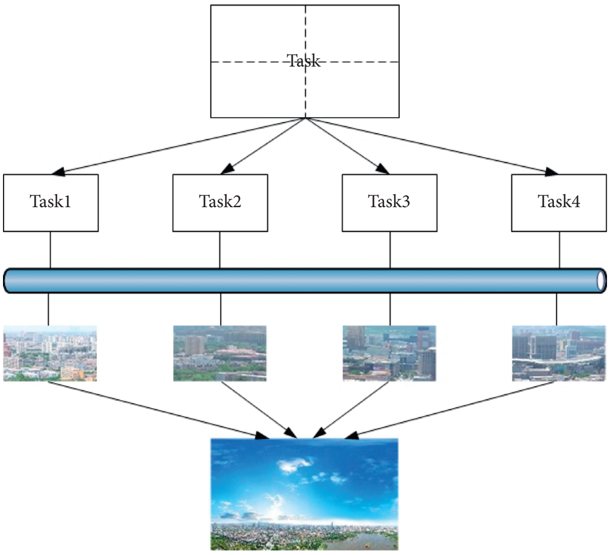


FIGURE 2: An example of GIS service workflow.

to be processed on a few cluster nodes in a parallel way. Each cluster node only processes a part of the original mapping data, and after completing the data processing, it returns the results to the centralized server for task convergence. The workflow of the GIS services is illustrated by Figure 2.

There are five nodes in our GIS systems. The centralized main server is equipped with an 8-core Intel Ice Lake CPU working at the maximum frequency of 4.7 GHz, and memory with capacity of 16 GB. Each cluster node has a CPU with 4 Intel Kaby Lake cores at maximum 3.8 GHz frequency as well as 16 GB or 8 GB memory.

The performance data are collected from such the GIS system during its service procedures for real-world users. We use the data to initialize the system parameters such as service rates and basic system architecture. Other parameters that we are not able to obtain from the system are set empirically shown as Table 2. Then, we apply our approach to analyze the impact of deploying the GIS services into edge computing architecture on the performance attributes, and validate our analytical results. During the experiments, we

Table 2: Parameter settings of the GIS system in experiments.

Name	Comment	Value
T	Number of terminals	[5, 25]
H	Number of tasks in each terminal	[10, 25]
λ_i	Task generation rate of t_i	[0, 1]/s
$p_{i,j}$	Probability for t_i to generate h_j	[0, 1]
q_j	Size of offloading request for h_j	[3, 5] MB
s_j	Size of offloading response for h_j	[1, 3] MB
c_j	Amount of h_j's computation	[200, 400] MI
μ_i	Service rate of t_i	$1.5\delta_i$ MIPS
μ^{Edge}	Service rate of the edge server	1000 MIPS
$r_i^{T\longrightarrow \mathrm{Edge}}$	Uplink transmission rate	[1, 3] MB/s
$r_i^{\mathrm{Edge}\longrightarrow T}$	Downlink transmission rate	[3, 5] MB/s
ξ_i	Each calculation energy consumption of t_i	[8, 10] × 10^{-3} J/MI
ξ^{Edge}	Each calculation energy consumption of edge	[3, 5] × 10^{-3} J/MI
ω_i	Transmission energy consumption of t_i	[100, 120] mW
ω^{Edge}	Transmission energy consumption of edge	[60, 80] mW
τ	Balance factor between energy and time	[0, 1]

also have to tune some system parameters for illuminating the effectiveness of our approach.

5.2. Experimental Results. In order to verify the applicability of the strategy, extensive simulations experiments are carried out to evaluate its efficacy. The simulation results demonstrate that the optimization approach based on the DE algorithm performs well in both utility function value and calculation time in different scenarios.

5.2.1. Efficacy Analysis. Although the DE algorithm cannot guarantee the global optimality, the simulation experiments show that the optimization algorithm has a strong global search ability. As shown in Figure 3, we illuminate the average utility values of population and their optimal values, which shows that our algorithm converges at about 300th generation.

We increase the dimension of decision space by increasing the number of terminals to 50. As shown in Figure 4, the algorithm converges at about 900th generation and the results are very close to the global optimal solutions.

With the further increase of the dimension of decision space by increasing the number of tasks in each terminal to 50, we find that the results converge over 1000th generations in Figure 5. The experimental results shown in Figures 3 to 5 validate that our approach performs well in solving large-scale optimization problems.

It has been well-known that, when the scale of the problem is small, the problem can be solved by some traditional optimization algorithms accurately. However, with the scale of the problem increases, the number of feasible solutions increases exponentially, which leads to the combination explosion of search space. And then, we analyze the calculation time of our algorithm in different dimension of decision space. Figure 6 shows that the computing time

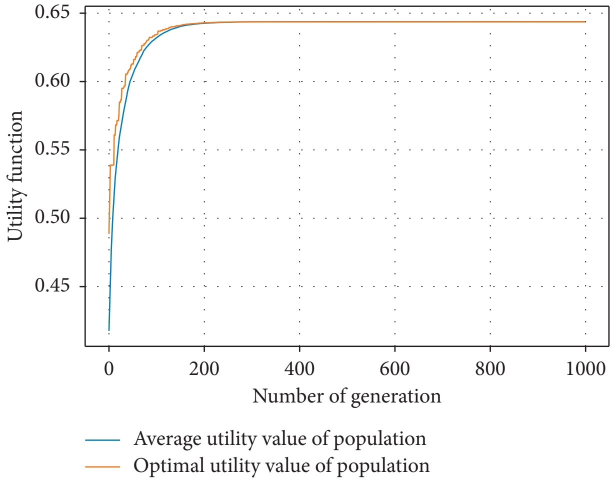


Figure 3: The utility function value of each generation by the DE algorithm, where $T = 5$ and $H = 10$.

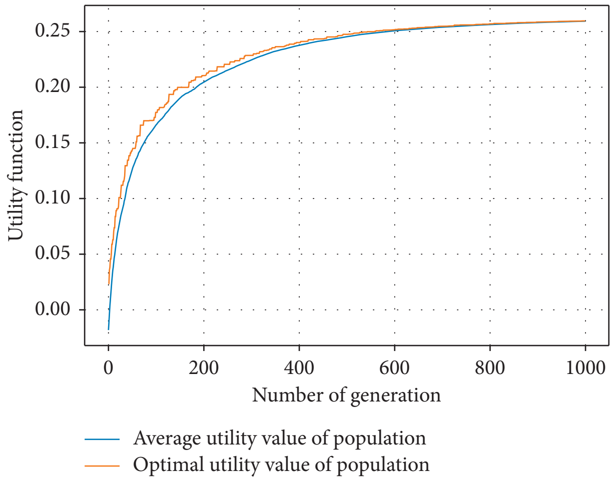


Figure 4: The utility function value of each generation by the DE algorithm, where $T = 50$ and $H = 10$.

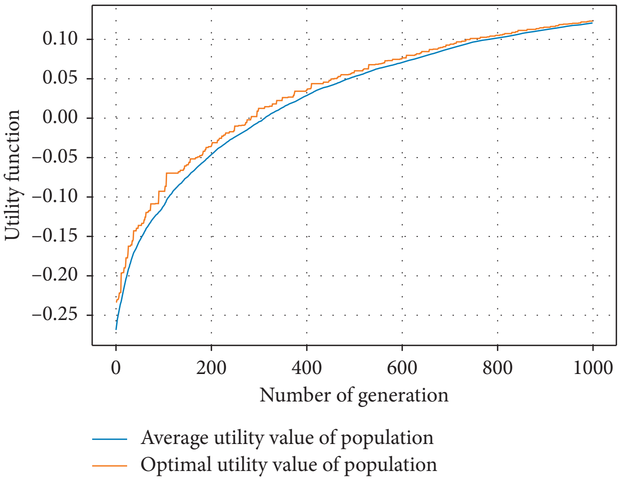


Figure 5: The utility function value of each generation by the DE algorithm, where $T = 50$ and $H = 50$.

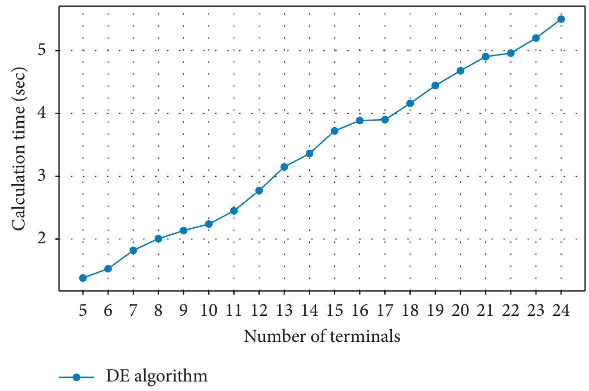


FIGURE 6: Empirical results of calculation time with the increase of the number of terminals.

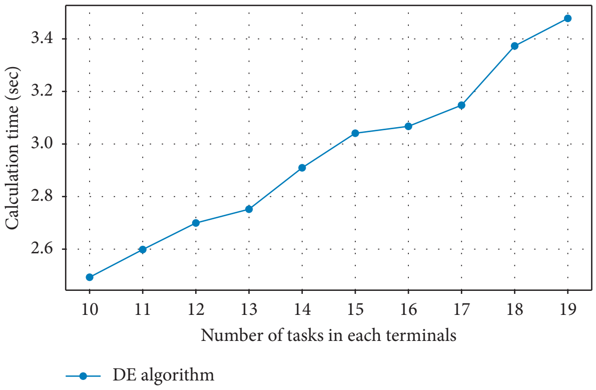


FIGURE 7: Empirical results of calculation time with the increase of tasks in each terminal.

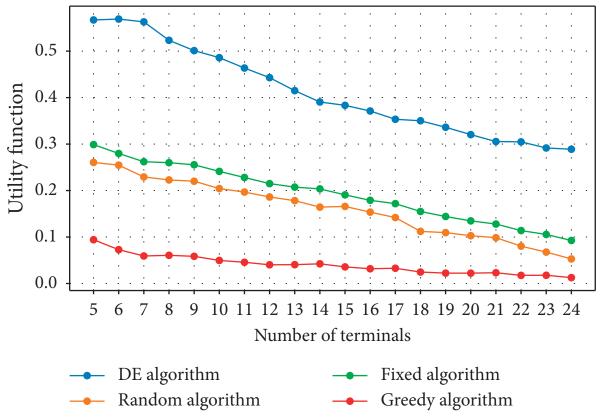


FIGURE 8: Empirical results of utility value with the increase of the number of terminals.

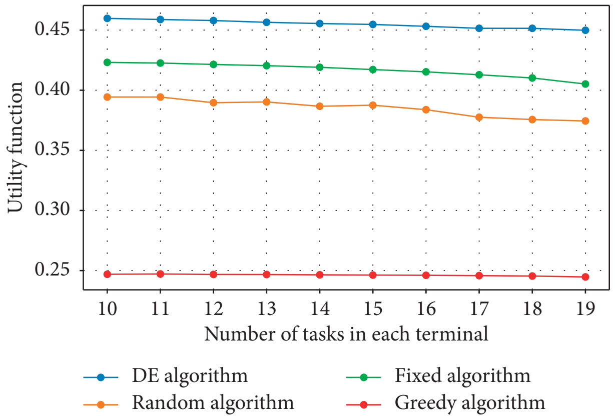


FIGURE 9: Empirical results of utility value with the increase of tasks in each terminal.

increases linearly with the number of terminals, where $H = 10$ and T increases from 5 to 24. Similarly, Figure 7 shows that the computing time increases linearly with the tasks of each terminal, where $T = 10$ and H increases from 10 to 19. Experimental results demonstrate that the DE algorithm is efficient in solving large-scale problems.

5.2.2. Comparison Analysis. Since there has been no existing well-developed scheme of service deployment optimization scheme for GIS services, we compare our approach with other three straightforward approaches which have been widely applied in practise. The first one is the random scheduling algorithm usually performs well in load balancing. The second approach is fixed algorithm which means that 50% of tasks are offloaded to the edge server. The third one is greedy algorithm in which tasks will be offloaded to the edge server as long as there is available resource.

We firstly tune the number of terminals T from 5 to 24, with fixed value $H = 10$ and $\tau = 0.5$, and the experimental results are shown in Figure 8. With the increase of T, the workload of the GIS system increases at both of the terminals and the edge server. Meanwhile, the time consumption and energy consumption increase so the utility function value decreases. Figure 8 also illustrates that our approach performs 50% better than the random algorithm, fixed algorithm, and greedy algorithm in terms of utility value.

We then tune the parameter H which is the number of tasks can be executed in each terminal from 10 to 19, and the empirical results are shown by Figure 9. We have similar conclusion that the scheme presented in this paper is 50% better than the random approach.

Finally, we discuss the impact brought by the balance factor τ, which trades off the weight between energy consumption and time consumption. The experimental results are shown in Figure 10. With the increase of τ, we pay more weight on optimizing the response time. In such scenario, introducing edge computing layer can benefit dramatically because of its additional computational capability. Since our algorithm is able to fully utilize the edge layer and optimize the global utility function, the utility values obtained by our DE approach are increasingly higher than random scheduling with the increase of τ.

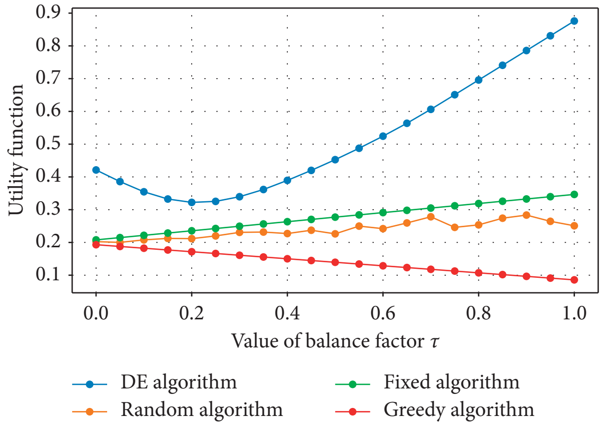


FIGURE 10: Empirical results of utility value with the increase of the balance factor between energy consumption and time consumption.

6. Conclusion

As GIS services become increasingly popular in daily life, the performance has drawn more and more attention. Deploying GIS services into edge computing architecture is an effective way for improving the performance. This paper conducts a quantitative study on the performance evaluation and optimization issue in deploying GIS services into the edge. Queueing models are presented for formulating the GIS services, and their corresponding analyses are provided in detail. Based on the analytical results, a heuristic approach is designed for obtaining the near-optimal solution of service deployment. Experiments based on the dataset collected from real-life GIS service systems are conducted, and the efficacy of the approach is validated. This work is expected to provide a theoretical reference of the evaluation and optimization of edge computing GIS systems.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Plan (no. 2016YFC0303700), National Natural Science Foundation of China (no. 61972414), Beijing Natural Science Foundation (no. 4202066), Beijing Nova Program (no. Z201100006820082), and the Fundamental Research Funds for Central Universities (no. 2462018YJRC040).

References

[1] M. Ehlers, "Remote sensing and geographic information systems: towards integrated spatial information processing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, no. 4, p. 763, 1990.

[2] S. Shekhar, S. Ravada, D. Chubb, and G. Turner, "Declustering and load-balancing methods for parallelizing geographic information system," *IEEE Transactions on Knowledge and Data Engineering*, vol. 10, no. 4, pp. 632–655, 1998.

[3] W. Li, M. Song, B. Zhou, K. Cao, and S. Gao, "Performance improvement techniques for geospatial web services in a cyberinfrastructure environment—a case study with a disaster management portal," *Computers, Environment and Urban Systems*, vol. 54, pp. 314–325, 2015.

[4] N. Torres-Cruz, I. Villordo-Jimenez, and A. Montiel-Saavedra, "Analysis of the geographical-information impact on the performance of ABS-CRE HetNets," *IEEE Latin America Transactions*, vol. 18, no. 3, pp. 613–622, 2020.

[5] P. G. Lopez, A. Montresor, D. Epema et al., "Edge-centric computing," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 5, pp. 37–42, 2015.

[6] Y. Chen, N. Zhang, Y. Zhang, and X. Chen, "Dynamic computation offloading in edge computing for internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4242–4251, 2019.

[7] M. Song, Y. Duan, T. Huang, and L. Zhan, "Inter-edge and cloud conversion accelerated user-generated content for virtual brand community," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 14, pp. 1–17, 2020.

[8] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2020.

[9] L. Qi, C. Hu, X. Zhang et al., "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, In press.

[10] Y. Duan, Z. Lu, Z. Zhou, X. Sun, and J. Wu, "Data privacy protection for edge computing of smart city in a DIKW architecture," *Engineering Applications of Artificial Intelligence*, vol. 81, pp. 323–335, 2019.

[11] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Transactions on Network Science and Engineering*, In press.

[12] Y. Zhang, J. Pan, L. Qi, and Q. He, "Privacy-preserving quality prediction for edge-based IoT services," *Future Generation Computer Systems*, vol. 114, pp. 336–348, 2021.

[13] Y. Duan, X. Sun, H. Che, C. Cao, Z. Li, and X. Yang, "Modeling data, information and knowledge for security protection of hybrid iot and edge resources," *IEEE Access*, vol. 7, pp. 99161–99176, 2019.

[14] R. Barik, H. Dubey, S. Sasane, C. Misra, N. Constant, and K. Mankodiya, "Fog2Fog: augmenting scalability in fog computing for health GIS systems," in *Proceedings of the 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pp. 241-242, Philadelphia, PA, USA, July 2017.

[15] H. Truong and M. Karan, "Analytics of performance and data quality for mobile edge cloud applications," in *Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD 2018)*, pp. 660–667, San Francisco, CA, USA, July 2018.

[16] R. Morabito, I. Farris, A. Iera, and T. Taleb, "Evaluating performance of containerized IoT services for clustered devices at the network edge," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 1019–1030, 2017.

[17] Y. Chen and T. Kunz, "Performance evaluation of IoT protocols under a constrained wireless access network," in *Proceedings of the 2016 International Conference on Selected Topics in Mobile Wireless Networking (MoWNeT 2016)*, pp. 1–7, Cairo, Egypt, April 2016.

[18] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou, and Q. Ni, "Data-driven web APIs recommendation for building web applications," *IEEE Transactions on Big Data*, In press.

[19] C. d. S. Baptista, C. P. Nunes, A. G. de Sousa, E. R. da Silva, F. L. Leite, and A. C. de Paiva, "On performance evaluation of web GIS applications," in *Proceedings of the 16th International Workshop on Database and Expert Systems Applications (DEXA'05)*, pp. 497–501, Copenhagen, Denmark, August 2005.

[20] Y. Zhang, K. Wang, Q. He et al., "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Transactions on Services Computing*, In press.

[21] Z. Wang, Q. Zhao, F. Xu, H. Dai, and Y. Zhang, "Detection performance of packet arrival under downclocking for mobile edge computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 9641712, 7 pages, 2018.

[22] L. Ni, J. Zhang, C. Jiang, C. Yan, and K. Yu, "Resource allocation strategy in fog computing based on priced timed Petri nets," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1216–1228, 2017.

[23] L. Li, S. Li, and S. Zhao, "QoS-aware scheduling of services-oriented internet of things," *IEEE Transactions on Industrial Informatics*, vol. 10, pp. 1497–1505, 2014.

[24] A. Mebrek, L. Merghem-Boulahia, and M. Esseghir, "Efficient green solution for a balanced energy consumption and delay in the IoT-fog-cloud computing," in *Proceedings of the 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA 2017)*, pp. 1–4, Cambridge, MA, USA, October 2017.

[25] H. Wu, S. Deng, W. Li, M. Fu, J. Yin, and A. Y. Zomaya, "Service selection for composition in mobile edge computing systems," in *Proceedings of the 2018 IEEE International Conference on Web Services (ICWS 2018)*, pp. 355–358, San Francisco, CA, USA, July 2018.

[26] Y. Zhang, C. Gao, Y. Guo et al., "Proactive video push for optimizing bandwidth consumption in hybrid CDN-P2P VoD systems," in *Proceedings of the 2018 IEEE Conference on Computer Communications (INFOCOM 2018)*, pp. 2555–2563, Honolulu, HI, USA, April 2018.

[27] X. Xu, Q. Wu, L. Qi, W. Dou, S.-B. Tsai, and M. Z. A. Bhuiyan, "Trust-aware service offloading for video surveillance in edge computing enabled internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, In press.

[28] Y. Zhang, G. Cui, S. Deng, F. Chen, Y. Wang, and Q. He, "Efficient query of quality correlation for service composition," *IEEE Transactions on Services Computing*, In press.

[29] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks (HotNets 2016)*, pp. 50–56, Atlanta, GA, USA, November 2016.

[30] Y. Chen, N. Zhang, Y. Zhang et al., "Energy efficient dynamic offloading in mobile edge computing for internet of things," *IEEE Transactions on Cloud Computing*, p. 1. In press.

[31] J. Huang, S. Li, Y. Chen, and J. Chen, "Performance modelling and analysis for IoT services," *International Journal of Web and Grid Services*, vol. 14, no. 2, pp. 146–169, 2018.

[32] J. Huang, Y. Lan, and M. Xu, "A simulation-based approach of QoS-aware service selection in mobile edge computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5485461, 10 pages, 2018.

[33] J. Huang, C. Zhang, and J. Zhang, "A multi-queue approach of energy efficient task scheduling for sensor hubs," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 242–247, 2020.

[34] J. Huang, S. Li, and Y. Chen, "Revenue-optimal task scheduling and resource management for IoT batch jobs in mobile edge computing," *Peer-To-Peer Networking and Applications*, vol. 13, no. 5, pp. 1776–1787, 2020.

[35] J. Huang, J. Liang, and S. Ali, "A simulation-based optimization approach for reliability-aware service composition in edge computing," *IEEE Access*, vol. 8, pp. 50355–50366, 2020.

[36] L.-J. Zhang, J. Zhang, and H. Cai, *Services Computing*, Springer, Berlin, Germany, 2007.

[37] E. Chlebus and J. Brazier, "Nonstationary poisson modeling of web browsing session arrivals," *Information Processing Letters*, vol. 102, no. 5, pp. 187–190, 2007.

[38] M.-S. Chen and H.-W. Yen, "A two-stage approach in solving the state probabilities of the multi-queueM/G/1 model," *International Journal of Systems Science*, vol. 47, no. 5, pp. 1230–1244, 2016.

[39] Y. Chen, Y. Zhang, Y. Wu, L. Qi, X. Chen, and X. Shen, "Joint task scheduling and energy management for heterogeneous mobile edge computing with hybrid energy supply," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8419–8429, 2020.

[40] X. Xu, S. Fu, W. Li, F. Dai, H. Gao, and V. Chang, "Multi-objective data placement for workflow management in cloud infrastructure using NSGA-II," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, pp. 605–615, 2020.

[41] W. Fan, Y. a. Liu, B. Tang, F. Wu, and Z. Wang, "Computation offloading based on cooperations of mobile edge computing-enabled base stations," *IEEE Access*, vol. 6, pp. 22622–22633, 2018.

[42] R. Storn and K. Price, "Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 23, no. 1, 1995.

[43] R. Storn, "On the usage of differential evolution for function optimization," in *Proceedings of the Fuzzy Information Processing Society, Nafips Biennial Conference of the North American*, Berkeley, CA, USA, June 1996.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8825749, 17 pages
https://doi.org/10.1155/2020/8825749

Research Article

High-Performance Routing Emulation Technologies Based on a Cloud Platform

Jianyu Chen, Xiaofeng Wang, Leiting Tao, and Yuan Liu

School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China

Correspondence should be addressed to Xiaofeng Wang; wangxf@jiangnan.edu.cn

Received 14 August 2020; Revised 3 October 2020; Accepted 10 October 2020; Published 24 October 2020

Academic Editor: Xiaolong Xu

Currently, the emergence of edge computing provides low-latency and high-efficiency computing for the Internet of Things (IoT). However, new architectures, protocols, and security technologies of edge computing need to be verified and evaluated before use. Since network emulation based on a cloud platform has advantages in scalability and fidelity, it can provide an effective network environment for verifying and evaluating new edge computing technologies. Therefore, we propose a high-performance emulation technology supporting the routing protocol based on a cloud platform. First, we take OpenStack as a basic network environment. To improve the performance and scalability of routing emulation, we then design the routing emulation architecture according to the software-defined network (SDN) and design the cluster scheduling mechanism. Finally, the design of the Open Shortest Path First (OSPF) protocol can support communication with physical routers. Through extensive experiments, we demonstrate that this technology not only can provide a realistic OSPF protocol but also has obvious advantages in the overhead and performance of routing nodes compared with those of other network emulation technologies. Furthermore, the realization of the controller cluster improves the scalability in the emulation scale.

1. Introduction

The Internet of Things (IoT) technology is an extension of the Internet and refers to the billions of physical devices around the world that are now connected to the Internet. Because of the arrival of super low-cost computer chips and the development of wireless networks [1], it is possible to turn anything into a part of the IoT, such as wearable Point-of-Care Testing (POCT) systems [2], which provides convenience for immediate diagnostic results. With the rapid growth of the IoT, a large amount of edge data has been explosively generated. Edge computing has emerged out of necessity to process these data [3–5]. Edge computing avoids problems of the slowdown in broadband expansion and delays in data transmission between central cloud servers and edge devices. A large amount of edge data can be processed at the edge-layer, which realizes low-latency and high-efficiency data processing.

At present, an increasing number of protocols and architectures for edge computing have been proposed [6–8]. Therefore, there is an urgent need for a testing platform for edge computing that provides network virtualization and computing realism at a low cost to test and verify these new protocols and architectures. Zeng et al. [9] show that a testing platform for edge computing mainly is composed of two aspects: the network and computing. It also proposes EmuEdge, a hybrid emulator based on Linux *netns* and Xen for full-stack edge computing emulation. Coutinho et al. [10] present a framework architecture to create virtualized fog environments that help researchers test and evaluate fog applications.

Although the above two methods can provide solutions for evaluating and testing edge computing, they have performance bottlenecks in network virtualization. In [9], the approach uses Linux *netns* and Xen to provide the basic environment for a testing platform for edge computing. However, the Xen virtualization is not as convenient and efficient as the cloud platform [11]. Coutinho et al. [10] use Mininet [12] to build an emulation network for fog virtualized environments. Compared with the cloud platform, Mininet may yield erroneous results if they inadequately manage the interaction between the emulation environment and the operating system that lies beneath the application and the virtual links [13].

Cloud platforms [14, 15] can provide a high-fidelity basic network environment for network emulation. OpenStack has become the standard for cloud platforms because it is open-source, scalable, and flexible [16–18]. Therefore, network emulation based on OpenStack is a research hotspot. At present, there are two routing emulation technologies based on OpenStack. One is to use the "qrouter" component of OpenStack to realize the routing emulation. Mengdong et al. [19] proposes a high-throughput routing emulation solution based on the "qrouter," but due to the lack of routing protocol design, its application scenarios are limited. Another is to integrate virtualization technologies, such as Kernel-based Virtual Machine (KVM) [20] and Docker [21], with routing software technologies, such as Quagga [22], XORP [23], and Click [24], to realize the routing emulation. The solution adopts a closed router architecture closely coupled with physical resources, operating systems, and network applications. Therefore, each routing node requires independent system space and relies on a virtual machine, which means that these solutions need to occupy a large number of physical resources to deploy a large-scale network emulation, thereby increasing costs. In terms of performance, the communication between virtual machines in OpenStack requires multiple forwarding of Linux-bridge [25] and Open vSwitch (OVS) [26], which leads to the problem of poor link performance of the routing emulation solution.

In this case, to reduce the overhead and improve the performance, this paper combines the software-defined network (SDN) [27] and OpenStack technologies to propose a high-performance routing emulation technology. OpenStack technology can provide a low-cost and realistic basic network emulation environment. SDN technology can optimize the architecture of this routing emulation technology. The routing node of the optimized architecture is responsible only for processing data packets according to flow rules distributed by the controller. Therefore, each routing node does not need an independent system space, which reduces the emulation overhead. In terms of the performance, because the routing node of the optimized architecture is implemented by the OVS bridge, it has high forwarding performance. We design the Open Shortest Path First (OSPF) protocol, which is a dynamic protocol and transmits routing information. In addition, we also design a cluster scheduling mechanism to schedule multiple controllers to provide the control service. The main contributions of this paper are summarized as follows:

(1) We propose a high-performance routing emulation architecture. This architecture can effectively reduce emulation overhead and improve emulation performance. In addition, the design of the load balancing of links can improve the throughput in the situation of congestion.

(2) We realize the OSPF protocol for this routing emulation technology. The realization of the OSPF protocol not only enables the routing node to communicate with the physical router but can also be used to analyse the OSPF attack.

(3) We design the controller cluster to strengthen the ability of control. The controller cluster ensures that all routing nodes can be controlled by multiple controllers simultaneously, which significantly improves the scalability in the emulation scale.

The rest of the paper is organized as follows. Section 2 describes some related work about network emulation. In Section 3, we introduce the architecture, routing function, and OSPF function. We introduce the load balancing of links in Section 4 and the load balancing of controllers in Section 5. Section 6 evaluates its function and performance. We conclude the paper in Section 7.

2. Related Work

Network emulation is one aspect of a testing platform for edge computing. The related routing emulation solutions are as follows.

Zeng et al. [9] present a hybrid emulator based on Linux *netns* and Xen for full-stack edge computing emulation, which is called EmuEdge. EmuEdge adopts *netns* for network-bounded node virtualization and provides full-system virtualization with Xen to combine the emulation of both the computation and network plane in an edge computing platform. Coutinho et al. [10] present a framework architecture to create fog testbeds in virtualized environments, and it uses Mininet to build an emulation network. Because the two solutions do not combine the cloud platform, they lack convenience, scalability, and scale.

NS3 [28], OPNET [29], and Glomosim [30] are widely adopted to simulate the network environment. Though these solutions can provide reproducible and convenient network simulation, they have defects in fidelity and ignore some essential details when acting. Therefore, these network simulators cannot offer realistic network results [31].

The cloud platform can provide an excellent essential environment for network emulation. In [19], an emulation technology is proposed based on a centralized routing engine and distributed router deployment. The technology uses the "qrouter" of Neutron [32] to construct a complex emulation network. However, this solution does not support the standard dynamic routing protocol, which leads to the inability of connected physical routers to learn routings from each other through the routing protocol and thus limits its application. Routing emulation software provides essential support for the routing emulation solutions based on cloud computing and virtualization technology. In [33], Quagga is deployed on a virtual machine, providing various routing function emulations. Huang et al. [34] use Linux Containers (LXC) [35] to provide a running environment for virtual routers and use the Xtensible Open Router Platform (XORP) or Click to build a routing node. However, the routing emulation solutions in [33, 34] build routing nodes by deploying routing emulation software on virtual machines, which leads to defects in overhead and performance of routing nodes in the routing emulation network.

3. Routing Emulation Design

3.1. Architecture Design. OpenStack, as the mainstream cloud platform technology, can provide outstanding operation and a basic network environment for network emulation with its high scalability, high fidelity, and low cost [36]. SDN technology optimizes the routing emulation architecture by separating the forwarding layer and the control layer, effectively reducing the emulation overhead and improving the emulation performance [27]. Therefore, we propose a high-performance routing emulation architecture based on OpenStack and SDN technology. The architecture consists of a 5-layer structure of the interconnection layer, data forwarding layer, control layer, decision layer, and cluster scheduling layer.

This multilayer structure can refine tasks and improve the scalability and reliability of the system [37]. The interconnection layer is the bottom layer of the architecture and is responsible for interconnecting with the OpenStack virtual network. The data forwarding layer is responsible for processing the data packets according to the flow rules. The control layer and decision layer are a vital part of the architecture. The control layer processes the information submitted by the data forwarding layer according to the fine-grained requirements specified by applications of the decision layer and distributes flow rules to each routing node. The decision layer provides various functions for the system through various applications, including monitoring, routing, and OSPF. The cluster scheduling layer relieves the pressure of the controller by scheduling multiple controllers to provide control services at the same time and improves the scalability of the emulation scale.

Figure 1 shows the architecture of routing emulation technology. The detailed introduction is as follows.

(1) Interconnection layer: routing emulation technology realizes communication with the OpenStack virtual network through the interconnection layer. With Neutron API [38], the interconnection layer uses an OpenStack virtual network to add a network interface for routing nodes. The routing nodes can send data packets to the OpenStack virtual network through this network interface.

(2) Data forwarding layer: this layer is composed of OVS bridge devices, and each bridge device represents a routing node in the emulation network. The network interface of each routing node is connected to the OVS of Neutron through the interconnection layer. Each node processes the data packets submitted by the interconnection layer according to the flow table rules distributed by the upper layer and submits information such as port status, traffic conditions, and network requests to the control layer.

(3) Control layer: the routing emulation technology manages the whole emulation network through centralized control. As an open-source SDN controller based on Python, Ryu [39] facilitates the study and research of researchers. Therefore, we adopt Ryu as the controller of the control layer. Ryu is responsible for processing the information submitted by the data forwarding layer according to the fine-grained requirements specified by applications of the decision layer and distributing flow rules to each routing node.

(4) Decision layer: the decision layer is crucial in the routing emulation architecture and consists of a network awareness app, network monitoring app, routing app, OSPF app, and a database. Each app performs its duties to achieve different functions. The network awareness app obtains the topology information of the emulation network and writes it into the database. The network monitoring app monitors the traffic and flow rules information of each routing node in real time and writes it into the database. The routing app is responsible for calculating all forwarding paths in the emulation network and generating flow rules based on these paths. The OSPF app is responsible for providing standard OSPF dynamic routing protocol [40] and supporting routing exchanges with physical routers. The network awareness app and routing app work together to provide the routing function, which will be described in Section 3.2. In Section 3.3, we will introduce how to realize the OSPF protocol with the OSPF app. Section 4 will introduce the load balancing of links realized by the network awareness app, network monitoring app, and routing app.

(5) Controller cluster: the control layer and decision layer compose the routing controller, and a group of routing controllers composes the controller cluster. Each routing controller will monitor its load information (including the number of connected nodes and the number of managed IP addresses) and submit it to the cluster scheduling layer through the message queue of RabbitMQ [41].

(6) Cluster scheduling layer: according to the load information of all routing controllers, the cluster scheduling centre can provide the load balancing of controllers, which will be introduced in Section 5.

3.2. Realization of High-Performance Routing Function. The routing function is the primary function of the routing node, which allows different networks to communicate with each other. Therefore, we realize the routing function through the network awareness app and routing app of the decision layer. The workflow of each app is shown in Figure 2.

According to Figure 2, the network awareness app is responsible for drawing the topology of the whole network, and the routing app is responsible for generating and distributing flow rules. First, the network awareness app obtains connection information by polling each routing node and draws the topology of the whole network based on this

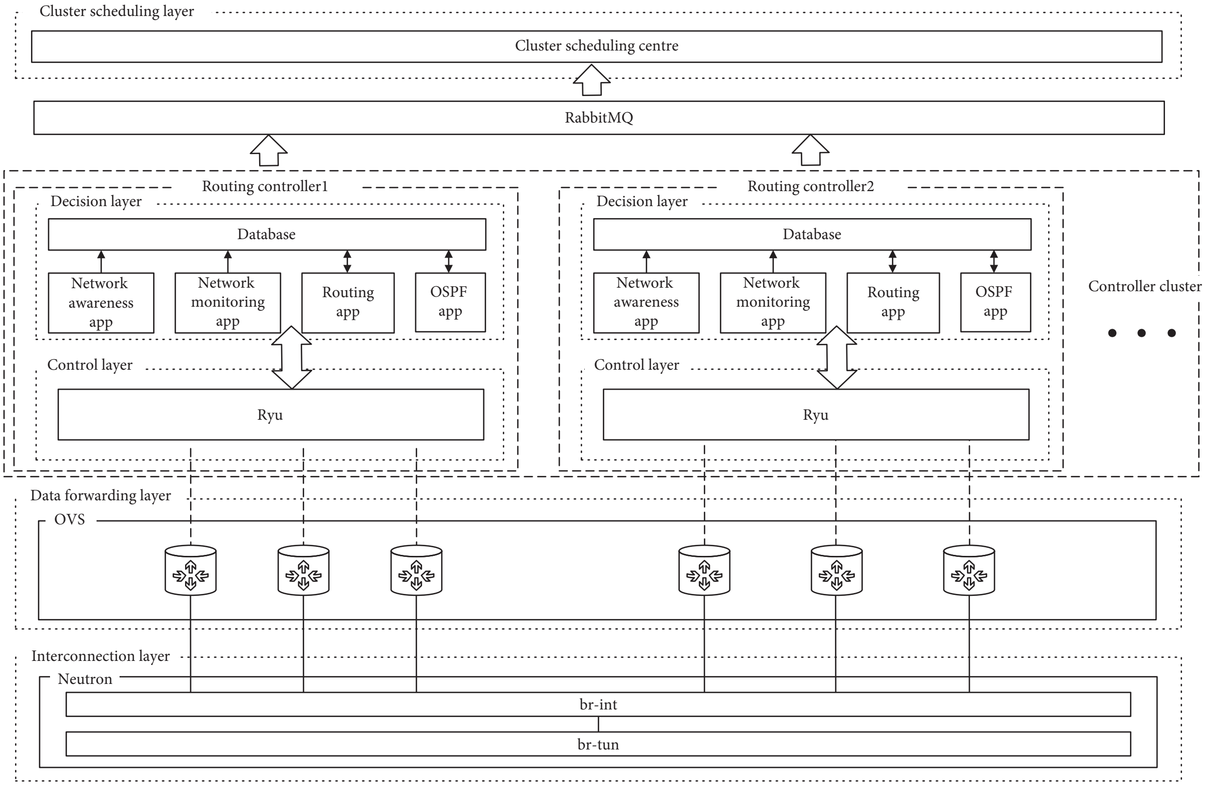


Figure 1: Architecture of routing emulation technology.

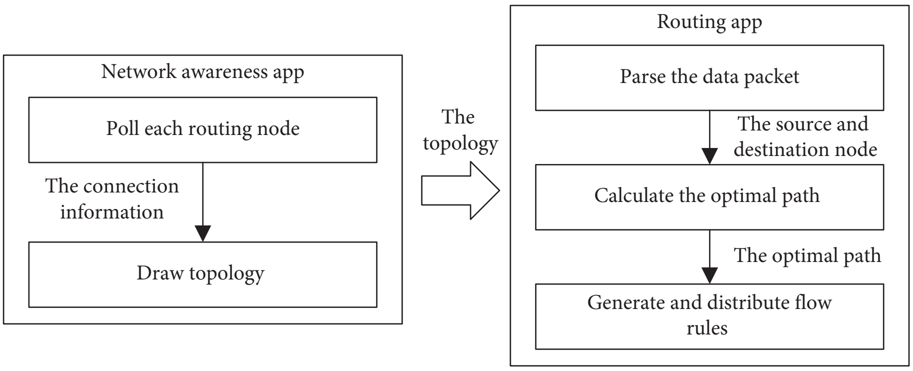

Network awareness app
Poll each routing node
The connection information
Draw topology
The topology
Routing app
Parse the data packet
The source and destination node
Calculate the optimal path
The optimal path
Generate and distribute flow rules

Figure 2: Workflow of the network awareness app and routing app.

information. Then, the routing app parses the data packet to obtain the information of the source and destination node and calculates the optimal path with the Dijkstra algorithm [42] between the two nodes according to the topology. Finally, the routing app generates flow rules based on the path and distributes them to the routing nodes in the path. Figure 3 shows the process of routing and forwarding.

As seen from Figure 3, when data packets arrive at the interconnection layer, they will be submitted to the data forwarding layer to match the flow rules (I in Figure 3). At the data forwarding layer, data packets will be matched to the flow rules based on the destination network, protocol type, and other fields. If the packets fail to match, the data forwarding layer will submit the header information of the packet to the control layer (II in Figure 3). Then, the control layer calculates the optimal path and generates flow rules with the network awareness app and routing app of the decision layer (III in Figure 3). Finally, the control layer distributes flow rules to routing nodes (IV in Figure 3). If the packets succeed to match, the TTL of the packets will be reduced by 1 to prevent the packets from generating loops in the network. The source media access control (MAC) of packets is modified to the MAC of the exit port. The destination MAC is modified to the MAC of the next hop. Then, the packets are forwarded to the exit port (V in Figure 3).

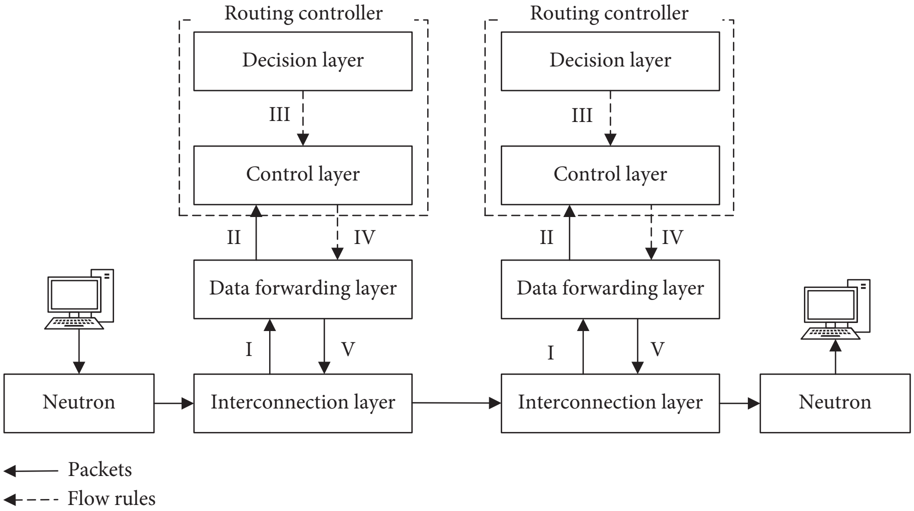


Figure 3: Process of routing and forwarding.

3.3. Realization of the OSPF Protocol. To support the construction of a complex emulation network with other emulation routers or physical routers through the OSPF protocol, we design the OSPF app at the decision layer. The OSPF protocol relies on five different types of messages to establish an OSPF neighbour adjacency and exchange routing information. The messages include Hello messages, Database Descriptor (DBD) messages, Link-State Request (LSR) messages, Link-State Update (LSU) messages, and Link-State Acknowledgement (LSAck) messages [40]. Therefore, the OSPF app also designs these five messages. Through neighbour discovery, database information exchange, routing calculation, and the flow rule distribution of the four steps, the app uses these messages to establish an OSPF neighbour adjacency and exchange routing information. The workflow of the OSPF app is shown in Figure 4.

The OSPF app calls different modules according to different message types. According to Figure 4, after the app parses the data packet, it will determine the type of the message. If it is a Hello message, the app will call the Hello module to respond to the message. If it is a DBD message, the app first calls the DBD module to support the exchange of link status information and then calls the LSR module to request detailed link status information. If it is an LSU message, the app will call the Daemon module to store the link status information and call the LSAck module to reply to the message. If it is an LSAck message, the app will call the Daemon module to record the message. Figure 5 shows the establishment process of OSPF.

When the routing node receives OSPF packets, the node will submit the packets to the routing controller for processing and reply to the neighbour router with the OSPF packets issued by the routing controller (I in Figure 5). When the routing controller receives OSPF packets, the OSPF app will construct corresponding reply packets according to the type of the OSPF message and issue them to the routing node (II in Figure 5). The specific process is as follows:

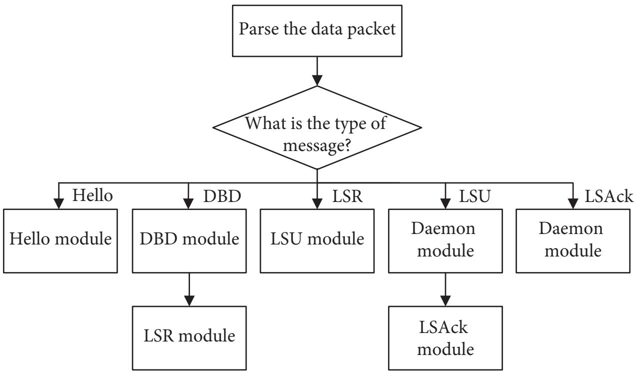


Figure 4: Workflow of the OSPF app.

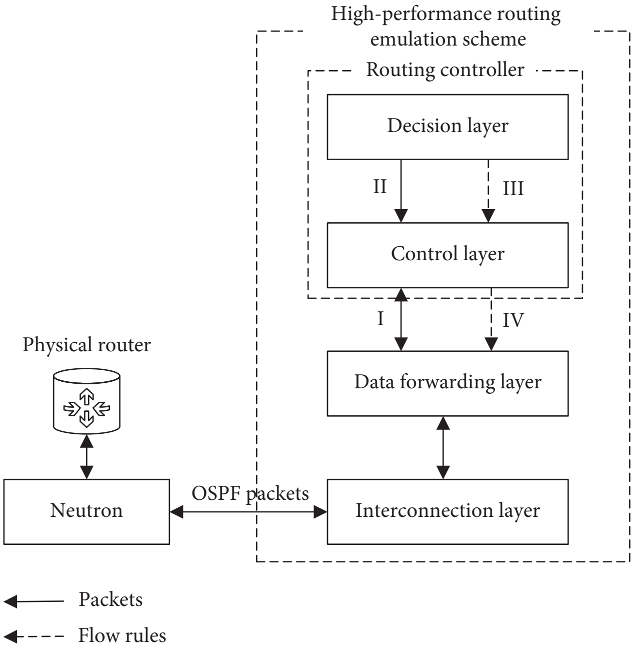


Figure 5: Establishment process of OSPF.

(1) Neighbour discovery: when the routing controller receives the neighbour Hello packets, the OSPF app constructs new Hello packets, and the control layer issues them to the routing node. Then, both sides establish an OSPF neighbour adjacency.

(2) Database information exchange: the OSPF app constructs DBD packets according to the link-state advertisement (LSA) information stored in the database to maintain the exchange of LSA information between the two sides. Then, according to the DBD packets of neighbour router, the app constructs LSR packets to request detailed LSA information.

(3) Routing calculation: according to all LSA information in the database, the OSPF app calculates all shortest path routings by the SPF algorithm [43] and stores them in the database (III in Figure 5).

(4) Flow rules distribution: after traversing the OSPF routes in the database, each route is resolved into forwarding flow table rules and distributed to routing nodes (IV in Figure 5).

Through the above four steps, the routing node establishes an OSPF neighbour adjacency and exchanges routing information with the physical router.

4. Realization of Load Balancing of Links

This paper proposes a load balancing of links mechanism to ensure that the emulation network can still obtain a high-throughput transmission path in the situation of congestion. The mechanism realizes load balancing of links through the network awareness app, network monitoring app, and routing app. The workflow of each app is shown in Figure 6.

According to Figure 6, the network awareness app is responsible for providing the topology of the emulation network. The network monitoring app is responsible for monitoring the load information of the emulation network (including the used bandwidth and the number of flow rules of a routing node) and calculating the load value of each path according to the topology and the evaluation model of the link. The routing app is responsible for calculating the optimal path according to the congestion avoidance algorithm based on the topology and the load value of each path. In Section 4.1, we will introduce the evaluation model of the link. The congestion avoidance algorithm will be introduced in Section 4.2.

4.1. Evaluation Model of the Link

4.1. Evaluation Model of the Link. In the process of network communication, bandwidth is the key factor affecting the transmission quality. The number of flow rules maintained by each routing node can reflect the load of the routing node. Therefore, we use the bandwidth and number of flow rules as two parameters of the model.

The bandwidth directly affects the speed of data transmission on the link. The ratio of used bandwidth can effectively reflect the busy degree of the link. In general, selecting the link with lower used bandwidth can yield higher-quality data transmission. The bandwidth used in the link can be obtained by the network monitoring app polling each routing node with the OpenFlow protocol [44]. We use (1) and (2) to obtain the bandwidth load value $B_{(i,j)}$ of the link (i,j) (i, j represent routing nodes i, j, and they are two adjacent nodes), where bandwidth_i and bandwidth_j, respectively, represent the port bandwidth of the nodes i and j, $\text{BW}_{(i,j)}$ represents the bandwidth of link (i,j) (since one link has a node on the left and right and the port bandwidth of each node is assumed to be different, we need to obtain the minimum of them as the bandwidth of the link), $\text{BW}_{\text{actual}(i,j)}$ represents the bandwidth actually used of the link (i,j), $\text{BW}_{(x,y)}$ represents the bandwidth of link (x, y) (x, y represent two adjacent nodes), and {nodes} is the set of all routing nodes:

$$\text{BW}_{(i,j)} = \min\left(\text{bandwidth}_i, \text{bandwidth}_j\right), \quad i \neq j, \tag{1}$$

$$B_{(i,j)} = 1 - \frac{\text{BW}_{(i,j)} - \text{BW}_{\text{actual}\,(i,j)}}{\sum_{x \neq y\text{a nd } x \text{ is adjacent to } y}^{x,y \in \{\text{nodes}\}} \text{BW}_{(x,y)}}, \qquad 0 \leq \text{BW}_{\text{actual}\,(i,j)} \leq \text{BW}_{(i,j)}. \tag{2}$$

Another parameter is the number of flow rules. The more the flow rules are maintained by the routing node, the more the time packets spend matching them. We use (3) and (4) to obtain the flow rule load value $F_{(i,j)}$ of the link (i, j) (i, j represent routing nodes i, j, and they are two adjacent nodes), where f_i and f_j represent the number of flow rules maintained by the routing nodes i, j (one link has a node on the left and right, so we take the average value of both nodes as $fn_{(i,j)}$ for convenience of calculation), fn_x represents the number of flow rules maintained by the routing node x, and {nodes} is the set of all routing nodes:

$$fn_{(i,j)} = \frac{fn_i + fn_j}{2}, \quad 0 \leq fn_i, fn_j, \ i \neq j, \tag{3}$$

$$F_{(i,j)} = \frac{fn_{(i,j)}}{\sum_{x \in \{\text{nodes}\}} fn_x}. \tag{4}$$

We use (5) to obtain the load value of the link (i, j) named as $L_{(i,j)}$ by weighting the sum of $B_{(i,j)}$ and $F_{(i,j)}$, where the weights are k_1 and k_2:

$$L_{(i,j)} = k_1 * B_{(i,j)} + k_2 * F_{(i,j)}, \quad k_1 + k_2 = 1. \tag{5}$$

4.2. Congestion Avoidance Algorithm. In this section, we will introduce the congestion avoidance algorithm. First, we obtain the topology of the emulation network through the network awareness app and load values (calculated by (5)) through the network monitoring app. We then use them as the input parameters of the congestion avoidance algorithm to calculate the optimal path in the situation of congestion. The pseudocode of the algorithm flow is shown in Algorithm 1.

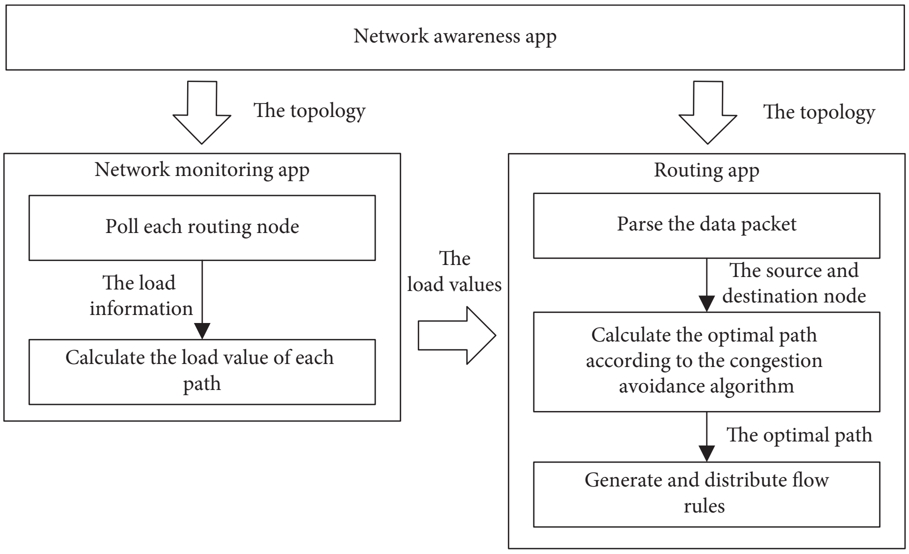

Network awareness app
The topology
The topology
Network monitoring app
Poll each routing node
The load information
Calculate the load value of each path
The load values
Routing app
Parse the data packet
The source and destination node
Calculate the optimal path according to the congestion avoidance algorithm
The optimal path
Generate and distribute flow rules

Figure 6: Workflow of the network awareness app, network monitoring app, and routing app.

```
Input:
nodes, weights, $s$, $d$;
//The nodes represents the set of all nodes, such as nodes = $\{R_1, \ldots, R_i, \ldots, R_n\}$. The weights represents the set of weights of all links, which are calculated by (5), such as weights = $\{(R_i, R_j) : L_{(Ri,Rj)}\}$. The $s$ represents the source node. The $d$ represents the destination node.
Output:
result;
//The result is a Boolean value, representing whether the optimal path is found.
Description:
Begin:
  result ← True;
  dict ← Dijkstra(nodes, weights, $s$);
  //nodes and $s$ are used to initialize the sets $S$ and $U$ in the Dijkstra algorithm, $S$ is the initial set of vertices (the initial set is $\{s\}$), and $U$ is the set of vertices to be selected. Weights are used to calculate the weight between two nodes (if there is no edge between the two nodes, the weight is infinite). The set of shortest paths from node $s$ to all other nodes can be obtained by the Dijkstra algorithm.
  If $d$ not in dict Then
    result ← False;
  Else
    path ← dict[$d$];
    //Get the shortest path from node $s$ to node $d$.
    Traverse all nodes in path and distribute flow rules;
End if
Return result;
```

Algorithm 1: Congestion avoidance algorithm.

5. Cluster Scheduling Mechanism

This paper adopts a routing emulation architecture that separates the forwarding layer and the control layer. The routing controller undertakes most of the work, including network awareness, network monitoring, and OSPF. Therefore, the pressure on the routing controller will increase as the number of routing nodes controlled by it increases. To solve this problem, this paper adopts the idea of distributed control and proposes a cluster scheduling mechanism. The mechanism balances multiple routing controllers to control more routing nodes simultaneously, which increases the scalability of the emulation network.

First, Section 5.1 describes how to realize the cluster scheduling mechanism. Then, we propose the evaluation model of the routing controller in Section 5.2 and propose the scheduling algorithm in Section 5.3.

5.1. Cluster Scheduling Centre. The cluster scheduling centre can manage multiple routing controllers simultaneously and evenly distributes the routing nodes in the emulation network to routing controllers for management.

As shown in Figure 7, routing controllers submit the load information (I in Figures 7 and 8) to the cluster scheduling centre through RabbitMq. The load information includes the number of routing nodes controlled by the controller, the number of IP addresses managed by the controller (because the routing node composed of the OVS bridge cannot set the IP address, the controller needs to manage the IP address of each access routing node), and the traffic of the southbound (it is the interactive traffic between the routing controller and routing nodes). The terminal is the entry point for creating a routing node. Whenever a routing node needs to be created, the terminal will send a request (II in Figures 7 and 8) to the cluster scheduling centre through API to obtain the idlest controller.

It can be seen from Figure 8 that when the load information is received, the centre will calculate the idlest controller according to the scheduling algorithm. In addition, it will return this controller information when receiving the request information.

5.2. Evaluation Model of the Routing Controller.

We propose an evaluation model of the routing controller and calculate the load situation of each controller according to the load information submitted by the routing controller.

The routing controller needs to provide the Transmission Control Protocol/IP (TCP/IP) stack [45] and routing calculation function to connect routing nodes. The more routing nodes are connected to the controller, the busier the controller is. Therefore, we define the value $RM_{i,}$ which represents the load situation of the ith controller on the number of routing nodes. The formula is shown as (6), where controller_i represents the number of routing nodes connected to the ith controller and N represents that there are N controllers:

$$\mathrm{RM}_i = \frac{\text{controller}_i}{\sum_{j=1}^{N} \text{controller}_j}, \quad 1 \le i \le N. \tag{6}$$

In the emulation network, the IP address of routing nodes is managed by the routing controller. The controller is responsible for answering ARP, ICMP, TCP, and UDP packets of these IP addresses. When the number of IP addresses is too large, it will reduce the speed of packet processing. Therefore, we define the value AM_i that represents the load situation of the ith controller on the number of IP addresses. The formula is shown as (7), where address_i represents the number of IP addresses managed by the ith controller and N represents that there are N controllers:

$$\mathrm{AM}_i = \frac{\text{address}_i}{\sum_{j=1}^{N} \text{address}_j}, \quad 1 \le i \le N. \tag{7}$$

When the southbound traffic is too heavy, the communication between controller and routing nodes will be blocked. Therefore, we define the value SF_i that represents the load situation of the ith controller on the southbound traffic. The formulas are shown as (8) and (9), where ST_i represents the southbound traffic of the ith controller, and it is calculated from the packet size passed within 30 seconds (packet_t represents the packet size at t). $SF_{\max}$ represents the maximum size of the specified southbound traffic:

$$\mathrm{ST}_i = \frac{\text{packet}_{t+\Delta t} - \text{packet}_t}{\Delta \mathrm{t}}, \quad \Delta t = 30_{\text{seconds}}, \tag{8}$$

$$\mathrm{SF}_i = \frac{\mathrm{ST}_i}{\mathrm{SF}_{\max}}, \quad 0 \le \mathrm{ST}_i \le \mathrm{SF}_{\max}. \tag{9}$$

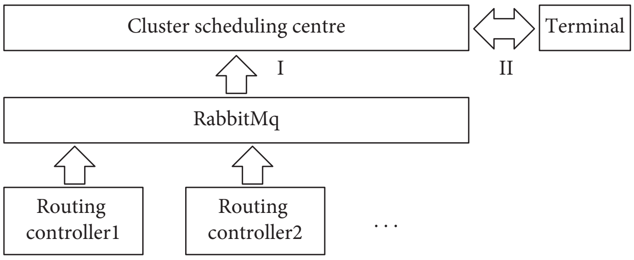


Figure 7: Process of information exchanging.

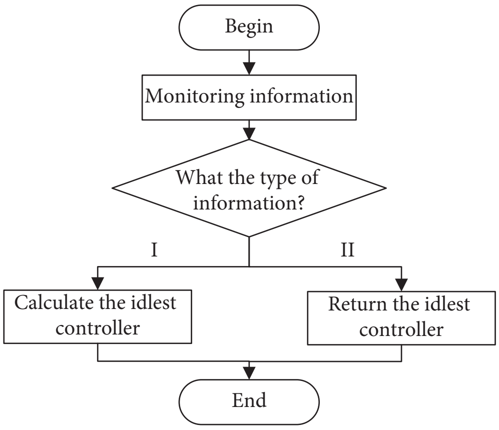

Begin
Monitoring information
What the type of information?
I
II
Calculate the idlest controller
Return the idlest controller
End

Figure 8: Workflow of the cluster scheduling centre.

5.3. Scheduling Algorithm.

This section elaborates on a scheduling algorithm. Combined with the evaluation model in the previous section, we can calculate the current idlest controller. The pseudocode of the algorithm flow is shown in Algorithm 2.

In the process of Algorithm 2, we calculate the busyness of each controller by using (10), where busyness = {id_1: value(id_1), . . ., id_i: value(id_i), . . ., id_n: value(id_n)}:

$$\begin{gathered}\text{value}(\text{id}_i) = k_1 * \mathrm{RM}(\text{id}_i) + k_2 * \mathrm{AM}(\text{id}_i) + k_3 * \mathrm{SF}(\text{id}_i), \\ \sum_{j=1}^{3} k_j = 1.\end{gathered} \tag{10}$$

Finally, we traverse busyness to find the least busy controller and return it.

```
Input:
controllers, nodes_num, addresses_num, southbound_flow;
//controllers contains all the controllers' ID, such as {$id_1$, . . ., $id_i$, . . ., $id_n$}. nodes_num contains the number of routing nodes
controlled by each controller, such as {$id_1$: $N_1$, . . ., $id_i$: $N_i$, . . ., $id_n$: $N_n$}, where $N_i$ represents the number of routing nodes.
addresses_num contains the number of IP addresses managed by each controller, such as {$id_1$: $A_1$, . . ., $id_i$: $A_i$, . . ., $id_n$: $A_n$}, where
$A_i$ represents the number of IP addresses. southbound_flow contains the size of southbound traffic of each controller, such as {$id_1$:
$T_1$, . . ., $id_i$: $T_i$, . . ., $id_n$: $T_n$}, where $T_i$ represents the size of southbound traffic.
Output:
free_controller;
//free_controller is a controller's ID that is the idlest among all controllers.
Description:
Begin:
free_controller ← None;
total_nodes ← SUM(nodes_num);
//Sum the number of routing nodes.
total_addresses ← SUM(addresses_num);
//Sum the number of IP addresses.
for con in controllers do
    $A$ ← nodes_num[con];
    //Get the number of routing nodes controlled by the controller con from nodes_num.
    Compute the RM according to A and total_nodes based on formula (6);
    $B$ ← addresses_num[con];
    //Get the number of IP addresses managed by the controller con from addresses_num.
    Compute the AM according to $B$ and total_addresses based on the formula (7);
    $C$ ← southbound_flow[con];
    //Get the size of the southbound traffic of the controller con from southbound_flow.
    Compute the SF according to C based on the formulas (8) and (9);
    Use formula (10) to obtain busyness according to RM, AM, and SF;
end for
free_controller ← MIN (busyness);
//Get the least busy controller.
return free_controller;
```

Algorithm 2: Controller scheduling algorithm.

6. Evaluation

In this section, we verify and evaluate the routing emulation technology of this paper. First, we verify the routing and forwarding function of routing nodes and OSPF function through experiments in Section 6.2. Then, we compare the routing emulation technology in this paper (named as *high-performance technology*), in [33, 34] (named as *virtualization technology*), and in [19] (named as *high-throughput technology*). Section 6.3 compares the cost of the three routing emulation technologies, and Section 6.4 compares the performance. Finally, in Sections 6.5 and 6.6, we, respectively, evaluated the effects of load balancing of links and load balancing of controllers, respectively.

6.1. Experimental Environment. In this paper, we use four Dell servers to build the OpenStack (version Queens) and use it as the basic environment of the experiment. The environment is shown in Figure 9, and the specifications of the servers are shown in Table 1.

The operating system of all the nodes is Ubuntu 16.04. We deploy the *high-performance technology* on the Compute1 node and deploy the *virtualization technology* or *high-throughput technology* on the Compute2 node for comparisons with our *high-performance technology* separately. R6 and R7 are Cisco's physical routers.

6.2. Verification of Routing Function Emulation. We build the emulation network, as shown in Figure 10, on the Compute1 node.

In Figure 10, VM1, VM2, and VM3, which represent clients, have IP addresses of 192.168.1.3/24, 192.168.6.3/24, and 10.1.9.3/24, respectively. One net represents a virtual network, and the details are shown in Table 2.

The emulation network is divided into a *high-performance technology* network and a physical network. The physical network is composed of two Cisco physical routers and connects to the *high-performance technology* network through the *net7* virtual network. The two networks use OSPF to exchange routing.

We take VM1 and VM2 as an example and use the traceroute tool [46] to test whether they can communicate with each other through forwarding routing nodes and whether the path the packet passes through is correct.

From Figure 11, we know that VM1 can communicate with VM2 and the path is correct. In the *high-performance technology* network, any two nodes can also communicate through the forwarding routing nodes to build an interconnected routing emulation network on OpenStack.

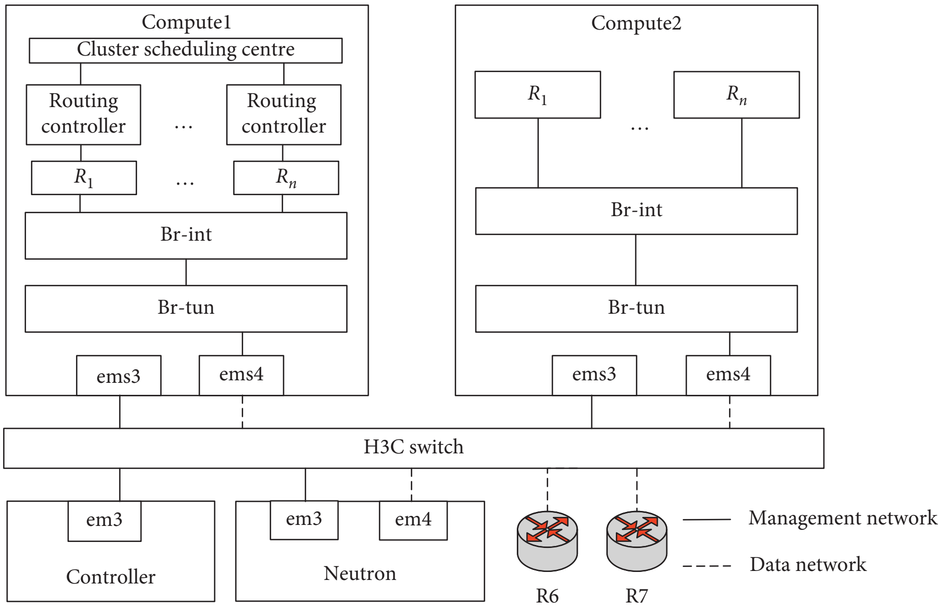


Figure 9: Experimental environment.

Table 1: Specifications of Dell servers.

Type	Controller node	Neutron node	Compute1 and Compute2 nodes
Model	R730	R730	R730
Core	8	6	6
Memory	64 GB	16 GB	64 GB
Disk	1 TB	1 TB	2 TB

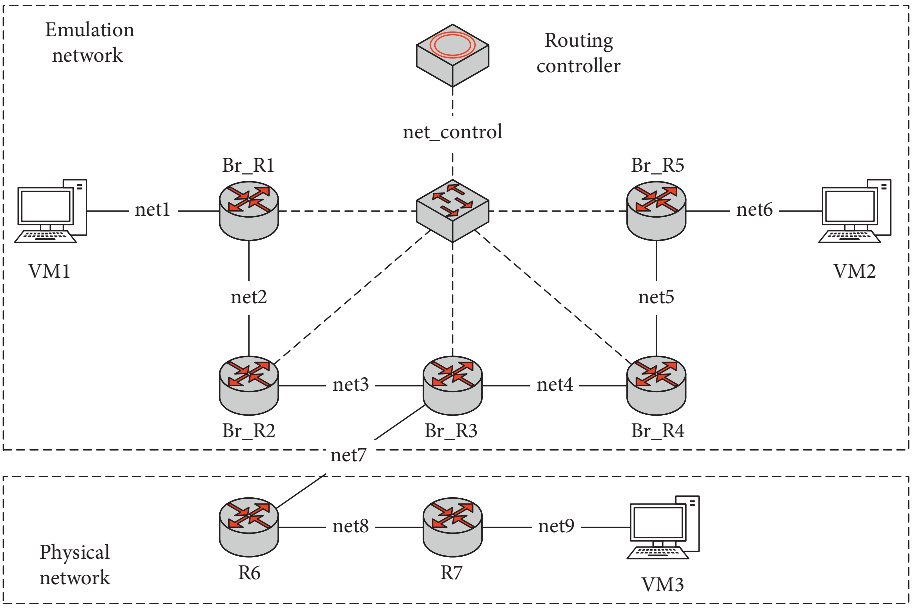


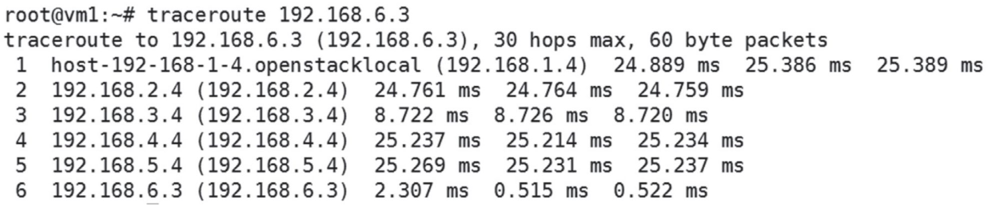
Figure 10: Topology for testing function.

We take VM1 and VM3 as an example and use the traceroute tool to test whether they can communicate over two kinds of networks and whether the path the packet passes through is correct.

From Figure 12, we know VM1 can communicate with VM3, and the path is correct. It can be proved that the emulation network constructed by the *high-performance technology* can exchange routings with the physical network by OSPF.

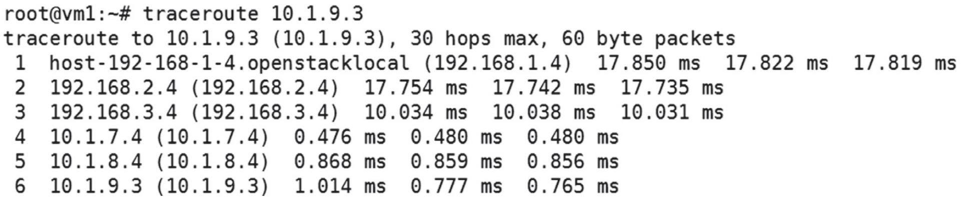
Table 2: Virtual network information.

Name	Segment
net1	192.168.1.0/24
net2	192.168.2.0/24
net3	192.168.3.0/24
net4	192.168.4.0/24
net5	192.168.5.0/24
net6	192.168.6.0/24
net7	10.1.7.0/24
net8	10.1.8.0/24
net9	10.1.9.0/24



```
root@vm1:~# traceroute 192.168.6.3
traceroute to 192.168.6.3 (192.168.6.3), 30 hops max, 60 byte packets
 1  host-192-168-1-4.openstacklocal (192.168.1.4)  24.889 ms  25.386 ms  25.389 ms
 2  192.168.2.4 (192.168.2.4)  24.761 ms  24.764 ms  24.759 ms
 3  192.168.3.4 (192.168.3.4)  8.722 ms  8.726 ms  8.720 ms
 4  192.168.4.4 (192.168.4.4)  25.237 ms  25.214 ms  25.234 ms
 5  192.168.5.4 (192.168.5.4)  25.269 ms  25.231 ms  25.237 ms
 6  192.168.6.3 (192.168.6.3)  2.307 ms  0.515 ms  0.522 ms
```

Figure 11: Traceroute path between VM1 and VM2.



```
root@vm1:~# traceroute 10.1.9.3
traceroute to 10.1.9.3 (10.1.9.3), 30 hops max, 60 byte packets
 1  host-192-168-1-4.openstacklocal (192.168.1.4)  17.850 ms  17.822 ms  17.819 ms
 2  192.168.2.4 (192.168.2.4)  17.754 ms  17.742 ms  17.735 ms
 3  192.168.3.4 (192.168.3.4)  10.034 ms  10.038 ms  10.031 ms
 4  10.1.7.4 (10.1.7.4)  0.476 ms  0.480 ms  0.480 ms
 5  10.1.8.4 (10.1.8.4)  0.868 ms  0.859 ms  0.856 ms
 6  10.1.9.3 (10.1.9.3)  1.014 ms  0.777 ms  0.765 ms
```

Figure 12: Traceroute path between VM1 and VM3.

In this section, we also verify whether the routing emulation technology in this paper supports OSPF protocol attack analysis. The adjacency attack [47] is a typical OSPF protocol attack method. We use VM1 and the routing node Br-R3 as an example to simulate this attack method.

The adjacency attack tampered with the routing table by simulating the virtual neighbour node of the OSPF router. Figure 13 shows that VM1 is simulated as the adjacent node of BR-R3 (192.168.3.100) and successfully establishes an OSPF neighbour with BR-R3 by sending OSPF messages, which verifies that the routing emulation technology in this paper supports OSPF protocol attack analysis.

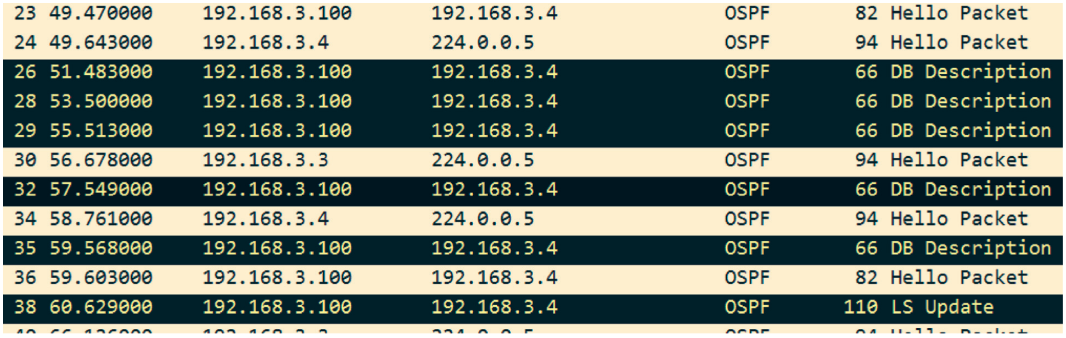
6.3. Comparisons of Overhead in Routing Emulation.

6.3. Comparisons of Overhead in Routing Emulation. The overhead of the routing emulation technology determines the size of the network that can be emulated in the same physical resource. In this section, we compare the CPU and memory consumption of the virtualization technology, high-throughput technology, and high-performance technology.

Tables 3 and 4 show the results of comparing how much CPU and memory are occupied when routing nodes are forwarding packets. In terms of CPU, compared with the *virtualization technology*, *high-performance technology* and *high-throughput technology* have a significant advantage. When there are 10 nodes, the *virtualization technology* takes up 45.6% of the CPU, which is 50.6 times of the *high-performance technology*. In terms of memory, *high-performance technology* has a significant advantage. When there are 10 nodes, the *high-performance technology* takes up 78 MB of the memory, which is 97.9% less than the *virtualization technology* and 92.3% less than the *high-throughput technology*. According to the above comparison results, using *high-performance technology* to build an emulation network can effectively reduce the costs.

6.4. Comparisons of Performance in Routing Emulation. The throughput, delay, and packet loss rate determine the performance of a routing emulation system. In this section, we compare the *virtualization technology*, *high-throughput technology*, and *high-performance technology*.

To more accurately verify the advantages of the *high-performance technology* on the performance of the routing node, two experimental scenarios are designed: the multihop situation and concurrent situation. We design a linear topology with 10 routing nodes, as shown in Figure 14, and design a concurrent topology with 5 clients, as shown in Figure 15. We use the iPerf3 [48] tool to test the throughput and packet loss rate and use the Ping tool to test [49] the delay.



```
23 49.470000   192.168.3.100   192.168.3.4   OSPF    82 Hello Packet
24 49.643000   192.168.3.4     224.0.0.5     OSPF    94 Hello Packet
26 51.483000   192.168.3.100   192.168.3.4   OSPF    66 DB Description
28 53.500000   192.168.3.100   192.168.3.4   OSPF    66 DB Description
29 55.513000   192.168.3.100   192.168.3.4   OSPF    66 DB Description
30 56.678000   192.168.3.3     224.0.0.5     OSPF    94 Hello Packet
32 57.549000   192.168.3.100   192.168.3.4   OSPF    66 DB Description
34 58.761000   192.168.3.4     224.0.0.5     OSPF    94 Hello Packet
35 59.568000   192.168.3.100   192.168.3.4   OSPF    66 DB Description
36 59.603000   192.168.3.100   192.168.3.4   OSPF    82 Hello Packet
38 60.629000   192.168.3.100   192.168.3.4   OSPF   110 LS Update
```

Figure 13: Process of the adjacency attack.

Table 3: CPU consumption comparison.

Number of nodes	Virtualization technology (%)	High-throughput technology (%)	High-performance technology (%)
2	9	0.3	0.2
4	19.1	0.5	0.4
6	30.6	0.6	0.5
8	39.3	0.8	0.7
10	45.6	1.0	0.9

Table 4: Memory consumption comparison.

Number of nodes	Virtualization technology (MB)	High-throughput technology (MB)	High-performance technology (MB)
2	907	852	78
4	1712	877	80
6	2519	998	81
8	3229	1054	83
10	4000	1092	84

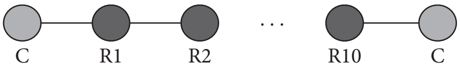


Figure 14: Multihop topology for testing.

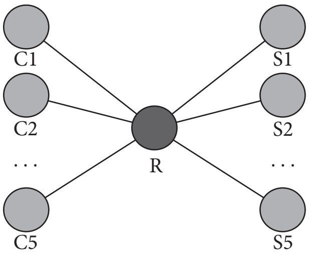


Figure 15: Concurrent topology for testing.

Figure 16 shows the comparison of the throughput, delay, and packet loss rate of the three routing emulation technologies in the multihop situation.

In terms of the throughput, in the 2-hop situation, the throughput of the virtualization technology is 11.8 Gbit/s and that of the high-throughput technology is 14.6 Gbit/s. However, the throughput of the *high-performance technology* is 17.2 Gbit/s, which is 1.46 times that of the *virtualization technology* and 1.18 times that of the *high-throughput technology*. With the increase in the number of hops, the *high-performance technology* still has advantages. In the 10-hop situation, the throughput of the *high-performance technology* is 2.49 times of the *virtualization technology* and 1.24 times that of the *high-throughput technology*.

In terms of the delay time, in the 2-hop situation, the delay time of the *virtualization technology* is 1.071 ms and that of the *high-throughput technology* is 0.73 ms. However, the delay time of the *high-performance technology* is 0.456 ms, which is reduced by 57% compared to the *virtualization technology* and reduced by 38% compared to the *high-throughput technology*. With the increase in the number of hops, the delay time of the *high-throughput technology* and *high-performance technology* appears to be stable, but that of the *virtualization technology* increases linearly.

In terms of the packet loss rate, the high-throughput technology and high-performance technology tend towards 0, and with the increase in the number of hops, the packet loss rate is still stable at 0. However, in the 2-hop situation, the packet loss rate of the virtualization technology is 10%, and it will increase as the number of hops increases.

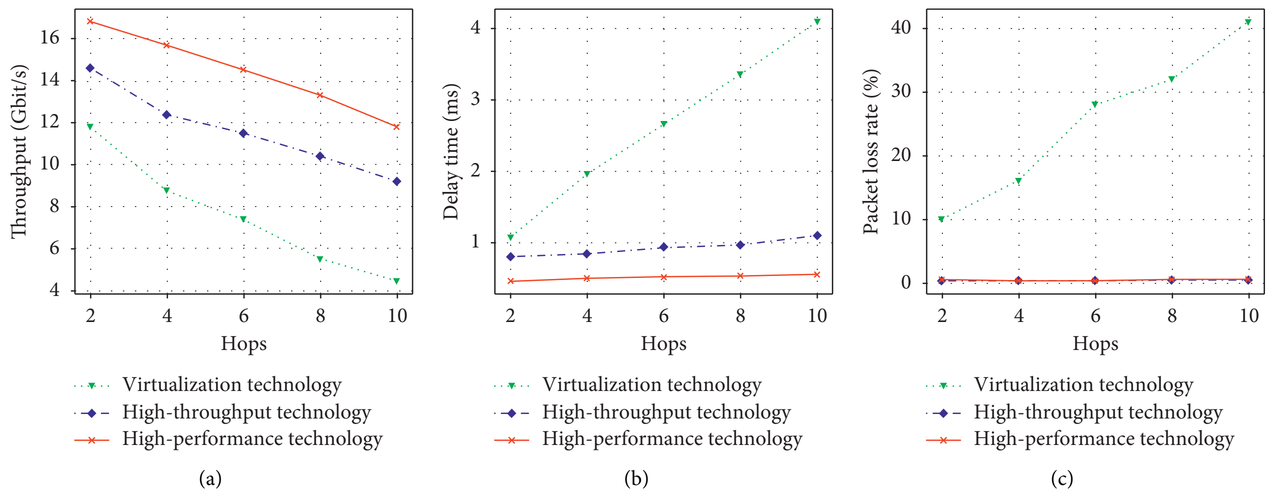


Figure 16: Comparison of performance in the multihop situation.

Therefore, in the multihop situation, the high-performance technology performs well in terms of the throughput, delay, and packet loss rate.

Figure 17 shows the comparison of the throughput, delay, and packet loss rate of the three routing emulation technologies in the concurrent situation.

In terms of the throughput, with the increase in the number of concurrencies, the throughput of the *high-throughput technology* and *high-performance technology* shows a linear growth, while that of the *virtualization technology* shows little change. When the number of concurrencies is 5, the throughput of the *virtualization technology* is 11.92 Gbit/s and that of the high-throughput technology is 39.7 Gbit/s. The throughput of the high-performance technology is 53.5 Gbit/s, which is 4.49 times that of the virtualization technology and 1.35 times that of the high-throughput technology.

In terms of the delay time, the three technologies are stable. The virtualization technology is stable at 0.9 ms, the high-throughput technology is stable at 0.7 ms, and the high-performance technology is stable at 0.46 ms. The delay time of the high-performance technology is reduced by 49% compared to that of the *virtualization technology* and reduced by 34% compared to that of the *high-throughput technology*.

In terms of the packet loss rate, *high-throughput technology* is stable at 0.4%. The *high-performance technology* will increase slowly as the number of concurrencies increases, but the *virtualization technology* always has a high packet loss rate.

Therefore, in the concurrent situation, *high-performance technology* performs well in terms of throughput and delay. However, because the *high-throughput technology* uses namespace [50], it can process packets more stably in the concurrent situation, and it has a lower packet loss rate than *high-performance technology*.

In conclusion, compared with the *virtualization technology* and *high-throughput technology*, *high-performance technology* has obvious performance advantages in multihop and concurrent situations.

6.5. Comparisons in Situations of Link Congestion. In the same environment, we compare the *virtualization technology*, *high-throughput technology*, and *high-performance technology* (with and without the load balancing of links). We design a testing topology, as shown in Figure 18. When C1 transmits many packets to C3 to block the communication link, we test the communication path and throughput of C2 and C4. The results are shown in Table 5 and Figure 19.

From Table 5 and Figure 19, we can see that, compared with the *virtualization technology*, *high-throughput technology*, and *high-performance technology* without the load balancing of links, the *high-performance technology* with the load balancing of links can select the relatively idle link as the forwarding path in the situation of congestion. Because the congestion link is selected for forwarding paths, the throughput of the *virtualization technology* is almost 0 Gbit/s, that of the *high-throughput technology* is 4.35 Gbit/s, and that of the *high-performance technology* without the load balancing of links is 9.8 Gbit/s. However, the throughput of the *high-performance technology* with the load balancing of links is 12.3 Gbit/s. Therefore, *high-performance technology* can provide a high-quality communication path in the situation of congestion through the load balancing of links.

6.6. Verification of Advantages of the Controller Cluster. In this section, we compare the average time for the single controller and the controller cluster (it contains three controllers) to process all packets. The shorter the time is, the more the routing nodes that can be controlled. We design a scenario as Figure 20 and use a single controller or a controller cluster to control all routing nodes.

In Figure 20, all the clients send packets to the controller of the emulation network, so the controller needs to process these packets and return them. Each client calculates the

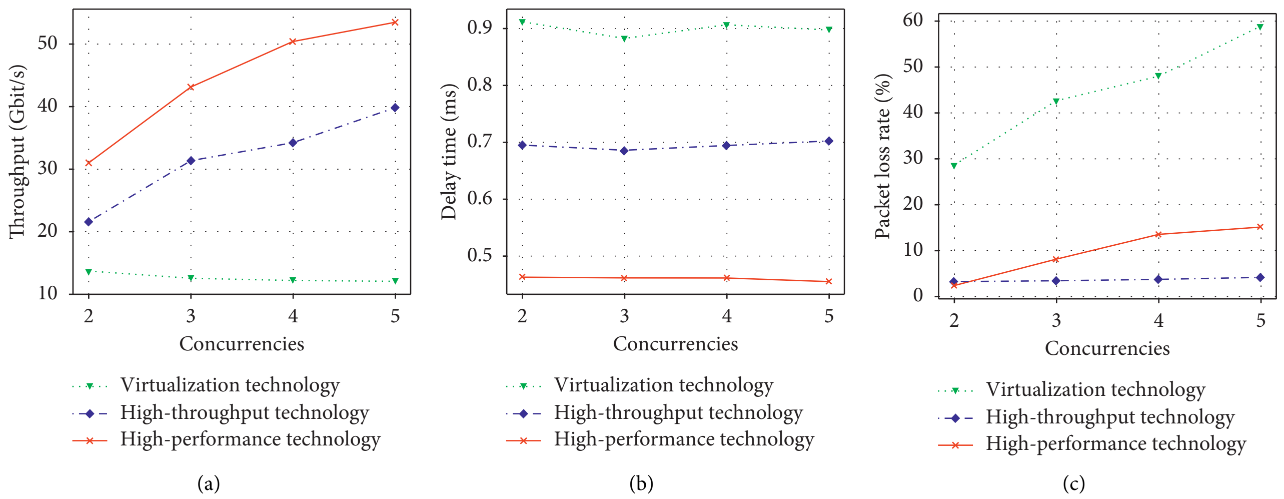


FIGURE 17: Comparison of performance in the concurrent situation.

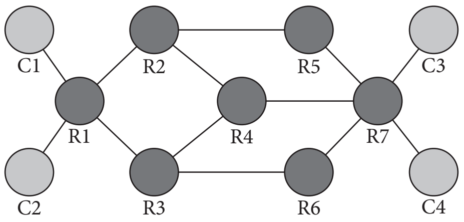


FIGURE 18: Topology for testing link load balancing.

TABLE 5: Route before and after congestion.

Routing emulation technology	The path of C1 to C3	The path of C2 to C4
Virtualization technology	[C1, R1, R2, R5, R7, C3]	[C2, R1, R2, R5, R7, C4]
High-throughput technology	[C1, R1, R2, R5, R7, C3]	[C2, R1, R2, R5, R7, C4]
Without the load balancing of links	[C1, R1, R2, R5, R7, C3]	[C2, R1, R2, R5, R7, C4]
With the load balancing of links	[C1, R1, R2, R5, R7, C3]	[C2, R1, R3, R6, R7, C4]

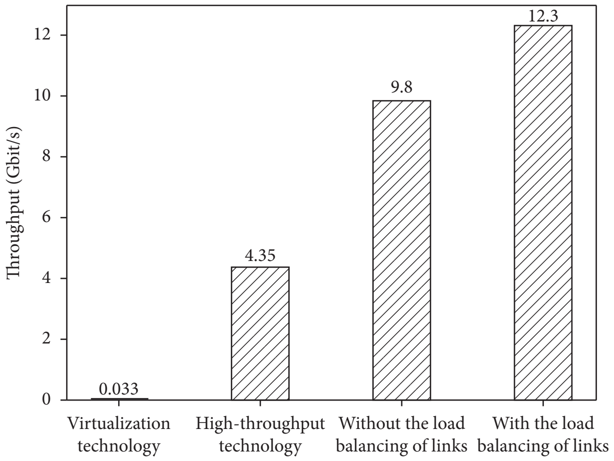


FIGURE 19: Comparison of throughput in the situation of congestion.

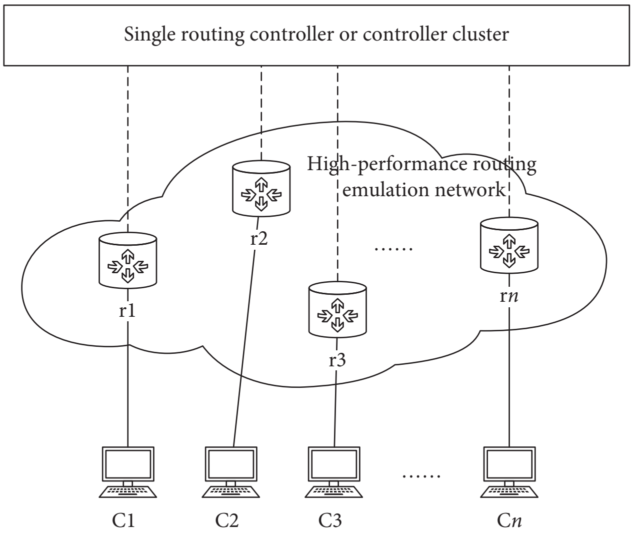

Single routing controller or controller cluster
High-performance routing emulation network
r1
r2
r3
rn
......
C1
C2
C3
Cn
......

Figure 20: Scenario for verifying advantages of the controller cluster.

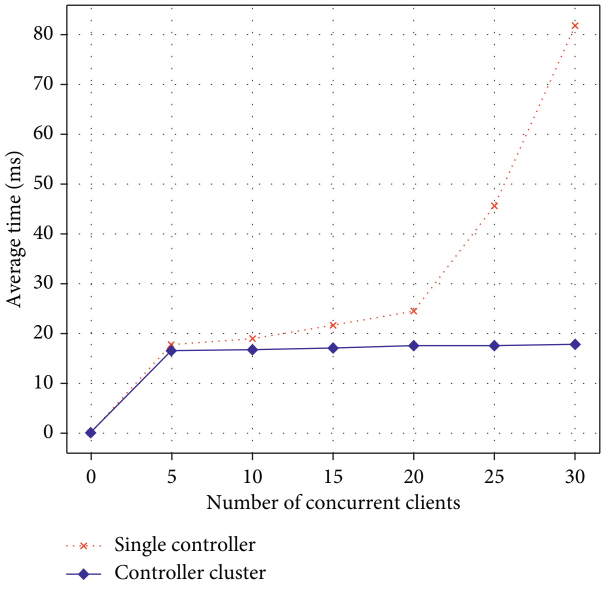


Figure 21: Comparison of processing time.

round-trip delay of the packets and obtains the average time T_{avg} by

$$T_i = \text{packet\_delay}_i - \text{link\_delay}_i, \quad (11)$$

$$T_{\text{avg}} = \frac{\sum_{i=1}^{N} T_i}{N}. \quad (12)$$

In (11), packet_delay_i represents the round-trip time of the packet sent by the ith client, and link_delay_i represents the link delay time from the ith client to the controller. The time for the controller to process the packet of the ith client can be obtained by subtracting link_delay_i from packet_delay_i. In (12), N represents the number of clients. Figure 21 shows the comparison of the value T_{avg} between a single controller and a controller cluster in the case of different numbers of concurrent clients.

As the number of concurrent clients increases, the time needed for the single controller to process packets increases dramatically. However, the time needed for the controller cluster to process packets increases slowly. Therefore, the controller cluster can control more routing nodes and improve the scalability of the emulation scale.

7. Conclusions

This paper proposes a high-performance routing emulation technology based on a cloud platform that provides a network environment for edge computing to verify and evaluate new architecture, protocol, and security technologies. First, we combine OpenStack and SDN technology to propose a high-performance routing emulation architecture. Then, we implement the routing function, OSPF protocol, and load balancing of links through apps of the decision layer. Finally, we propose a distributed control method and improve the scalability of the emulation scale with the controller cluster. Experiments show that, compared with other routing emulation technologies, this technology achieves less overhead, higher performance, and a realistic OSPF protocol. The controller cluster can also control more routing nodes than the single controller.

However, this paper studies only how to support the OSPF protocol, and other dynamic routing protocols (such as the Routing Information Protocol (RIP) and the Border Gateway Protocol (BGP)) need to be studied further. In addition, how to realize the hot backup of the controller will be the direction of the follow-up research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61672264 and 61972182) and National Key R&D Program of China (Grant no. 2016YFB0800801).

References

[1] D. Acarali, M. Rajarajan, N. Komninos, and B. B. Zarpelão, “Modelling the spread of botnet malware in IoT-based wireless sensor networks,” *Security and Communication Networks*, vol. 2019, Article ID 3745619, 13 pages, 2019.

[2] C. Zhou, A. Li, A. Hou et al., “Modeling methodology for early warning of chronic heart failure based on real medical big data,” *Expert Systems with Applications*, vol. 151, Article ID 113361, 2020.

[3] X. Xu, Q. Huang, X. Yin, M. Abbasi, M. R. Khosravi, and L. Qi, "Intelligent offloading for collaborative smart city services in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7919–7927, 2020.

[4] W. Zhong, X. Yin, X. Zhang et al., "Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment," *Computer Communications*, vol. 157, pp. 116–123, 2020.

[5] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[6] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, "Community-diversified influence maximization in social networks," *Information Systems*, vol. 92, pp. 1–12, 2020.

[7] L. Wang, X. Zhang, R. Wang, C. Yan, H. Kou, and L. Qi, "Diversified service recommendation with high accuracy and efficiency," *Knowledge-Based Systems*, vol. 204, Article ID 106196, 2020.

[8] L. Wang, X. Zhang, T. Wang et al., "Diversified and scalable service recommendation with accuracy guarantee," *IEEE Transactions on Computational Social Systems*, pp. 1–12, 2020.

[9] Y. Zeng, M. Chao, and R. Stoleru, "EmuEdge: a hybrid emulator for reproducible and realistic edge computing experiments," in *Proceedings of the 2019 IEEE International Conference on Fog Computing (ICFC)*, June 2019.

[10] A. A. T. R. Coutinho, F. Greve, and C. Prazeres, "An architecture for fog computing emulation," in *Proceedings of the Anais do XV Workshop em Clouds e Aplicações*, SBC, Belém, Brazil, May 2017.

[11] Y. Xing and Y. Zhan, "Virtualization and cloud computing," *Future Wireless Networks and Information Systems*, pp. 305–312, Springer, Berlin, Germany, 2012.

[12] R. L. S. De Oliveira, C. M. Schweitzer, A. A. Shinoda et al., "Using mininet for emulation and prototyping software-defined networks," in *Proceedings of the 2014 IEEE Colombian Conference on Communications and Computing (COLCOM)*, pp. 1–6, IEEE, Bogota, Colombia, June 2014.

[13] J. D. Beshay, A. Francini, and R. Prakash, "On the fidelity of single-machine network emulation in linux," in *Proceedings of the 2015 IEEE 23rd International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pp. 19–22, IEEE, Atlanta, GA, USA, October 2015.

[14] X. Xu, S. Fu, W. Li, F. Dai, H. Gao, and V. Chang, "Multi-objective data placement for workflow management in cloud infrastructure using NSGA-II," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, pp. 605–615, 2020.

[15] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2020.

[16] O. Sefraoui, M. Aissaoui, and M. Eleuldj, "OpenStack: toward an open-source solution for cloud computing," *International Journal of Computer Applications*, vol. 55, no. 3, pp. 38–42, 2012.

[17] R. Kumar, N. Gupta, S. Charu et al., "Open source solution for cloud computing platform using openstack," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 5, pp. 89–98, 2014.

[18] T. Rosado and J. Bernardino, "An overview of openstack architecture," in *Proceedings of the 18th International Database Engineering & Applications Symposium*, pp. 366-367, Porto, Portugal, July 2014.

[19] Z. Mengdong, J. Xin, and W. Xiaofeng, "Research on high-throughput routing simulation based on open-stack," *Computer Engineering and Applications*, vol. 54, no. 22, pp. 74–79, 2018.

[20] T. Hirt, *Kvm-the Kernel-Based Virtual Machine*, Red Hat Inc., Raleigh, NC, USA, 2010.

[21] Docker: storagedriver, https://docs.docker.com/storage/storagedriver/.

[22] P. Jakma and D. Lamparter, "Introduction to the quagga routing suite," *IEEE Network*, vol. 28, no. 2, pp. 42–48, 2014.

[23] M. Handley, O. Hodson, and E. Kohler, "XORP: an open platform for network research," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 53–57, 2003.

[24] J. Martins, M. Ahmed, C. Raiciu et al., "ClickOS and the art of network function virtualization," in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, USENIX Association, Seattle, WA, USA, April 2014.

[25] N. Varis, "Anatomy of a linux bridge," in *Proceedings of the Seminar on Network Protocols in Operating Systems*, p. 58, Espoo, Finland, September 2012.

[26] B. Pfaff, J. Pettit, T. Koponen et al., "The design and implementation of open vswitch," in *Proceedings of the 12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*, pp. 117–130, Oakland, CA, USA, May 2015.

[27] M. K. Shin, K. H. Nam, and H. J. Kim, "Software-defined networking (SDN): a reference architecture and open APIs," in *Proceedings of the 2012 International Conference on ICT Convergence (ICTC)*, pp. 360-361, IEEE, Jeju, South Korea, October 2012.

[28] P. S. Katkar and D. V. R. Ghorpade, "Comparative study of network simulator: NS2 and NS3," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 6, no. 3, pp. 608–612, 2016.

[29] X. Chang, "Network simulations with opnet," in *Proceedings of the 31st Conference on Winter Simulation: Simulation—A Bridge to the Future-Volume 1*, pp. 307–314, Phoenix, AZ, USA, December 1999.

[30] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: a library for parallel simulation of large-scale wireless networks," *ACM SIGSIM Simulation Digest*, vol. 28, no. 1, pp. 154–161, 1998.

[31] B. Heller, *Reproducible Network Research with High-Fidelity Emulation*, Stanford University, Stanford, CA, USA, 2013.

[32] O. Tkachova, M. J. Salim, and A. R. Yahya, "An analysis of SDN-openstack integration," in *Proceedings of the 2015 Second International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T)*, pp. 60–62, IEEE, Kharkiv, Ukraine, October 2015.

[33] J. Chen, H. Song, X. Wang et al., "Emulation of router functions based on a cloud platform," in *Proceedings of the 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 271–276, IEEE, Beijing, China, October 2019.

[34] M. H. Huang, Y. X. Zhang, and X. U. Fei, "Design of virtualization platform for router emulation," *Journal of System Simulation*, vol. 26, no. 8, pp. 1672–1677, 2014.

[35] N. Memari, S. J. B. Hashim, and K. B. Samsudin, "Towards virtual honeynet based on LXC virtualization," in *Proceedings of the 2014 IEEE Region 10 Symposium*, pp. 496–501, IEEE, Kuala Lumpur, Malaysia, April 2014.

[36] L. Lian, Y. Zhang, H. Zhang et al., "Constructing virtual network attack and defense platform based on openstack," in *Proceedings of the 2015 International Conference on*

Automation, Mechanical Control and Computational Engineering, Changsha, China, April 2015.

[37] D. Huang, B. He, and C. Miao, “A survey of resource management in multi-tier web applications,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1574–1590, 2014.

[38] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou, and Q. Ni, “Data-driven web APIs recommendation for building web applications,” *IEEE Transactions on Big Data*, p. 1, 2020.

[39] S. Y. Wang, H. W. Chiu, and C. L. Chou, “Comparisons of SDN OpenFlow controllers over EstiNet: Ryu vs. NOX,” in *Proceedings of the ICN 2015*, p. 256, San Francisco, CA, USA, September 2015.

[40] J. Moy, *OSPF Version 2*, 1998.

[41] M. Rostanski, K. Grochla, and A. Seman, “Evaluation of highly available and fault-tolerant middleware clustered architectures using RabbitMQ,” in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, pp. 879–884, IEEE, Warsaw, Poland, September 2014.

[42] Y. Deng, Y. Chen, Y. Zhang, and S. Mahadevan, “Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment,” *Applied Soft Computing*, vol. 12, no. 3, pp. 1231–1237, 2012.

[43] Z. Wang and J. Crowcroft, “Shortest path first with emergency exits,” in *Proceedings of the ACM Symposium on Communications Architectures & Protocols*, pp. 166–176, Philadelphia, PA, USA, September 1990.

[44] Open Networking Foundation, 2012, https://www.opennetworking.org/.

[45] C. Hunt, *TCP/IP Network Administration*, O’Reilly Media, Inc., Newton, MA, USA, 2002.

[46] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari et al., “Reverse traceroute,” in *Proceedings of the NSDI*, vol. 10, pp. 219–234, Boston, MA, USA, April 2010.

[47] B. Al-Musawi, P. Branch, M. F. Hassan, and S. R. Pokhrel, “Identifying OSPF LSA falsification attacks through non-linear analysis,” *Computer Networks*, vol. 167, Article ID 107031, 2020.

[48] A. Agusriandi and E. Elihami, “Developing delay jitter, throughput, and package lost IPERF3 for learning Islamic education,” *Jutkel: Jurnal Telekomunikasi, Kendali dan Listrik*, vol. 2, no. 1, pp. 23–30, 2020.

[49] C. Pelsser, L. Cittadini, S. Vissicchio et al., “From Paris to Tokyo: on the suitability of ping to measure latency,” in *Proceedings of the 2013 Conference on Internet Measurement Conference*, pp. 427–432, Barcelona, Spain, October 2013.

[50] E. W. Biederman and L. Networx, “Multiple instances of the global linux namespaces,”vol. 1, pp. 101–112, in *Proceedings of the Linux Symposium*, vol. 1, pp. 101–112, Citeseer, Ottawa, Canada, July 2006.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8881640, 11 pages
https://doi.org/10.1155/2020/8881640

Research Article

An Intelligent Real-Time Traffic Control Based on Mobile Edge Computing for Individual Private Environment

Sa Math,[1] **Lejun Zhang**,[2] **Seokhoon Kim**,[3] **and Intae Ryoo**[4]

[1] *Department of Software Convergence, Soonchunhyang University, Asan-si, Chungcheongnam-do 31538, Republic of Korea*
[2] *Department of Information Engineering, Yangzhou University, Yangzhou 225127, China*
[3] *Department of Computer Software Engineering, Soonchunhyang University, Asan-si, Chungcheongnam-do 31538, Republic of Korea*
[4] *Department of Computer Engineering, Kyung Hee University, Gwangju-si, 17104 Gyeonggi-Do, Republic of Korea*

Correspondence should be addressed to Seokhoon Kim; seokhoon@sch.ac.kr and Intae Ryoo; itryoo@khu.ac.kr

Received 10 August 2020; Revised 17 September 2020; Accepted 5 October 2020; Published 22 October 2020

Academic Editor: Xiaolong Xu

The existence of Mobile Edge Computing (MEC) provides a novel and great opportunity to enhance user quality of service (QoS) by enabling local communication. The 5th generation (5G) communication is consisting of massive connectivity at the Radio Access Network (RAN), where the tremendous user traffic will be generated and sent to fronthaul and backhaul gateways, respectively. Since fronthaul and backhaul gateways are commonly installed by using optical networks, the bottleneck network will occur when the incoming traffic exceeds the capacity of the gateways. To meet the requirement of real-time communication in terms of ultralow latency (ULL), these aforementioned issues have to be solved. In this paper, we proposed an intelligent real-time traffic control based on MEC to handle user traffic at both gateways. The method sliced the user traffic into four communication classes, including conversation, streaming, interactive, and background communication. And MEC server has been integrated into the gateway for caching the sliced traffic. Subsequently, the MEC server can handle each user traffic slice based on its QoS requirements. The evaluation results showed that the proposed scheme enhances the QoS and can outperform on the conventional approach in terms of delays, jitters, and throughputs. Based on the simulated results, the proposed scheme is suitable for improving time-sensitive communication including IoT sensor's data. The simulation results are validated through computer software simulation.

1. Introduction

The presence of the 5th generation (5G) communication network significantly aims to deliver an extremely fast communication speed, high reliability, and low end-to-end (E2E) delay in milliseconds (ms). Each base-station cell coverage provides local service within 100 meters, which attempts to offer strong connectivity, real-time (RT) communication, and Device-to-Device (D2D) and supports massive User Equipment (UE) connection as well [1, 2]. 5G provides huge bandwidth and ultra-low latency (ULL) ten times over 4G-LTE. Moreover, the 5G paradigm aims to support future user applications such as IoT traffic, Wireless Sensor Network (WSN) traffic, automotive transportation, and gaming traffic. With this support and these contributions, the huge traffic is generated from the Radio Access Network (RAN) devices and goes through to the Evolved Packet Core (EPC) gateways such as a Service Gateway (S-GW) and Packet Data Gateway (P-GW). Nevertheless, there is still limited capacity in the EPC area while it is a common optical network. Therefore, EPC architecture keeps the similarity to the previous communication system which possibly arises traffic congestion problems and insufficient resources in the EPC area. To reduce the outgoing traffic to the remote network, local clouds have been proposed to establish local communication [3]. Currently, Mobile Edge Computing (MEC), Network Slicing (NS), Software-Defined Network (SDN), and Network Function Virtualization

(NFV) are necessary technologies which have been proposed to overcome the aforementioned troubles and challenges in 5G communication, to improve the network performance, and to take benefits from cost reduction [4, 5]. Figure 1 illustrates the typically 5G end-to-end (E2E) communication system architecture. The bottleneck network area is located in the fronthaul and backhaul, whenever the incoming traffic from the variety of Radio Remote Heads (RRH) surpasses the serving capacity, the network congestion will occur. The ULL perspective is required for RT communication, so it is obliged to cope with the existing problems in the EPC gateways. Due to the fact that the fronthaul (S-GW) and backhaul gateway (P-GW) share the same network architecture, in this paper, the gateway refers to the S-GW or P-GW. MEC servers were integrated into the gateway to cache the traffic sliced before forwarding to the remote network.

2. Related Works

2.1. Real-Time Communication. The time-sensitive communications refer to video conference, mobile video streaming, game streaming, voice over Internet protocol (VoIP), and other RT traffic running over an unreliable network transport protocol called user datagram protocol (UDP). These communication types required ULL (delay and jitter) for Round-Trip Time (RTT) and required sufficient communication bandwidth. Daily communication traffic typically comprised two classes, namely, time-sensitive and time-insensitive. Time-sensitive (RT) classes are conversation and streaming; therefore, these communication types are less restricted on the packet error ratio during communication but require extremely high network status. The time-insensitive class consists of two communication types including interactive and background traffic. These communication types required extremely high communication reliability and very low packet error ratio, since it sends information through transmission control protocol (TCP), so the retransmission packet will occur whenever the destination missed the previous packets. However, there is less restriction on communication latency and bandwidth. In nowadays communication environments, both RT and nonreal-time (NRT) are sent through the same network environments with insufficient dynamic resources management to meet the QoS perspective for each traffic class. During the last several years, RT communications take benefits from local cloud services.

2.2. Mobile Edge Computing (MEC). The local clouds (i.e., MEC, fog computing, and cloudlet) has been released to enhance QoS for various network applications such as the Internet of Things (IoT), Heterogeneous Internet of Things (HetIoTs), gaming, and other applications, especially for RT applications [6–9]. The presence of MEC establishes the intelligence network in the edge area [10–12]. This technology claims to gain higher communication bandwidth and provides ULL for real-time communication. Meanwhile, MEC consists of challenging issues in capacity limitation, while it is required to offer heterogeneous services for massive users. Some applications with higher resource computation requirements are required to access remotely the MCC server. Moreover, privacy protection for the local cloud is required to be considered for safety communication and data integrity [13–16]. As shown in Figure 2, the caching method is enabled by MEC servers which synchronized with MCC servers, and the frequently requested contents or popularly used applications are targeted to be cached to MEC servers.

The caching methods are beneficial in latency reduction, gain a higher bandwidth, and save the resources at EPC for both user-plane and data-plane. Anyway, several challenges have been introduced in MEC employment, such as expanded RRH infrastructure, power consumption, resource management, and security problems [17]. Due to that a variety of user's information is stored in MEC in an edge network, the complicated security methods are required to enable trusted communications for edge networking [3, 8]. There exists a huge, especially, convergence of heterogeneous applications, services, and infrastructures in 5G edge networks, both physical and virtual. These network environments are not convenient to handle for both security and excellent network QoS [18, 19]. So, Network Slicing presents a novel opportunity to handle the issues by slicing the user applications into different groups. With machine learning algorithms combination, it is possible to facilitate Network Slicing in terms of classification of the complicated user information, applications, and devices [20, 21]. The user applications can be sliced by grouping the user applications which are sharing the same or similar resource requirement into the same group. The sliced applications are more convenient to control and provide flexible control and security configuration by the controller. Undoubtedly, Network Slicing is a key candidate to enhance future network QoS and network safety to meet the perspective of 5G technology [22, 23].

2.3. Software-Defined Network (SDN). SDN is a key adoption candidate to enable future networking driven to softwarization and intelligent networks. SDN provides a global view of network status and a completely programmable system at the control plane [24]. Also, SDN is a concept of decoupling a forwarding plane from the control plane. Plus, this separation gains more convenience in terms of flexibility and scalability, while the user-plane requires higher bandwidth and the control plane requires lower latency [25, 26]. The computing, routing, monitoring, scheduling, policy control, security, and load-balancing are performed by the SDN controller [27]. Not only can SDN, especially, be used to enhance the QoS for RT traffic, but also it can be used to enhance trusted communication based on blockchain [28]. The controller gathers information from the user-plane by the southbound interface and communication with the upper layer by the northbound interface. The communication interface between them is provided by the OpenFlow protocol [29]. Even though SDN could independently stand without other technologies getting involved, but the

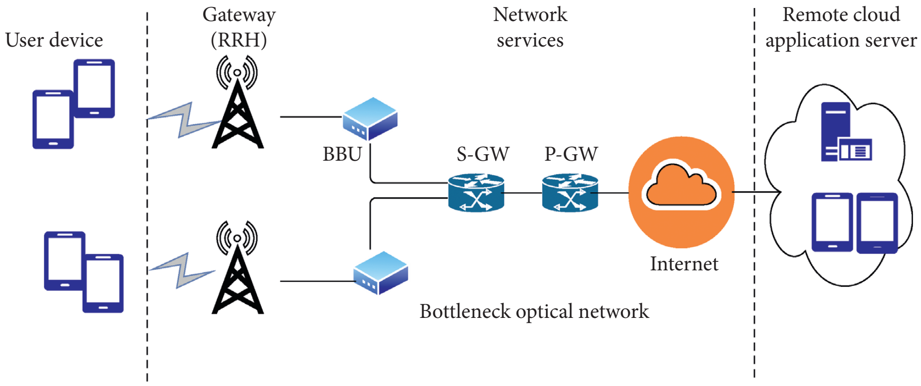


FIGURE 1: The typical future 5G communication architecture.

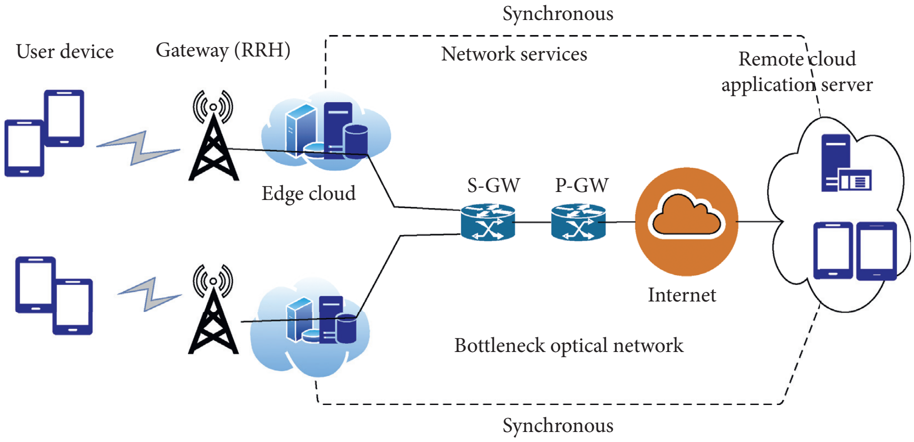


FIGURE 2: The typical future 5G communication architecture based on the MEC server.

integration of SDN and NFV presents a great opportunity to enhance virtualized computing in future network environments [30, 31]. This idea aims to provide virtualized resources to SDN entities and enables the controller to generate both physical and virtual resources. In the cloud systems, the converged SDN and NFV can be benefited in computing resources and dynamic resource configurations with a fault-tolerant technique [32, 33]. Based on this mention, the controller possibly generates the virtual controller and offloads from the physical to the virtualized for computing purposes.

2.4. The Proposed Intelligent Real-Time Traffic Control. The proposed method is to enhance the QoS for RT communications that can be caused by the limited resources at the backhaul gateway.

The proposed intelligent real-time traffic control handles the incoming traffic based on traffic classification and integration of the MEC server. Figure 3 shows the proposed network architecture by integrating the MEC server with the backhaul gateway. The MEC servers act as a caching server that buffers the incoming traffic, such as conversation, streaming, interactive, and background communication.

As formerly mentioned, the proposed scheme comprises three stages, namely, traffic classification, caching, and controlling the classified traffic. The following expressions are details about the three stages above.

2.5. Traffic Classification and MEC Caching. In this paper, Network Slicing is referred to as the splitting of user traffic as the 4 different slices such as slice 1 for conversation, slice 2 for streaming, slice 3 for interactive, and slice 4 for background communication. Therefore, the classification process was based on each traffic characteristic such as a packet error ratio (PER), protocol data unit (PDU) size, and other QoS parameters.

Subsequently, each slice of the traffic was cached to different MEC pools, and each MEC pool provides the buffer resources for queueing the incoming traffic to wait for serving. The traffic slicing can be made by employing the K-mean machine learning method as shown in Figure 4. Start with the determining number of groups $K = 4$ and then calculate the centroid. For the first time, the centroid has to select 4 different subsets as 4 classes randomly. In the next step, distance has to be calculated for each class. The distance can be calculated by using the Euclidean distance (ED) equation given as follows:

$$\mathrm{ED} = \sqrt{(x-a)^2 + (y-b)^2}. \quad (1)$$

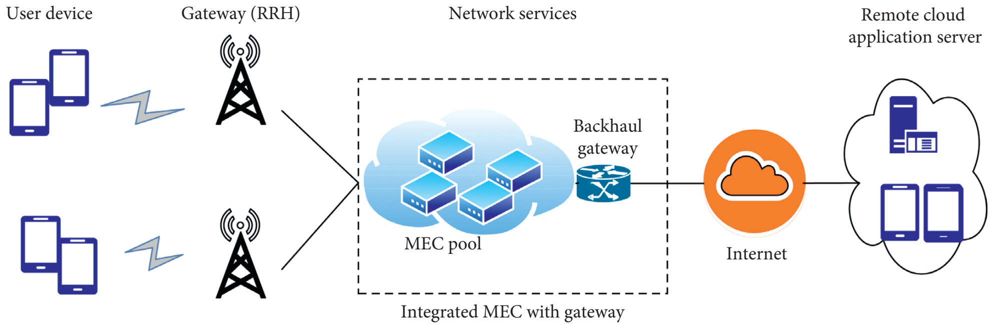


Figure 3: The proposed network architecture based on integrated MEC servers at the backhaul network.

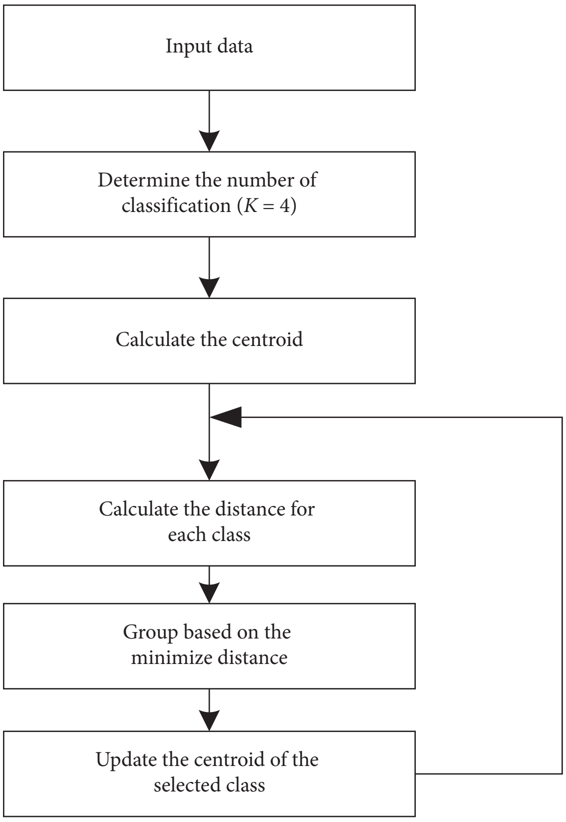

Input data
Determine the number of classification ($K = 4$)
Calculate the centroid
Calculate the distance for each class
Group based on the minimize distance
Update the centroid of the selected class

Figure 4: The diagram of traffic slice functions based on K-mean clustering flow.

The subset variables x, y, a, and b can present PDU size, PER, TCP, and UDP, respectively. As depicted in Figure 3, four MEC servers were integrated into the backhaul gateway to serve as buffers for the four traffic classes, so each traffic class has its individual MEC server. In this paper, the traffic classification was generated by computer software simulation. The conversation, streaming, interactive, and background traffic was generated based on its QoS parameters.

2.6. Management of the Classified Traffic. When the backhaul network becomes bad condition, it is required to serve real-time communication classes as the first priority. In the proposed scheme, the gateway has been configured to serve the cached traffic based on conditions of backhaul. The scheme will not be crucial to use when the backhaul gateway is considered as a normal status; otherwise, it is critical to employ the scheme during the backhaul gateway assumed as the congestion state.

The backhaul network can be defined as the M/M/1 queue model as follows:

$$\partial = \frac{\lambda_c + \lambda_s + \lambda_i + \lambda_b}{\mu}, \tag{2}$$

where ∂ is denoted as the ratio of incoming traffic and serving rate and also represents the status of backhaul. λ_c, λ_s, λ_i, and $\lambda_b$are denoted as the incoming rate of conversation, streaming, interactive, and background traffic, respectively. μ represents the serving rate of the backhaul gateway.

The gateway condition was referred to the user-plane status as the forwarding traffic based on the controller. The controller handles each slice of traffic from MEC pools. The backhaul status can be analyzed based on ∂, if $\partial \geq 1$ (the backhaul gateway resources are sufficient to handle for incoming traffic), and the status can be assumed as a normal condition. During the backhaul condition assumed as natural status (normal), the serving rule has been configured as default. The default rule handles the incoming traffic based on first come first serve (FCFS). So, the serving resources and rules are equalized for any incoming traffic classes. Thus, the real-time traffic will be dropped and low QoS will increase the waiting period in the MEC server. In another scenario, if $\partial \geq 1$ (this means that the serving resource of backhaul gateway μ is less than the incoming rate of user traffic λ), the network congestion in the system will occur, so the priority control of each traffic class has to be considered, as shown in Figure 5. RT traffic classes have to be considered as primary control rather than NRT. The scheme increases the communication rating of the RT traffic classes and reduces the communication rating of NRT traffic classes based on the increasing rate ratio of RT. The proposed scheme considers/classifies the cached traffic by four different classes as shown in Figure 6:

(i) Conversation (RT) has been configured as a first primary class and the serving resources have to be increased more than the other communication classes

```
while (communication ← true)
    monitoring ← true
        if ($\partial \leq 1$)
            IsConversation ← true
            if (state ← state 0)
                state ← state 1
            IsStreaming ← true
            else if (state ← state 0)
                state ← state 3
            IsInteractive ← true
            else if (state ← state 0)
                state ← state 4
            IsBackground ← true
            else if (state ← state 0)
                state ← state 2
            end if
        end if
    end while
```

FIGURE 5: The handling traffic based on the proposed scheme when the backhaul is considered as the bad statuses.

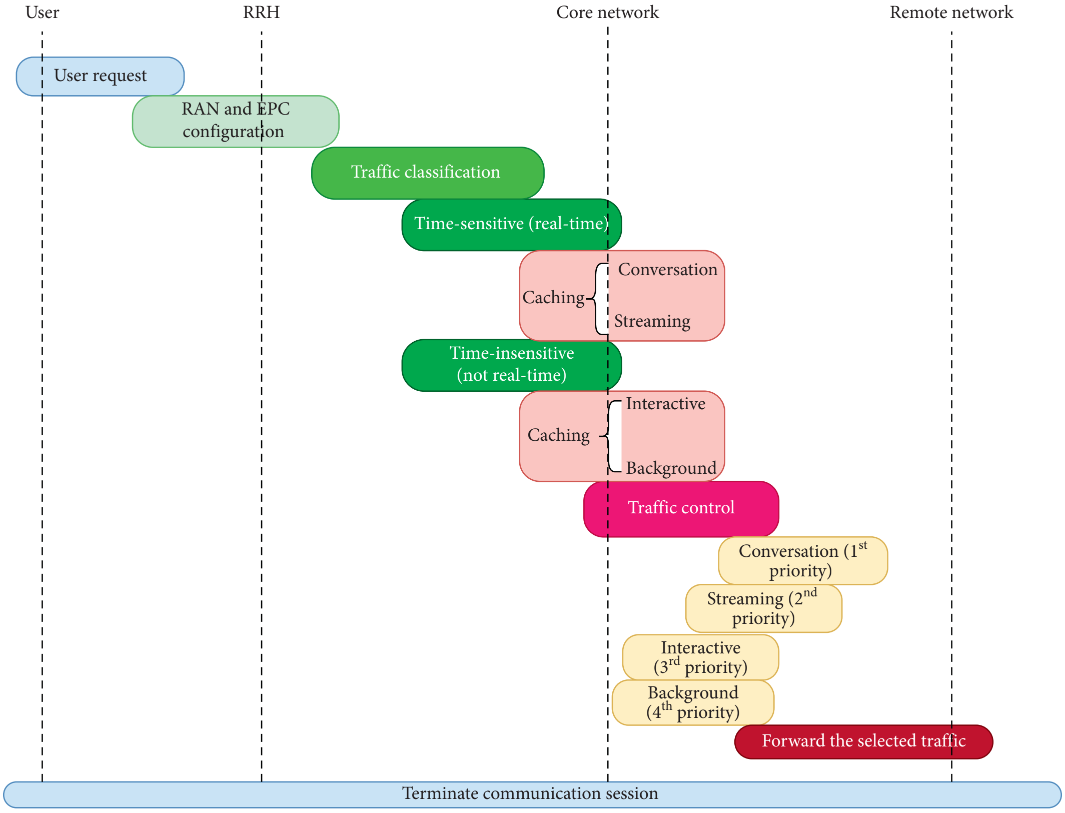

User
RRH
Core network
Remote network
User request
RAN and EPC configuration
Traffic classification
Time-sensitive (real-time)
Caching
Conversation
Streaming
Time-insensitive (not real-time)
Caching
Interactive
Background
Traffic control
Conversation (1st priority)
Streaming (2nd priority)
Interactive (3rd priority)
Background (4th priority)
Forward the selected traffic
Terminate communication session

FIGURE 6: The proposed scheme for RT handling based on MEC caching.

(ii) Streaming (RT) has been configured as a second primary class and the backhaul resources will be increased to greater than the interactive and background communication but lower than conversation

(iii) Interactive (NRT) has been configured as a third priority class and the serving resources will be decreased to lower than conversation and streaming but greater than background communication

(iv) Background (NRT) communication has been configured as the fourth priority and the backhaul resources are limited to lower than other communication classes

The increasing rates of conversation class based on the reducing rate of background communication class and the increasing rate of streaming class are based on the reducing rate of interactive communication class as shown in Table 1.

Moreover, the RT traffic in the MEC servers will be keeping adjusted similar to the normal network status, because, during the network limitation of backhaul resources, the algorithm is restricted to serving resources for NRT. In this scenario, NRT user traffic will be queued in longer periods, due to the fact that some of the NRT resources will be used for RT traffic. The scheme limits the NRT resources until the backhaul gateway becomes a normal status $\partial \geq 1$ and will configure the serving scheme to handle without restriction, as depicted in Figure 7.

2.7. Performance Evaluation

2.7.1. Analysis. The E2E latency occurs during packet transmission in the 5G communication system and can be written as T as follows:

$$T = T_{\text{RAN}} + T_{\text{Backhaul}} + T_{\text{Core}} + T_{\text{Transport}}, \quad (3)$$

where

(i) T_{RAN} is the latency of packet transmission from UEs to eNB. This latency is mainly from the physical and data-link layer, such as time of negotiation, channel coding, modulation, cyclic redundancy check, and other duties consisted in the physical and data-link layer.

(ii) T_{Backhaul} is the latency of the packet transmitted from eNB to the backhaul. The common connection between eNB and Core can be fiber optic or microwave link. The latency of the switching process can occur at the SGW.

(iii) T_{Core} is the time of building the connection to the core gateway. The latency can be contributed by both the control plane and the user-plane. The control plane consists of the latency of various EPC entities such as Mobile Management Entity (MME), Home Subscriber Server (HSS), Policy and Charging Rule Function (PCRF), and the SDN controller.

(iv) $T_{\text{Transport}}$ is the time taken by data transmission to the remote network; the latency is depending on the distance, link bandwidth, routing, and switching protocol:

$$T_{\text{RAN}} = t_0 + t_{\text{FA}} + t_{TX} + t_{bsp} + t_{mpt}, \quad (4)$$

where

(i) t_Q is the waiting time of incoming traffic depending on λ, if $\lambda \geq \mu$ then t_Q is increased.

(ii) t_{FA} is the latency that occurred by frame alignment.

(iii) t_{TX} is the latency of transmission depending on radio channel condition, payload size, and transport protocol.

(iv) t_{bsp} is the latency at the eNB.

(v) t_{mpt} is the delay of at UEs and eNBs terminal; it is depending on the capacity of both terminals:

$$\begin{aligned} T_{\text{Backhaul}} &= t_Q + t_E + t_{TX} + t_S, \\ T_{\text{Core}} &= t_Q + t_E + t_{TX} + t_{epc} + t_{SR}, \end{aligned} \quad (5)$$

where

(i) t_E is the time delay of the circuit through network devices.

(ii) t_S is represented as the switching delay.

(iii) t_{epc} is represented as the delay between communication interfaces of EPC entities such as MME, HSS, and PCRF. The communication delay between EPC entities will take a few microseconds.

(iv) t_{SR} is represented the latency of the switching and routing periods.

MEC pool can be modeled as m/m/1 queue model. So, the average waiting time of user traffic is denoted as t_Q, where

$$t_Q = \frac{\lambda}{\mu(\mu - \lambda)}. \quad (6)$$

Then, the Round-Trip Time (RRT) of E2E delay is defined as $2 \times T$ approximately. The E2E delay and latency occurring in the communication system are well discussed in [34].

In the communication environments, delay D_t (at any time t) can occur constantly and vary based on the network statuses. So, the variance delays occur during the communication called jitters ΔJ, since the jitters at the time t are denoted as ΔJ_t and can be calculated as the following equation:

$$\Delta J_t = \sqrt{\left(D_t - D_{t-1}\right)^2}, \qquad \Delta J_t \geq 0. \quad (7)$$

According to equation (7) , the communication jitters of the system at time t are denoted as $\Delta J(\text{syst})_t$ and can be formed as

$$\Delta J(\text{syst})_t = \sum_t^n \Delta J_t. \quad (8)$$

Based on equation (8), the average jitters of the four traffic classes in the system can be modeled as $\Delta J(c, s, i, b)_t$, where

Table 1: The configuration states for RT and NRT communication classes.

State	Rating control
State 0	None control (100%)
State 1	Increase rating (70%) for communication
State 2	Reduce rating (70%) for background
State 3	Increase rating (60%) for streaming
State 4	Reduce rating (60%) for interactive

```
while (communication ← true)
    monitoring ← true
        if ($\partial \geq 1$)
            IsConversation ← true
            if (state ← state 1)
                state ← state 0
            IsStreaming ← true
            else if (state ← state 3)
                state ← state 0
            IsInteractive ← true
            else if (state ← state 4)
                state ← state 0
            IsBackground ← true
            else if (state ← state 2)
                state ← state 0
            end if
        end if
    end while
```

Figure 7: The default handling incoming traffic at the backhaul gateway.

$$\Delta J(c,s,i,b)_t = \sum_{t=1}^{n} \frac{\Delta J(c)_t + \Delta J(s)_t + \Delta J(i)_t + \Delta J(b)_t}{4}. \tag{9}$$

$\Delta J(c)_t \geq 0$; $\Delta J(s)_t \geq 0$; $\Delta J(i)_t \geq 0$; and $\Delta J(b)_t \geq 0$, $t = 1, 2, 3, 4, \ldots, n$, where $\Delta J(c)_t$, $\Delta J(s)_t$, $\Delta J(i)_t$, and $\Delta J(b)_t$ are the average jitters of conversation, streaming, interactive, and background, respectively.

The communication delays of the system at any time t are conveyed by $D(\text{syst})_t$ and can be formed as

$$D(\text{syst})_t = \sum_{t=1}^{n} \frac{D_t}{n}, \quad n \geq 0. \tag{10}$$

Corresponding to equation (10), the average delay of the system with four traffic classes can be determined as $D(c,s,i,b)_t$, where

$$D(c,s,i,b)_t = \sum_{t=1}^{n} \frac{D(c)_t + D(s)_t + D(i)_t + D(b)_t}{4}, \tag{11}$$

$D(c)_t$, $D(s)_t$, $D(i)_t$, and $D(b)_t$, are the average delays of conversation, streaming, interactive, and background, respectively.

2.7.2. Simulation Environments. The experiment was conducted by using a computer simulation program named network simulation version 3 (NS3) that was implemented by using the C++ library. The simulation topology was composed of RAN area, fronthaul, and backhaul gateway. The RED-queue disc has been used to buffer the incoming traffic to represent the MEC server. The total simulation packets are 1025148, conversation packets are 458226, streaming packets are 532598, interactive packets are 22605, and background packets are 11719. Due to the communication link interval that was configured to 10 milliseconds, simulation times for each traffic class are 600 seconds. There are 8 user devices used for simulation, and the distances were configured with 15 meters between each other.

Figure 8 illustrates that the simulation stages were conducted for experimentation, such as initialization which initializes the state for simulation of conversation, streaming, interactive, and background traffic and will be generated for periodic communication. NCON is referred to as the configuration of network condition, and TCON is the handling of incoming traffic based on the proposed scheme. And finally, it is the collection of the simulated results.

3. Experiment Results and Discussion

In this paper, the system evaluations are based on the comparison between the proposed approach and the conventional approach. Evaluations are regarding the average E2E delays, average E2E jitters, and average throughputs of the individual of each RT communication (conversation and streaming) and total average values integrating RT and NRT in the communication system.

Figure 9 shows the comparative average delays of the proposed approach with the conventional approach. The revaluation results are related to the analysis in equation (11) and the average delays are compared by integrating each average delay value of the four communication classes $D(c)_t$, $D(s)_t$, $D(i)_t$, and $D(b)_t$, respectively. The graph shows that the average delays of the proposed approach are lower than the conventional approach. Referring to the graph, the average value of the proposed approach is mostly 0.01361326 seconds, while the average delay of the conventional approach is mostly 0.013676041 seconds. For the RT communication system, E2E delays have to be ultra low to perform the great QoS for each user. Typically, the backhaul traffic will be reduced rapidly during the increasing of forwarding rate of the RT traffic. The proposed approach can reduce the number of traffic queues in the MEC server and possibly reserve or reduce the MEC resources. With the possibility of the higher forwarding rate at the backhaul, the buffer resources will not be required and lessen the computing resources of network devices.

Figure 10 shows the comparison of average throughputs of the system between the proposed and conventional approaches. Based on the showing graphs, the proposed approach has higher communication throughputs than the conventional. The average throughputs are relying on the average E2E delays and PDU sizes. The throughputs can be calculated by division of PDU size with communication delays. In this paper, the PDU size was configured as constant; thus, the throughput will be varied based on communication delays. The evaluation was conducted by

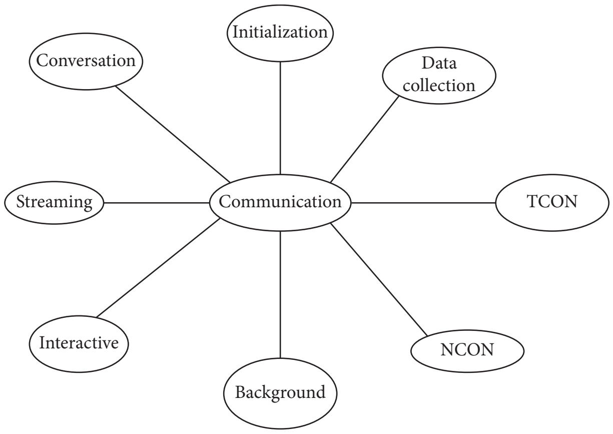


Figure 8: The simulation diagram.

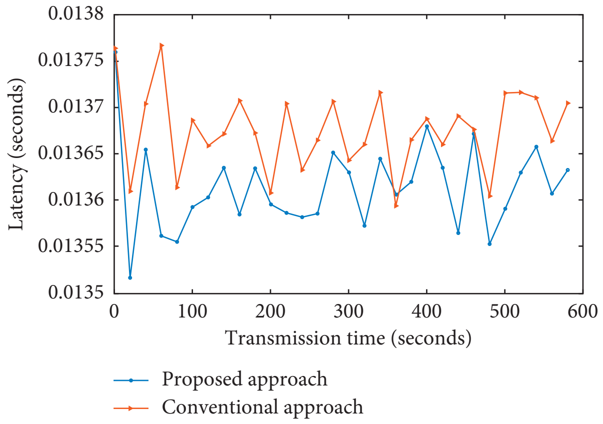


Figure 9: The comparison of average delays with integrating both RT and NRT.

calculating the average throughputs of conversation, streaming, interactive, and background communication and sum as a total average. The average E2E delays of the proposed approach are lower than the conventional, as shown in Figure 9.

The proposed scheme enhances the higher communication capacity of forwarding incoming traffic. So, the heavy user traffic in the backhaul gateway and bad statuses can be reduced. Moreover, this proposed approach is suitable to use for handling massive 5G user traffic, as well as enhancing QoS for RT communication classes.

Figure 11 shows that communication jitters of the proposed approach are lower if compared to the jitters of the conventional approach. The jitter evaluation is based on equation (9). The average jitters are compared based on each average of communication class, including average jitters of conversation, streaming, interactive, and background denoted as $\Delta J(c)_t$, $\Delta J(s)_t$ $\Delta J(i)_t$, and $\Delta J(b)_t$, respectively. Lower jitters are indicating the communication stability of the network. The communication jitter will occur when the backhaul network becomes congested. Consequently, the serving interval at gateway will be varied based on the obvious situations; when the network fluctuation of the serving times are higher, the communication QoS will be decreased concurrently. In the RT communication, especially, ultralow communication jitters are required. The proposed approach provides ultra-low jitters in communication systems, so the E2E communication jitter will be consistent.

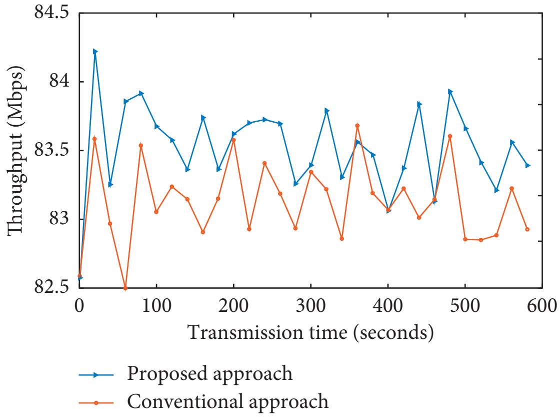


Figure 10: The average communication throughputs of the system.

The comparison of E2E communication jitters of conversation and streaming is presented in Figures 12(a) and 12(b), respectively. Figures 12(a) and 12(b) illustrate that the E2E jitters of the proposed approach outperform the conventional for both conversation and streaming traffic classes. The jitter evaluation was analyzed based on equation (8). Based on the graphs in Figure 12(a), the jitters of conversation traffic class of the proposed scheme have been improved, because the proposed scheme is restricted on the time-insensitive user traffic serving rate and increased the serving rate for the conversation traffic class. Thus, the communication stability can increase. The streaming jitters are shown in Figure 12(b). The graphs show that the E2E jitters of streaming traffic class of the proposed scheme have been improved, while there are higher jitters occurring in the conventional approach. Based on the graphs, the proposed scheme is significant to control the serving resources to enhance the quality of services for time-sensitive communications.

The E2E delays comparison between the proposed and conventional schemes of conversation and streaming communication is exhibited in Figures 13(a) and 13(b), respectively. The evaluation graphs in both Figures 13(a) and 13(b) were analyzed based on equation (10), in the above section. As shown in the evaluation graphs, the proposed approach has lower communication delays for both conversation and streaming, while the conventional approach has higher communication delays for both conversation and streaming communications. Due to the scheme being targeted for RT classes, the serving rate of NRT classes has been restricted and increased the serving rate of RT classes. So, the waiting time of NRT classes will be increased while the waiting time of RT will be reduced. However, the network performance of time-insensitive traffic does not rely on communication times.

Figures 14(a) and 14(b) show the comparison of average throughputs between the proposed and conventional

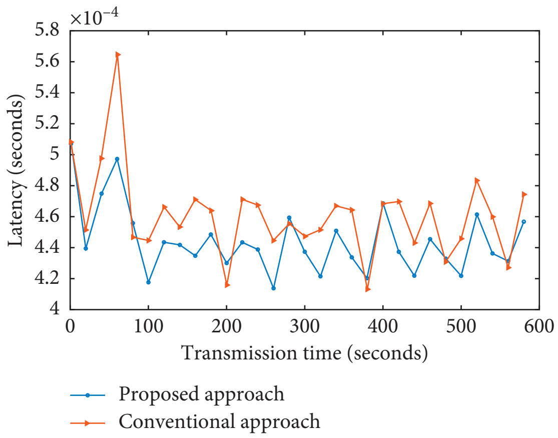
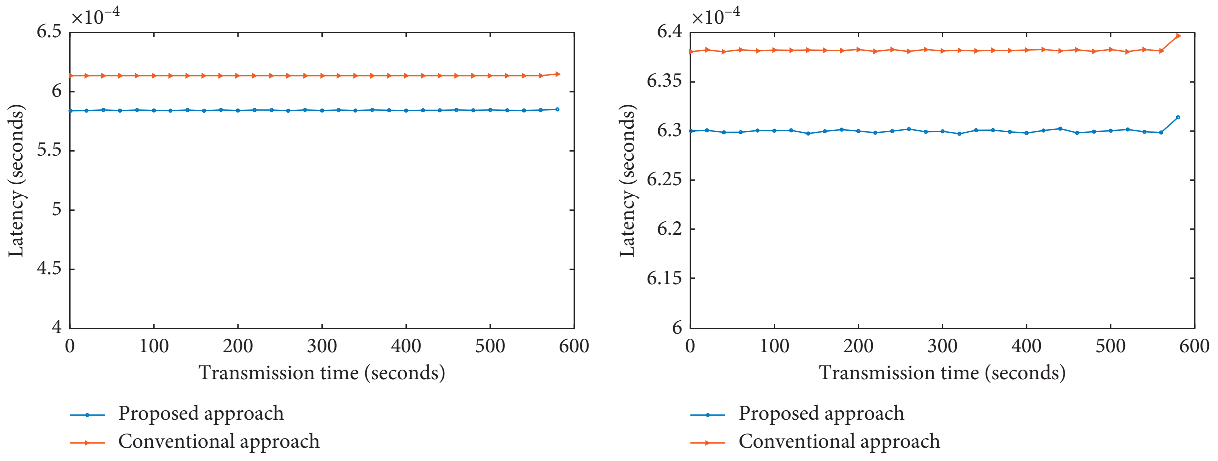


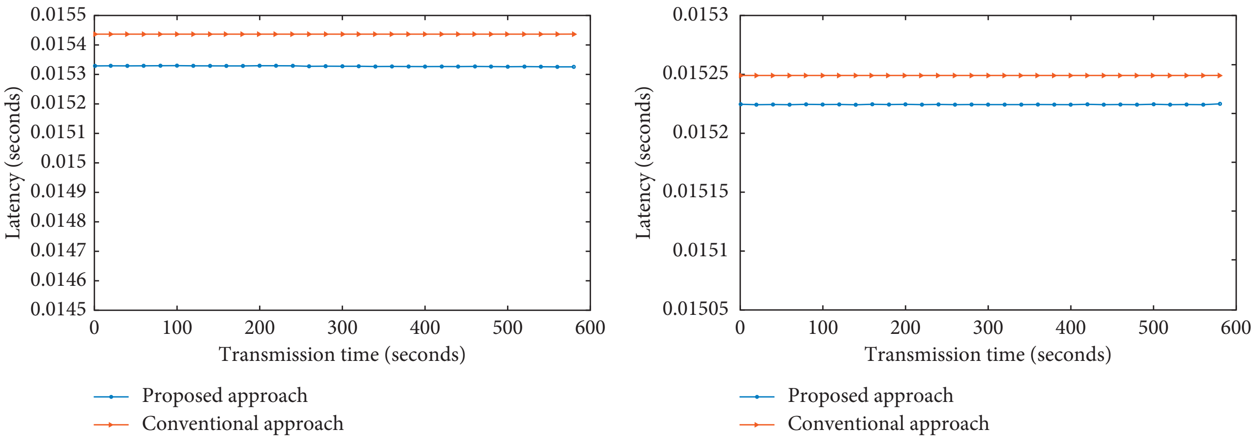
Figure 11: The comparison of average E2E communication jitters.



(a)

(b)

Figure 12: The comparison of average E2E communication jitters between the proposed and conventional approach of conversation (a) and streaming (b).



(a)

(b)

Figure 13: The comparison of average E2E communication delays between the proposed and conventional approach of conversation (a) and streaming (b) communication.

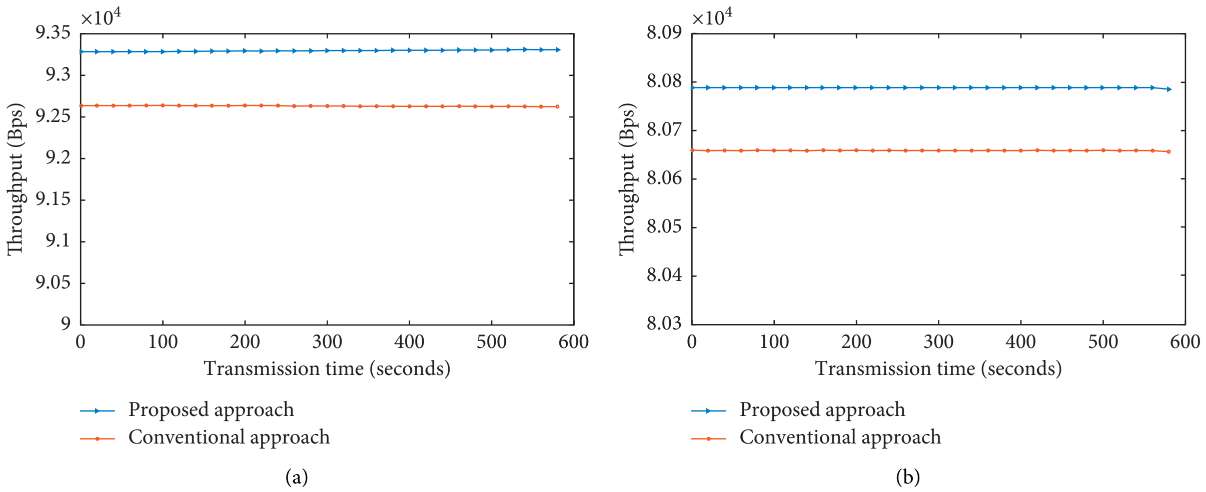


Figure 14: The comparison of average E2E communication throughputs between the proposed and conventional approach of conversation (a) and streaming (b) communication.

approaches for conversation and streaming communication. The graph shows that the proposed approach has higher communication throughputs for both conversation and streaming, while the throughputs of the conventional approach are lower. Because the E2E delays have been reduced in the proposed scheme, as shown in Figures 13(a) and 13(b) above, communication throughputs also improve relying on communication delays. Based on these evaluations, the proposed scheme enhances the communication throughputs for RT communication classes. According to the results in Figures 12–14, it is shown that the proposed scheme can be used to improve the network performance for RT communications. This proposed scheme, especially, meets the RT QoS perspectives and is suitable to be applied in bottleneck 5G backhaul gateway.

4. Conclusions

The 5G backhaul gateway consists of massive incoming traffic from heterogeneous devices with a variety of communication traffic. Thus, it is necessary to handle the communication traffic based on each traffic class, especially for RT communication that required ultra-low latency and higher communication rates more than the NRT traffic classes. The proposed approach handles the incoming traffic based on classifying the user traffic into four different classes, including conversation, streaming, interactive, and background communication. The MEC servers have been used to integrate with the backhaul gateway to buffer each of the traffic classes individually. Each communication class has an individual MEC server. When the backhaul is considered as a bad status, the proposed approach will be used to handle by giving more communication rates to RT (conversation and streaming) and reducing the communication rates of NRT based on the increasing ratio of the RT communication. Based on the simulation results, the proposed approach enhances the QoS over the conventional approach for RT communication in terms of reducing jitters, delays, and enhancing higher communication throughputs. This approach is suitable for enhancing QoS for RT in bottleneck 5G backhaul network environments and the privacy protection for each communication class based on Network Slicing. Finally, for further research, we aim to integrate more effective methods to enhance the massive user traffic in the bottleneck area.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by Korea Electric Power Corporation (Grant no. R18XA02), and this work was supported by the Soonchunhyang University Research Fund.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: the communication perspective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[2] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, “A survey on low latency towards 5G: RAN, core network and caching solutions,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3098–3130, 2018.

[3] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, and L. Qi, “Trust-oriented IoT service placement for smart cities in edge computing,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4084–4091, 2020.

[4] C. Xue, C. Lin, and J. Hu, “Scalability analysis of request scheduling in cloud computing,” *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 249–261, 2019.

[5] D. Shen, J. Luo, F. Dong, and J. Zhang, "VirtCo: joint coflow scheduling and virtual machine placement in cloud data centers," *Tsinghua Science and Technology*, vol. 24, no. 5, pp. 630–644, 2019.

[6] L. Liu, X. Chen, Z. Lu, L. Wang, and X. Wen, "Mobile-edge computing framework with data compression for wireless network in energy Internet," *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 271–280, 2019.

[7] L. Qi, W. Dou, W. Wang, G. Li, H. Yu, and S. Wan, "Dynamic mobile crowdsourcing selection for electricity load forecasting," *IEEE Access*, vol. 6, pp. 46926–46937, 2018.

[8] Y. Xu, L. Qi, W. Dou, and J. Yu, "Privacy-preserving and scalable service recommendation based on SimHash in a distributed cloud environment," *Complexity*, vol. 2017, Article ID 3437854, 9 pages, 2017.

[9] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2616–2624, 2017.

[10] X. Xu, H. Cao, Q. Geng, X. Liu, F. Dai, and C. Wang, "Dynamic resource provisioning for workflow scheduling under uncertainty in edge computing environment," *Concurrency and Computation: Practice and Experience*, 2020.

[11] L. M. Pham and T. Nguyen, "Flexible deployment of component-based distributed applications on the cloud and beyond," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 3, pp. 1141–1163, 2019.

[12] J. W. Jang, S. Kwon, S. Kim, J. Seo, J. Oh, and K. Lee, "Cybersecurity framework for IIoT-based power system connected to microgrid," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 5, pp. 2221–2235, 2020.

[13] H. Liu, H. Kou, C. Yan, and L. Qi, "Link prediction in paper citation network to construct paper correlation graph," *EURASIP Journal on Wireless Communications and Networking*, vol. 1, 2019.

[14] X. Chi, C. Yan, H. Wang, W. Rafique, and L. Qi, "Amplified LSH-based recommender systems with privacy protection," *Concurrency and Computation: Practice and Experience*, 2020.

[15] L. Liu, Y. Du, and Q. Fan, "A constrained multi-objective computation offloading algorithm in the mobile cloud computing environment," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 9, pp. 4329–4348, 2019.

[16] W. Tang, B. Qin, Y. Li, and Q. Wu, "Functional privacy-preserving outsourcing scheme with computation verifiability in fog computing," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 1, pp. 281–298, 2020.

[17] A. Rasheed, P. H. J. Chong, I. W. Ho, X. J. Li, and W. Liu, "An overview of mobile edge computing: architecture," technology and direction," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 10, pp. 4849–4864, 2019.

[18] L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, and J. Chen, "A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment," *Future Generation Computer Systems*, vol. 88, pp. 636–643, 2018.

[19] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, "Joint optimization of offloading utility and privacy for edge computing enabled IoT," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2622–2629, 2020.

[20] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: a systematic survey," in," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1988–2014, 2019.

[21] J. Xie, F. Richard Yu, T. Huang et al., "A survey of machine learning techniques applied to software defined networking (SDN): research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 393–430, 2019.

[22] M. Finsterbusch, C. Richter, E. Rocha, J. Muller, and K. Hanssgen, "A survey of payload-based traffic classification approaches," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1135–1156, 2014.

[23] L. Velasco, L. Gifre, J.-L. Izquierdo-Zaragoza et al., "An architecture to support autonomic slice networking," *Journal of Lightwave Technology*, vol. 36, no. 1, pp. 135–141, 2018.

[24] F. A. Lopes, M. Santos, R. Fidalgo, and S. Fernandes, "A software engineering perspective on SDN programmability," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1255–1272, 2016.

[25] C. Song, M. Zhang, Y. Zhan et al., "Hierarchical edge cloud enabling network slicing for 5G optical fronthaul," *Journal of Optical Communications and Networking*, vol. 11, no. 4, pp. B60–B70, 2019.

[26] Z. Zaidi, V. Friderikos, Z. Yousaf, S. Fletcher, M. Dohler, and H. Aghvami, "Will SDN be part of 5G?" *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3220–3258, 2018.

[27] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3075849, 8 pages, 2018.

[28] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2020.

[29] J. Li, X. Shen, L. Chen, J. Ou, L. Wosinska, and J. Chen, "Delay-aware bandwidth slicing for service migration in mobile backhaul networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, no. 4, pp. B1–B9, 2019.

[30] D. A. Chekired, M. A. Togou, L. Khoukhi, and A. Ksentini, "5G-Slicing-enabled scalable SDN core network: toward an ultra-low latency of autonomous driving service," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1769–1782, 2019.

[31] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[32] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2020.

[33] X. Li, C. Guo, L. Gupta, and R. Jain, "Efficient and secure 5G core network slice provisioning based on VIKOR approach," *IEEE Access*, vol. 7, pp. 150517–150529, 2019.

[34] S. Kim and W. Na, "Safe data transmission architecture based on cloud for Internet of things," *Wireless Personal Communications*, vol. 86, no. 1, pp. 287–300, 2015.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8822985, 9 pages
https://doi.org/10.1155/2020/8822985

Research Article

Routing Strategy for LEO Satellite Networks Based on Membership Degree Functions

Jian Zhou,[1,2] Qian Bo,[1,2] Lijuan Sun,[1,2] Juan Wang,[1,2] and Xiaoyong Yan[1,2]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing University of Posts and Telecommunications, Nanjing 210023, China*

Correspondence should be addressed to Jian Zhou; zhoujian@njupt.edu.cn and Lijuan Sun; lucifinil919@126.com

Received 3 August 2020; Revised 10 September 2020; Accepted 5 October 2020; Published 16 October 2020

Academic Editor: Xiaolong Xu

The deployment of Mobile Edge Computing (MEC) servers on Low Earth Orbit (LEO) satellites to form MEC satellites is of increasing concern. A routing strategy is the key technology in MEC satellites. To solve the uncertainty problem of LEO satellite link information caused by complex space environments, a routing strategy for LEO satellite networks based on membership degree functions is proposed. First, a routing model based on uncertain link information is established. In particular, the membership function is designed to describe the uncertain link information. Based on this, the comprehensive evaluation of the path is calculated, and the routing model considering uncertainty is established with the comprehensive evaluation of the path as the optimization objective. Second, in order to quickly calculate the path, a grey wolf optimization algorithm is designed to solve the routing model. Finally, simulation results show that the proposed strategy can achieve efficient and secure routing in complex space environments and improve the overall performance compared with the performances of traditional routing strategies.

1. Introduction

Mobile Edge Computing (MEC), which is a novel and powerful paradigm, is a promising alternative for providing computing capabilities at the edges of networks [1–3]. Low Earth Orbit (LEO) satellite networks possess the advantages of near real-time, wide coverage, and anti-destructive properties [4, 5]. Recently, some studies have combined LEO satellite networks with MEC to deploy MEC servers on LEO satellites for reducing delays and more general purposes [6–8]. As a new MEC architecture, MEC satellites are an emerging topic [9–11]. The design of communication protocols in MEC satellites for achieving efficient and secure data transmission is currently a research hotspot.

As the core of the communication protocol, the routing strategy for satellite networks is responsible for data transmission between the intersatellite links and for determining the overall performance of satellite networks [12]. Compared with terrestrial networks, satellite networks possess dynamic topological structures and unbalanced data traffic, which makes the routing strategy for terrestrial networks unsuitable for satellite networks. Thus, the routing strategy for satellite networks has been specially researched [13]. The existing research has focused on the high dynamic change issue of satellite networks. Based on the predictability, periodicity, regularity, and other features of satellite networks, routing strategies based on virtual topology [14–16], virtual nodes [17, 18], and coverage domain partitioning [19] were proposed. At the same time, the data traffic imbalance problem was also of high concern, and a routing strategy based on the load balance was proposed [20–22].

However, the link information of LEO satellite networks has a certain degree of uncertainty. On the one hand, the complexity of LEO satellite networks and operating space environments leads to uncertainty in the measured values of link information [23]. For example, harsh space environments such as vacuums, solar radiation, and weak magnetic fields bring about LEO satellite faults, which result in packet loss. The high-speed movement of LEO satellites brings about the issue of dynamic distances between satellites, which results in the instability of the transmission delay. The relatively long delay of intersatellite links results in the nonreal-time updating of link information. On the other hand, link evaluations themselves, such as "low delay," "high reliability," and "high bandwidth," possess a certain degree of uncertainty. In summary, traditional routing strategies for satellite networks are not efficient because they do not consider the complicated factors that affect the routing process. Therefore, the uncertainty of link information cannot be ignored.

In recent years, uncertain routing strategies in complex environments have attracted the attention of researchers [24] and have been successfully applied in the field of wireless sensor networks [25, 26]. At present, some researchers have focused on uncertain routing strategies in satellite networks. Zhang et al. [27] transformed multiattribute parameters of satellites into a comprehensive parameter through the Choquet fuzzy integral. Li et al. [28] estimated the congestion status between adjacent satellites through fuzzy congestion indicators and then proposed a fuzzy routing strategy to avoid congestion. Jiang et al. [29] established a fuzzy rule set and proposed a fuzzy routing strategy that satisfies multiservice QoS for satellite networks. This paper considers the uncertainty of link information, such as the transmission delay, packet loss rate, and available bandwidth, and studies the uncertain routing strategy for LEO satellite networks.

To solve the uncertainty problem of LEO satellite link information caused by complex space environments, a routing strategy based on membership degree functions is proposed in this paper. The contributions of this paper are as follows:

(1) A routing model based on uncertain link information is established. Specifically, the uncertain link information is described by the membership degree function, and then the comprehensive evaluation of the path is obtained by integrating different link information of each link in the path. On this basis, the routing model considering uncertainty is established according to the comprehensive evaluation of the path.

(2) A grey wolf optimization (GWO) algorithm is designed to solve the routing model on the premise of ensuring the validity of the path.

(3) Simulation results are presented to show that the proposed strategy can obtain the optimal paths in complex space environments and improve the performance in terms of average delay, packet loss rate, and throughput, compared with traditional routing strategies.

The remainder of the paper is organized as follows: the background of membership degree functions is given in Section 2. The overall process of the proposed strategy is described in Section 3. The routing model based on uncertain link information is established in Section 4. The GWO algorithm is designed to solve the routing model in Section 5. The simulation analysis is given in Section 6. Finally, the conclusion of the paper is given in Section 7.

2. Background of Membership Degree Functions

With the increasing demand for the description and calculation of uncertain data, uncertainty theory continues to evolve. The membership degree function in fuzzy set theory is an efficient tool for describing and dealing with the fuzziness of data [30]. The basic idea of the membership degree function is to extend classical Boolean logic to continuous logic with arbitrary values in a certain interval [31]. Due to the uncertainty of link information, the uncertain link information is described by the membership degree function in fuzzy set theory, which can better reflect the actual situation of a LEO satellite network. The basic information regarding membership degree functions is given in this section.

Let U be the domain. $\mu_A : U[0,1]$ is a mapping from the domain U to the interval $[0,1]$, and A is called a fuzzy subset of U. μ_A is called the membership degree function of A. If $x \in U$, the value of $\mu_A(x)$ represents the fuzzy degree to which x belongs to A [32].

The closer the value of $\mu_A(x)$ is to 1, the higher the degree to which x belongs to A. In contrast, the closer the value of $\mu_A(x)$ is to 0, the lower the degree to which x belongs to A. In particular, when $\mu_A(x) = 1$, x is considered to belong to A entirely. On the other hand, when $\mu_A(x) = 0$, x is not considered to belong to A at all.

In this paper, we suppose that $x \in R^+, U$ is the link information, and A is a certain kind of link information in U, e.g., a low transmission delay, low packet loss rate, or high available bandwidth.

3. Overall Process of the Proposed Strategy

The overall process of the proposed strategy is shown in Figure 1. The specific steps are as follows: first, the virtual node method is employed to solve the dynamic change issue of the LEO satellite network, and then the LEO satellite network is represented by a directed graph. Second, the uncertain link information is described by the designed membership degree function. Third, the distance between the membership degree function of link information and the ideal point is calculated, which is employed as the comprehensive evaluation of the link, and then the comprehensive evaluation of the path is obtained by accumulating the comprehensive evaluations of all links in the path. Fourth, the comprehensive evaluation of the path is taken as the optimization objective for establishing the routing model. Finally, the routing model is solved by GWO, and the optimal path can be obtained.

4. Routing Model Based on Uncertain Link Information

4.1. Link Information Description. In this section, taking the transmission delay, packet loss rate, and available bandwidth as examples, the link information of LEO satellite networks

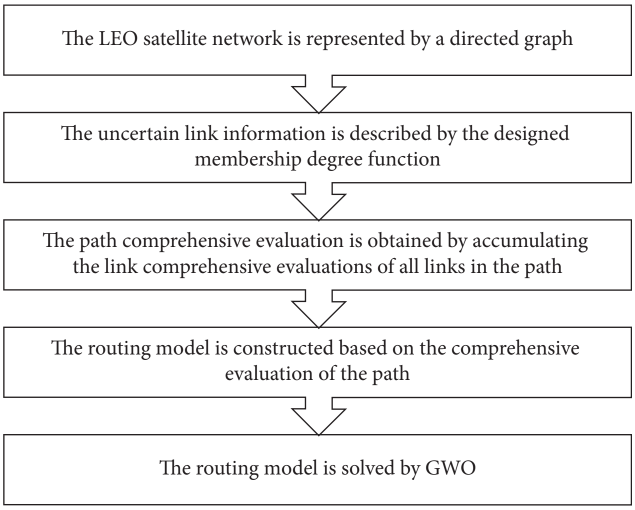

The LEO satellite network is represented by a directed graph
The uncertain link information is described by the designed membership degree function
The path comprehensive evaluation is obtained by accumulating the link comprehensive evaluations of all links in the path
The routing model is constructed based on the comprehensive evaluation of the path
The routing model is solved by GWO

Figure 1: Overall process of the proposed strategy.

is represented by the designed membership degree function, which can better reflect the actual situation of a LEO satellite network. The transmission delay from satellite node g to satellite node k is denoted as td_{gk}. The packet loss rate from satellite node g to satellite node k is denoted as lr_{gk}. The available bandwidth from satellite node g to satellite node k is denoted as bd_{gk}. In particular, td_{gk}, lr_{gk}, and $\mathrm{bd}_{gk} \in R^{+}$.

In general, the link information can be divided into efficiency indexes (the larger the better) and cost indexes (the smaller the better). To eliminate the influences of different types of link information on LEO satellite networks, the link information needs to be normalized.

4.1.1. Membership Degree Function of the Transmission Delay. The high-speed movement of LEO satellites brings about the issue of dynamic distances between satellites. Hence, the transmission delay is dynamically changed, which results in the uncertainty of the transmission delay. When td_{gk} is less than or equal to $\overline{\mathrm{Td}_{gk}}$, the transmission delay is considered to have already conformed to the optimal link, so the membership degree function of the transmission delay is invariably 1. When td_{gk} is greater than $\overline{\mathrm{Td}_{gk}}$, the degree of conformation of the transmission delay to the optimal link stabilizes relatively at the beginning and then decreases rapidly. For that reason, the membership degree function of the transmission delay of link (g,k) is described by the exponential function as follows:

$$\mathrm{mf}_{gk}^{\mathrm{td}} = \begin{cases} 1, & \mathrm{td}_{gk} \le \overline{\mathrm{Td}_{gk}}, \\ e^{-\mathrm{td}_{gk}+\overline{\mathrm{Td}_{gk}}}, & \mathrm{td}_{gk} > \overline{\mathrm{Td}_{gk}}, \end{cases} \quad (1)$$

where $\overline{\mathrm{Td}_{gk}}$ is the transmission delay threshold of link (g,k). Since the transmission delay is an additive parameter, it is calculated as $\overline{\mathrm{Td}_{gk}} = \mathrm{Td}_{ij}/\mathrm{Hc}_{ij}$, where Td_{ij} is the maximum transmission delay threshold of the path, and Hc_{ij} is the maximum hop count threshold of the baseline path. $\overline{\mathrm{Td}_{gk}}$ represents the fact that the transmission delay threshold is averaged to every link in the path. The transmission delay can be denoted by $\mathrm{td}_{gk} = L_{gk}/\mathrm{vt}_{gk}$, where L_{gk} is the link distance from satellite node g to satellite node k and vt_{gk} is the constant of light velocity. The smaller td_{gk} is, the more consistent the optimal link. When td_{gk} is less than or equal to $\overline{\mathrm{Td}_{gk}}$, $\mathrm{mf}_{gk}^{\mathrm{td}}$ is invariably 1, which demonstrates that the current link completely conforms to the optimal link. When td_{gk} is greater than $\overline{\mathrm{Td}_{gk}}$, the membership degree of the transmission delay decreases as td_{gk} increases. That is, the degree of conformation to the optimal link decreases.

4.1.2. Membership Degree Function of the Packet Loss Rate. A harsh space environment, such as a vacuum, an area with solar radiation, or a weak magnetic field, brings about LEO satellite faults, which result in packet loss. Thus, the packet loss rate possesses a certain uncertainty. When lr_{gk} is less than or equal to $\overline{\mathrm{Lr}_{gk}}$, the packet loss rate is considered to have already conformed to the optimal link, so the membership degree function of the packet loss rate is invariably 1. When lr_{gk} is greater than $\overline{\mathrm{Lr}_{gk}}$, the degree of conformation of the packet loss rate to the optimal link decreases linearly as the packet loss rate increases. For that reason, the membership degree function of the packet loss rate of link (g,k) is described by the linear function as follows:

$$\mathrm{mf}_{gk}^{\mathrm{lr}} = \begin{cases} 1, & \mathrm{lr}_{\mathrm{gk}} \le \overline{\mathrm{Lr}_{gk}}, \\ \dfrac{1-\mathrm{lr}_{gk}}{1-\overline{\mathrm{Lr}_{gk}}}, & \mathrm{lr}_{gk} > \overline{\mathrm{Lr}_{gk}}, \end{cases} \quad (2)$$

where $\overline{\mathrm{Lr}_{gk}}$ is the packet loss rate threshold of link (g,k). Since the packet loss rate is a multiplicative parameter, it is calculated as $\overline{\mathrm{Lr}_{gk}} = \sqrt[Hcij]{\mathrm{Lr}_{ij}}$, where Lr_{ij} is the maximum packet loss rate threshold of the path and Hc_{ij} is the maximum hop count threshold of the baseline path. $\overline{\mathrm{Lr}_{gk}}$ represents the fact that the packet loss rate threshold is averaged to every link in the path. The packet loss rate can be denoted by $\mathrm{lr}_{gk} = 1 - (\mathrm{pk}_{gk}^{e}/\mathrm{pk}_{gk}^{s})$, where pk_{gk}^{e} is the total number of packets received by satellite node k and pk_{gk}^{s} is the total number of packets sent by satellite node g. The smaller lr_{gk} is, the more consistent the optimal link. When lr_{gk} is less than or equal to $\overline{\mathrm{Lr}_{gk}}$, $\mathrm{mf}_{gk}^{\mathrm{lr}}$ is invariably 1, which demonstrates that the current link completely conforms to the optimal link. When lr_{gk} is greater than $\overline{\mathrm{Lr}_{gk}}$, the membership degree of the packet loss rate decreases as lr_{gk} increases. That is, the degree of conformation to the optimal link decreases.

4.1.3. Membership Degree Function of the Available Bandwidth. Due to the high-speed movement of LEO satellites, the LEO satellite network is in a state of dynamic change, which results in the uncertainty of the available bandwidth. When bd_{gk} is less than $\overline{\mathrm{Bd}_{gk}}$, the degree of conformation of the available bandwidth to the optimal link increases rapidly at the beginning and then stabilizes relatively. When bd_{gk} is greater than or equal to $\overline{\mathrm{Bd}_{gk}}$, the available bandwidth is considered to have already conformed to the optimal link, so the membership degree function of the available bandwidth is invariably 1. For that reason, the membership degree function of the available bandwidth of link (g,k) is described by the logarithmic function as follows:

$$\mathrm{mf}_{gk}^{\mathrm{bd}} = \begin{cases} 1, & \mathrm{bd}_{gk} \geq \overline{\mathrm{Bd}_{gk}}, \\ \ln\left(\dfrac{\mathrm{bd}_{gk}}{\overline{\mathrm{Bd}_{gk}}} + e - 1\right), & \mathrm{bd}_{gk} < \overline{\mathrm{Bd}_{gk}}, \end{cases} \tag{3}$$

where $\overline{\mathrm{Bd}_{gk}}$ is the available bandwidth threshold of link (g,k). Since the available bandwidth is a concavity parameter, it is calculated as $\overline{\mathrm{Bd}_{gk}} = (\mathrm{bd}_{gk}^{\min} + \mathrm{bd}_{gk}^{\max})/2$, where $\mathrm{bd}_{gk}^{\min}$ is the minimum value of available bandwidth in the previous time period t and $\mathrm{bd}_{gk}^{\max}$ is the maximum value of available bandwidth in the previous time period t. $\overline{\mathrm{Bd}_{gk}}$ represents the average available bandwidth threshold. The available bandwidth can be denoted by $\mathrm{bd}_{gk} = \mathrm{bd}_{gk}^{\mathrm{sum}} - \mathrm{bd}_{gk}^{\mathrm{oc}}$, where $\mathrm{bd}_{gk}^{\mathrm{sum}}$ is the total bandwidth of link (g,k) and $\mathrm{bd}_{gk}^{\mathrm{oc}}$ is the occupied bandwidth of link (g,k). The larger bd_{gk} is, the more consistent the optimal link. When bd_{gk} increases to $\overline{\mathrm{Bd}_{gk}}$, the membership degree of the available bandwidth increases. That is, the degree of conformation to the optimal link increases. When bd_{gk} is greater than $\overline{\mathrm{Bd}_{gk}}$, $\mathrm{mf}_{gk}^{\mathrm{bd}}$ is invariably 1, which demonstrates that the current link completely conforms to the optimal link.

4.2. Comprehensive Evaluation of the Path. For the convenience of the routing calculation, ideal point theory [33] is employed to integrate these three link information criteria, i.e., the transmission delay, packet loss rate, and available bandwidth, into a comprehensive evaluation of the link. Euclidean distance is used to measure the distance between two points in ideal point theory. Therefore, Euclidean distance is adopted to calculate the distance between the membership degree function of the link information and the ideal point. This distance is defined as the comprehensive evaluation of the link. The comprehensive evaluation of the link from satellite node g to satellite node k is calculated as follows:

$$\mathrm{ld}_{gk} = \sqrt{\left(\mathrm{mf}_{gk}^{\mathrm{td}} - 1\right)^2 + \left(\mathrm{mf}_{gk}^{\mathrm{lr}} - 1\right)^2 + \left(\mathrm{mf}_{gk}^{\mathrm{bd}} - 1\right)^2}. \tag{4}$$

The comprehensive evaluation of the path from the source satellite node i to the destination satellite node j can be obtained by accumulating the comprehensive evaluations of all links in path_{ij} as follows:

$$\mathrm{pd}_{ij} = \sum_{\forall (g,k)\in \mathrm{path}_{ij}} \mathrm{ld}_{gk}, \tag{5}$$

where the smaller the pd_{ij} is, the greater the conformation of the path to the optimal path.

4.3. Routing Model Establishment. The routing model is established with the comprehensive evaluation of the path as the optimization objective, and the transmission delay of the path, the packet loss rate of the path, the available bandwidth of the path, and the hop count of the path as constraints. The LEO satellite network is regarded as a directed graph $G(V\ ,\ E)$ using virtual nodes, where V represents the set of all nodes and E represents the set of all links in the satellite network. Suppose $i,j,g,k \in V$, $(g,k) \in E$, and $\mathrm{path}_{ij}$is the path from node i to node j; then, the routing model of $G(V,E)$ is as follows:

$$\begin{cases} \min & \left\{\mathrm{pd}_{ij}\right\} \\ & \displaystyle\sum_{\forall (g,k)\in \mathrm{path}_{ij}} \mathrm{td}_{\mathrm{gk}} \leq \mathrm{Td}_{ij} \\ \text{s.t.} & \\ & \displaystyle\prod_{\forall (g,k)\in \mathrm{path}_{ij}} \left(1 - \mathrm{lr}_{\mathrm{gk}}\right) \geq 1 - \mathrm{Lr}_{ij} \\ & \left\{\mathrm{bd}_{\mathrm{gk}}\right\} \geq \mathrm{Bd}_{ij} \\ \min_{\forall (g,k)\in \mathrm{path}_{ij}} & \displaystyle\sum_{\forall (g,k)\in \mathrm{path}_{ij}} \mathrm{Hc}_{\mathrm{gk}} \leq \mathrm{Hc}_{ij}. \end{cases} \tag{6}$$

In particular, pd_{ij} is the comprehensive evaluation of the path from the source satellite node i to the destination satellite node j. $\sum_{\forall (g,k)\in \mathrm{path}_{ij}} \mathrm{td}_{\mathrm{gk}} \leq \mathrm{Td}_{ij}$ is the transmission delay constraint of the path; that is, the sum of the transmission delays of all links in the selected path must be less than or equal to the maximum transmission delay threshold of path (Td_{ij}). $\prod_{\forall (g,k)\in \mathrm{path}_{ij}} (1 - \mathrm{lr}_{\mathrm{gk}}) \geq 1 - \mathrm{Lr}_{ij}$ is the packet loss rate constraint of the path; that is, the product of the reliability of every link in the selected path must be greater than or equal to the minimum reliability threshold of path $(1 - \mathrm{Lr}_{ij})$. In particular, Lr_{ij} is the maximum packet loss rate threshold of the path. $\min_{\forall (g,k)\in \mathrm{path}_{ij}} \left\{\mathrm{bd}_{gk}\right\} \geq \mathrm{Bd}_{ij}$ is the available bandwidth constraint of the path; that is, the minimum available bandwidth of every link in the selected path must be greater than or equal to the minimum available bandwidth threshold of path (Bd_{ij}). $\sum_{\forall (g,k)\in \mathrm{path}_{ij}} H_{c_{gk}} \leq \mathrm{Hc}_{ij}$ is the hop count constraint of the path; that is, the hop count of the selected path must be less than or equal to the maximum hop count threshold of the baseline path. The baseline path is generated by a depth search with a depth limit (Hc_{ij}). $\mathrm{Hc}_{ij} = \mathrm{num}_{\mathrm{orbit}} + ((\mathrm{num}_{\mathrm{orbit\_sat}})/2)$, where $\mathrm{num}_{\mathrm{orbit}}$ is the number of orbits of LEO satellite networks and $\mathrm{num}_{\mathrm{orbit\_sat}}$ is the number of satellites in one orbit. In particular, if the values of pd_{ij} of two paths are the same, the optimal path is randomly selected from the two paths.

5. Routing Model Solution Based on the Grey Wolf Optimization Algorithm

The solution of the multiconstraint model is an NP-hard problem [34, 35]. The computational complexity of the routing model increases as the satellite network scale increases. As a new-element heuristic group intelligence algorithm, GWO was proposed by Mirjalili et al. [36]. GWO possesses the advantages of fast convergence and a small amount of required calculations, compared with traditional optimization algorithms. Due to the limited computing capacity of satellite networks, GWO is used to solve the routing model in this section. The flowchart of the solution of the routing model based on GWO is shown in Figure 2.

The critical steps of GWO include individual position coding, initial population generation, fitness function design, and prey hunting.

5.1. Individual Position Coding. In this paper, the path is encoded by the satellite node sequence. An individual, which is recorded by a complete path from the source satellite node i to the destination satellite node j, can be expressed as $\text{Paths}_{ij} = \{i, g, k, \ldots, m, j\}$, $i, g, k, m, j \in V$. Correspondingly, the path information set of path_{ij} is expressed as $\text{Info}_{ij} = \{(\text{td}_{ig}, \text{lr}_{ig}, \text{bd}_{ig}),\ (\text{td}_{gk}, \text{lr}_{gk}, \text{bd}_{gk}), \ldots, (\text{td}_{mj}, \text{lr}_{mj}, \text{bd}_{mj})\}$. As shown in Figure 3, the link information of the source satellite node i and the destination satellite node j is, respectively, represented by $(\text{td}_{ig}, \text{lr}_{ig}, \text{bd}_{ig})$ and $(\text{td}_{mj}, \text{lr}_{mj}, \text{bd}_{mj})$, where the transmission delay, packet loss rate, and available bandwidth from satellite node $i$to satellite node g are represented by $\text{td}_{ig}, \text{lr}_{ig}, \text{bd}_{ig}$, respectively. In the process of solving the routing model, the source satellite node i and the destination satellite node j of each individual remain unchanged.

5.2. Initial Population Generation. To ensure the validity of the generated path, the initial path is randomly generated from the source satellite node i to the destination satellite node j. Afterwards, the initial path is placed into the path set (Paths_{ij}).

5.3. Fitness Function Design. The optimization objective (pd_{ij}) of the routing model is used as the fitness function of GWO. The path set (Paths_{ij}) is sorted in ascending order according to the value of the fitness function, and then the top three individuals are selected as path_α, path_β, and path_δ in turn.

5.4. Prey Hunting. path_α, path_β, and path_δ are considered as prey positions. Any path_ω is continuously updated until the optimal path is obtained, as shown in Figure 4, and the updated path (path_ω') is calculated with the guidance of path_α, path_β, and path_δ as follows:

$$\begin{aligned}
R_\alpha &= \left|C_1 \text{path}_\alpha - \text{path}_\omega\right|, \\
R_\beta &= \left|C_2 \text{path}_\beta - \text{path}_\omega\right|, \\
R_\delta &= \left|C_3 \text{path}_\delta - \text{path}_\omega\right|, \\
\text{path}_\omega' &= \frac{\text{path}_\alpha - A_1 R_\alpha + \text{path}_\beta - A_2 R_\beta + \text{path}_\delta - A_3 R_\delta}{3},
\end{aligned} \tag{7}$$

where R_α, R_β, and R_δ are the absolute values of the relative distances between $\text{path}_\omega$and path_α, path_β, and path_δ, respectively. C_1, C_2, and C_3 are the influence factors. A_1, A_2, and A_3 are the convergence impact factors.

6. Simulation Analysis

6.1. Simulation Environment. We simulate the proposed strategy in NS2 and Visual Studio to verify its effectiveness. The LEO satellite network is constructed with the Iridium constellation of polar orbits in this simulation. The network topology is composed of 66 LEO satellites, and the specific parameters are shown in Table 1. The virtual node method is employed to solve the dynamic change issue of LEO satellite networks. According to the coverage area of each satellite in Iridium, Earth's surface is divided into 66 virtual nodes, which is equal to the number of LEO satellites. The positions and topologies of the 66 virtual nodes are shown in Figure 5.

Moreover, the routing update method is consistent with the literature [37]. A centralized routing computing method is adopted at the regional centre of the virtual nodes. LEO satellites monitor their link information in real time. When the monitored link information exceeds a certain threshold, the link information is sent to the regional centre of the virtual nodes. The route is calculated based on the collected link information at the regional centre. After the calculation is completed, the route is sent to the LEO satellites in the region.

A total of 120 user nodes are generated uniformly on the surface of Earth, and each user node generates traffic flows with the same probability. The source-destination pairs are randomly generated.

6.2. Simulation Results. The routing strategy for LEO satellite networks based on membership degree functions (RSSN-MDF) is compared with the routing strategy for satellite networks based on Dijkstra algorithm (RSSN-D), the routing strategy for satellite networks based on multilayer decision making (RSSN-ML) [38], and the routing strategy for satellite networks based on multiobjective decision making (RSSN-MDM) [39] in terms of average delay, delay jitter, packet loss rate, throughput, and comprehensive performance.

Figure 6 shows the comparison of the average delays at different packet sending rates. As shown in this figure, the average delay of RSSN-D increases rapidly as the packet sending rate increases. This is because RSSN-D only takes the transmission delay as the basis of the path calculation and easily becomes congested. RSSN-MDM takes the available bandwidth as one of the link evaluation indicators, effectively avoiding congestion. Consequently, the average delay of RSSN-

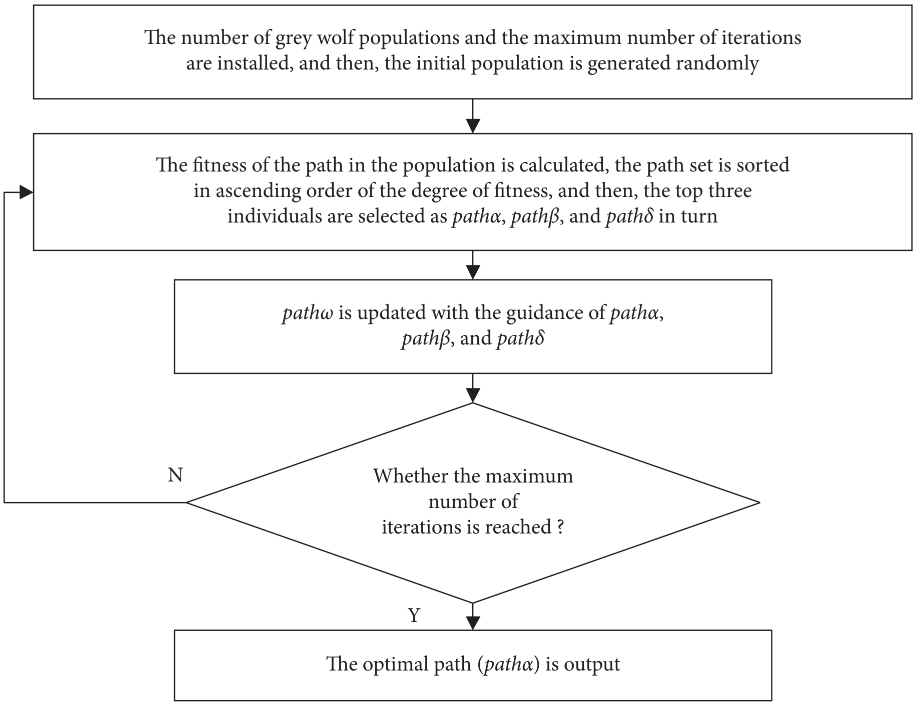

The number of grey wolf populations and the maximum number of iterations are installed, and then, the initial population is generated randomly
The fitness of the path in the population is calculated, the path set is sorted in ascending order of the degree of fitness, and then, the top three individuals are selected as *pathα*, *pathβ*, and *pathδ* in turn
pathω is updated with the guidance of *pathα*, *pathβ*, and *pathδ*
Whether the maximum number of iterations is reached ?
N
Y
The optimal path (*pathα*) is output

Figure 2: Flowchart of the solution of the routing model based on GWO.

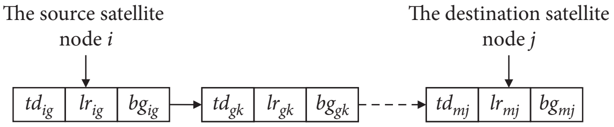


Figure 3: Individual position coding.

MDM is better than that of RSSN-ML. RSSN-MDF takes the transmission delay, packet loss rate, and available bandwidth as the optimization objectives and also considers the uncertainty of link information, so RSSN-MDF can adapt to complex space environments and avoid congestion to some extent. Thus, the optimal path can be obtained by RSSN-MDF. As a result, the average delay of RSSN-MDF is always low.

Figure 7 shows the comparison of delay jitters at different packet sending rates. The delay jitter is the difference between the maximum delay and the minimum delay when the packets pass through the path. As shown in this figure, the delay jitters of RSSN-MDM, RSSN-MDF, and RSSN-D are high and that of RSSN-ML is low. This is because RSSN-ML uses the hierarchical clustering method to calculate the route, which can reduce the complexity of the calculation and decrease the disturbance of the delay. RSSN-MDF takes the transmission delay of the path, packet loss rate of the path, available bandwidth of the path, and hop count of the path as constraints. Meanwhile, RSSN-MDF adopts a centralized routing computing method. The above reasons lead to higher computational complexity. Moreover, there is a delay in the routing update of RSSN-MDF. Some satellites may forward packets according to the old path, and some may forward packets according to the new path. Thus, the difference in the packet delays between RSSN-MDF and the other methods is large. As a result, the delay jitter of RSSN-MDF is high.

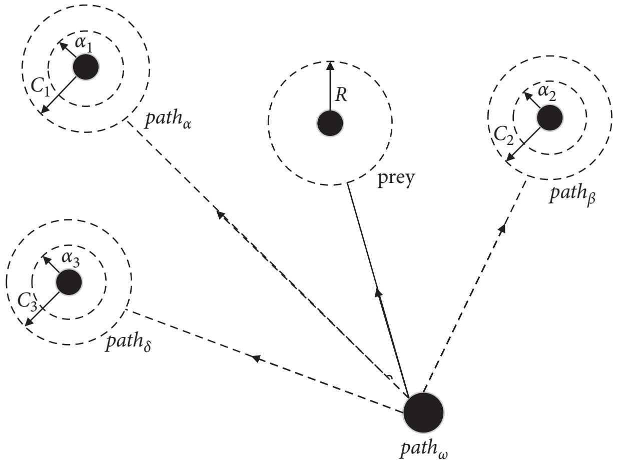


Figure 4: Prey hunting.

Table 1: Parameters of the LEO satellite network.

Parameters	Values
Model	Iridium
Orbit number	6
Satellite number	66
Adjacent orbit interval	27°
Orbit height	780 km
Link bandwidth	25 mbps
Queue buffer size	50 kb

Figure 8 shows the comparison of packet loss rates at different packet sending rates. As shown in this figure, the packet loss rates of RSSN-MDF and RSSN-MDM are better

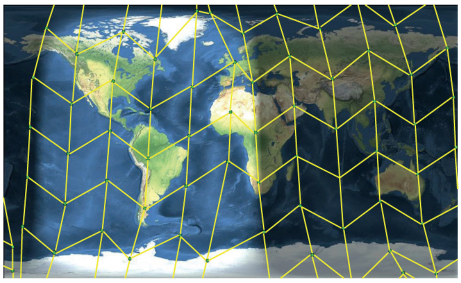


Figure 5: Positions and topologies of the virtual nodes.

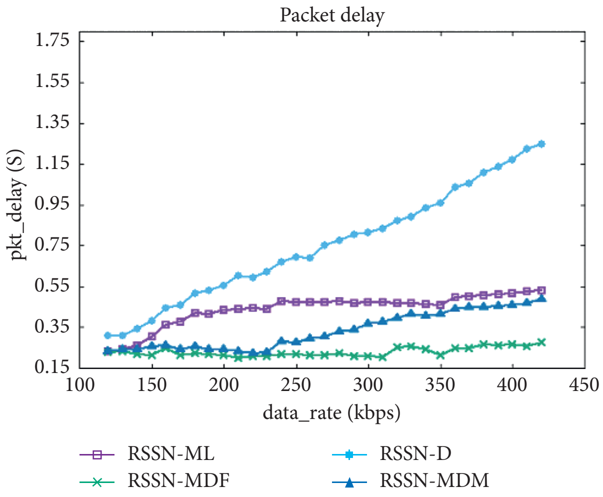


Figure 6: Comparison of average delays.

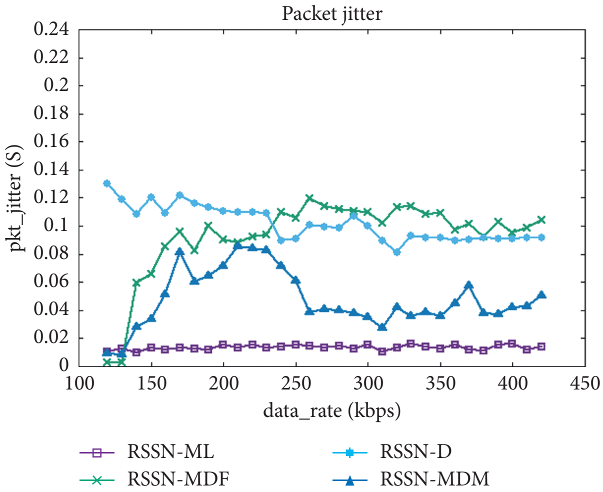


Figure 7: Comparison of delay jitters.

than those of RSSN-ML and RSSN-D. This is because RSSN-MDF and RSSN-MDM are similar in terms of their transmission delays, packet loss rates, and available bandwidth as their link evaluation indicators. In addition, RSSN-ML and RSSN-D calculate their paths according to the determined values without considering the uncertainty of link information. Therefore, the packet loss rates of RSSN-ML and

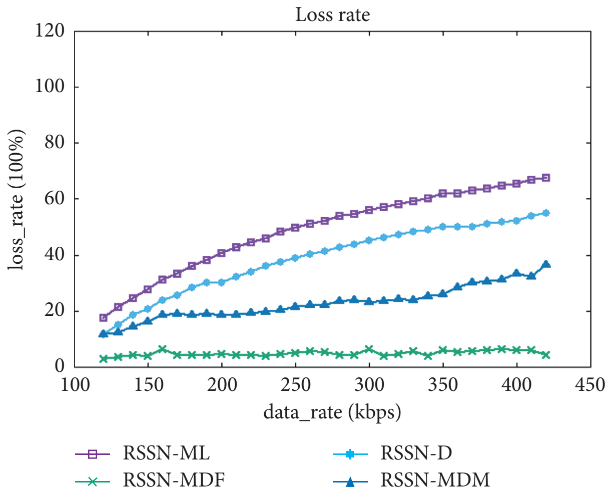


Figure 8: Comparison of packet loss rates.

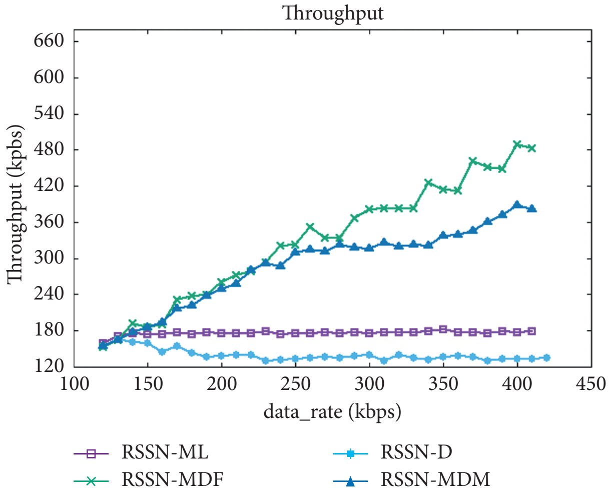


Figure 9: Comparison of throughputs.

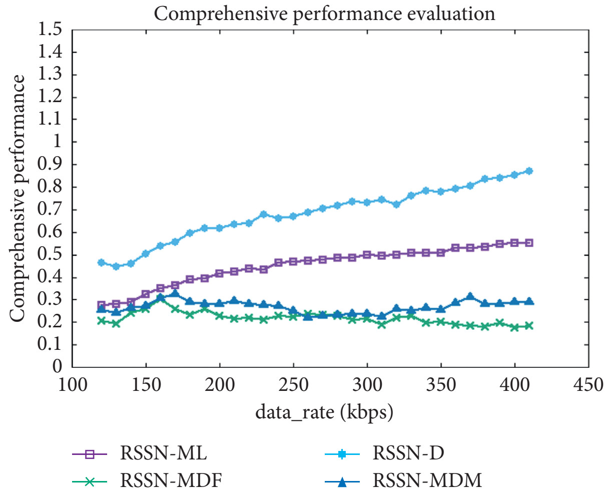


Figure 10: Comparison of comprehensive performances.

RSSN-D rapidly increase as the packet sending rates increase. RSSN-MDF takes the uncertainty of link information into account and introduces path constraints into the routing model, which further improves the reliability of the routing calculation. As a result, the packet loss rate of RSSN-MDF is always low.

Figure 9 shows the comparison of throughputs at different packet sending rates. As shown in this figure, the throughputs of RSSN-ML and RSSN-D are always low. This is because these two strategies calculate their paths according to the determined values; the optimal paths cannot be obtained in complex space environments. The throughput of RSSN-MDF is similar to that of RSSN-MDM at the beginning and then it becomes superior to that of RSSN-MDM. This is because RSSN-MDF not only considers the uncertainty of link information but also takes the available bandwidth as the link evaluation indicator. Thus, RSSN-MDF can adapt to complex space environments and avoid congestion to some extent. As a result, the throughput of RSSN-MDF increases as the packet sending rate increases.

To evaluate the overall performances of the four strategies, the comprehensive performance is used as an index. First, the average delay, delay jitter, packet loss rate, and throughput are all transformed into cost indexes, and min-max normalization is carried out. Then, the comprehensive performance is calculated by the average weighted summation of these normalized indexes. The smaller the comprehensive performance value is, the better the routing strategy will perform. Figure 10 shows the comparison of comprehensive performances at different packet sending rates. As the packet sending rates increase, the comprehensive performances of RSSN-D and RSSN-ML worsen, while those of RSSN-D and RSSN-ML are relatively stable. As shown in this figure, the comprehensive performance of the proposed strategy is the best. In summary, RSSN-MDF can achieve efficient and secure routing in complex space environments.

7. Conclusions

Research on routing strategies is significant for MEC satellites. However, the complexity of LEO satellite networks leads to the uncertainty of link information. In this paper, a routing strategy for LEO satellite networks based on membership degree functions is proposed. First, a routing model based on uncertain link information is established. Specifically, the uncertain link information is described by the designed membership degree function, and then the comprehensive evaluation of the path is obtained by integrating different link information of each link in the path. Afterwards, the comprehensive evaluation of the path is taken as the optimization objective to establish the routing model. Second, a GWO algorithm is designed to solve the routing model. Finally, simulation results show that the proposed strategy can achieve efficient and secure routing in complex space environments. Although the proposed strategy loses some delay jitter, it improves the performance in terms of average delay, packet loss rate, and throughput, compared with the performances of traditional routing strategies. The influence of the degree of uncertainty will be taken into consideration in future research.

Data Availability

The simulated evaluation data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61972210, 61873131, 61872191, and 61803212 and by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province under Grant 18KJB520034.

References

[1] X. Xu, Q. Wu, L. Qi, W. Dou, S.-B. Tsai, and M. Z. A. Bhuiyan, "Trust-Aware service offloading for video surveillance in edge computing enabled internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, p. 1, 2020.

[2] W. Hou, Z. Ning, and L. Guo, "Green survivable collaborative edge computing in smart cities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1594–1605, 2018.

[3] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, and L. Qi, "Trust-oriented IOT service placement for smart cities in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4084–4091, 2020.

[4] Y. Lu, Y. J. Zhao, F. C. Sun et al., "Routing techniques on satellite networks," *Journal of Software*, vol. 25, no. 5, pp. 1085–1100, 2014.

[5] J. Sun, Y. Zhang, Z. Wu et al., "An efficient and scalable framework for processing remotely sensed big data in cloud computing environments," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4294–4308, 2019.

[6] Z. Zhang, W. Zhang, and F. H. Tseng, "Satellite mobile edge computing: improving QoS of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Network*, vol. 33, no. 1, pp. 70–76, 2018.

[7] F. Wang, D. Jiang, S. Qi, C. Qiao, and L. Shi, "A dynamic resource scheduling scheme in edge computing satellite networks," *Mobile Networks and Applications*, vol. 14, 2020.

[8] L. Yan, S. Cao, Y. Gong et al., "SatEC: a 5G satellite edge computing framework based on microservice architecture," *Sensors*, vol. 19, no. 4, Article ID 831, 2019.

[9] Y. Wang, J. Yang, X. Guo, and Z. Qu, "A game-theoretic approach to computation offloading in satellite edge computing," *IEEE Access*, vol. 8, pp. 12510–12520, 2020.

[10] Y. Wang, J. Yang, X. Guo et al., "Satellite edge computing for the internet of things in aerospace," *Sensors*, vol. 19, no. 20, Article ID 4375, 2019.

[11] J. Wei, J. Han, and S. Cao, "Satellite IoT edge intelligent computing: a research on architecture," *Electronics*, vol. 8, no. 11, Article ID 1247, 2019.

[12] P. Cao, M. M. Fan, K. Liu et al., "Dynamic programming modeling of satellite system communication routing

problems," in *Proceedings of the National Youth Conference on Information and Management Sciences*, New York, NY, USA, 2012.

[13] L. Wood, A. Clerget, I. Andrikopoulos et al., "IP routing issues in satellite constellation networks," *International Journal of Satellite Communications*, vol. 19, pp. 69–92, 2010.

[14] M. Werner, "A dynamic routing concept for ATM-based satellite personal communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 8, pp. 1636–1648, 1997.

[15] D. Fischer, D. Basin, and T. Engel, "Topology dynamics and routing for predictable mobile networks," in *Proceedings of the IEEE International Conference on Network Protocols*, Berlin, Germany, 2008.

[16] J. Wang, L. Li, and M. Zhou, "Topological dynamics characterization for LEO satellite networks," *Computer Networks*, vol. 51, no. 1, pp. 43–53, 2007.

[17] R. Mauger and C. Rosenberg, "QoS guarantees for multimedia services on a TDMA-based satellite network," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 56–65, 1997.

[18] E. Ekici, I. F. Akyildiz, and M. D. Bender, "A distributed routing algorithm for datagram traffic in LEO satellite networks," *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, pp. 137–147, 2001.

[19] Y. Hashimoto, "Design of IP-based routing in a LEO satellite network," in *Proceedings of the International Workshop on Satellite-Based Information Services*, London, UK, 1998.

[20] H. Nishiyama, Y. Tada, N. Kato et al., "Toward optimized traffic distribution for efficient network capacity utilization in two-layered satellite networks," *IEEE Transactions On Vehicular Technology*, vol. 62, pp. 1303–1313, 2012.

[21] F. Xiao, L. J. Sun, X. G. Ye et al., "Routing algorithm for MPLS traffic engineering in satellite network," *Journal on Communications*, vol. 35, pp. 104–111, 2011.

[22] J. Wang, Y. J. Guo, L. J. Sun et al., "Load balancing algorithm for multi-traffic in double layered satellite network," *Journal of Systems Engineering and Electronics*, vol. 38, pp. 2156–2161, 2016.

[23] J. Feng and G. Q. Gu, "Research on QoS routing based on uncertain parameters," *Journal of Computer Research and Development*, vol. 39, pp. 553–539, 2002.

[24] X.-W. Wang, Q. Wang, M. Huang et al., "A fuzzy integral and game theory based QoS multicast routing scheme," *Journal of Software*, vol. 19, no. 7, pp. 1743–1752, 2008.

[25] L. Cobo, A. Quintero, and S. Pierre, "Ant-based routing for wireless multimedia sensor networks using multiple QoS metrics," *Computer Networks*, vol. 54, no. 17, pp. 2991–3010, 2010.

[26] X.-M. Wang, J.-L. Lu, Y.-S. Li, and K.-G. Hao, "Multiconstrained multipath routing for wireless sensor networks in the fuzzy random environment," *Chinese Journal of Computers*, vol. 34, no. 5, pp. 779–791, 2011.

[27] X. Y. Zhang and T. Zhang, "The mechanism based on fuzzy integral in multi-attribute optimal routing," in *Proceedings of the IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, New York, NY, USA, 2011.

[28] C. Li, C. Liu, Z. Jiang et al., "A novel routing strategy based on fuzzy theory for NGEO satellite networks," in *Proceedings of the IEEE Vehicular Technology Conference*, Berlin, Germany, 2015.

[29] Z. Jiang, C. Liu, S. He et al., "A QoS routing strategy using fuzzy logic for NGEO satellite IP networks," *Wireless Networks*, vol. 24, pp. 1–13, 2018.

[30] X. Zhou, G. Zhang, J. Sun, J. Zhou, T. Wei, and S. Hu, "Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT," *Future Generation Computer Systems*, vol. 93, pp. 278–289, 2019.

[31] X. Y. Zhou, K. Q. Zou, and Y. F. Wang, "Fuzzy variable time series based on fuzzy membership function and econometrics," in *Proceedings of the International Symposium on Knowledge Acquisition and Modeling*, New York, NY, USA, 2010.

[32] S. L. Chen, J. G. Li, and X. G. Wang, *Fuzzy Set Theory and its Application*, Science Press, Beijing, China, 2005.

[33] C. L. Hwang and K. Yoon, *Multiple Attribute Decision Making*, Spring-Verlag, Berlin, Germany, 1981.

[34] M. Miklós, B. Alia, and L. Samer, "The cost optimal solution of the multi-constrained multicast routing problem," *Computer Networks*, vol. 56, pp. 3136–3149, 2012.

[35] J. L. Zhou, J. Sun, P. J. Cong et al., "Security-critical energy-aware task scheduling for heterogeneous real-time MPSoCs in IoT," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 1–14, 2020.

[36] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[37] S. D. Zhang, L. J. Sun, J. Zhou et al., "Destruction-resistant routing strategy for GEO/LEO double-layer satellite networks," *Journal of Nanjing University of Posts and Telecommunications*, vol. 38, pp. 1–7, 2018.

[38] I. F. Akyildiz, E. Ekici, and M. D. Bender, "MLSR: a novel routing algorithm for multilayered satellite IP networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 3, pp. 411–424, 2002.

[39] L. Yang, J. Sun, C. S. Pan et al., "LEO multi-service routing algorithm based on multi-objective decision making," *Journal on Communications*, vol. 37, pp. 25–32, 2016.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8848389, 15 pages
https://doi.org/10.1155/2020/8848389

Research Article

A Novel Machine Learning-Based Approach for Security Analysis of Authentication and Key Agreement Protocols

Behnam Zahednejad, Lishan Ke, and Jing Li

Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou, China

Correspondence should be addressed to Lishan Ke; kelishan@gzhu.edu.cn

Received 2 May 2020; Revised 29 July 2020; Accepted 25 September 2020; Published 16 October 2020

Academic Editor: Xiaolong Xu

The application of machine learning in the security analysis of authentication and key agreement protocol was first launched by Ma et al. in 2018. Although they received remarkable results with an accuracy of 72% for the first time, their analysis is limited to replay attack and key confirmation attack. In addition, their suggested framework is based on a multiclassification problem in which every protocol or dataset instance is either secure or prone to a security attack such as replay attack, key confirmation, or other attacks. In this paper, we show that multiclassification is not an appropriate framework for such analysis, since authentication protocols may suffer different attacks simultaneously. Furthermore, we consider more security properties and attacks to analyze protocols against. These properties include strong authentication and Unknown Key Share (UKS) attack, key freshness, key authentication, and password guessing attack. In addition, we propose a much more efficient dataset construction model using a tenth number of features, which improves the solving speed to a large extent. The results indicate that our proposed model outperforms the previous models by at least 10–20 percent in all of the machine learning solving algorithms such that upper-bound performance reaches an accuracy of over 80% in the analysis of all security properties and attacks. Despite the previous models, the classification accuracy of our proposed dataset construction model rises in a rational manner along with the increase of the dataset size.

1. Introduction

Security protocols (cryptographic protocols) are widely used to transport application-level data in a secure manner. These protocols usually apply a sequence of cryptographic primitives such as (a)symmetric encryption, digital signature, and hash function. The most important goals of security protocols include key agreement or establishment, entity authentication, message authentication, and nonreputation [1]. For instance, Transport Layer Security (TLS) [2] is a well-known cryptographic protocol that is used to provide secure web connections (HTTPS). To prove the correctness of security protocols, various methods were developed over the last decades. These methods can be divided into two main categories.

Model-checking methods refer to the set of automated tools and methods that try to find attacks which violate security goals, rather than proving their correctness. ProVerif [3], Scyther [4], AVISPA [5], CryptoVerif [6], and so on are among the most well-known tools. *Theorem-proving* methods are less automated methods that consider all possible protocol behavior to check whether the security goal is achieved or not. Although they cannot give a security attack, they provide a proof of the correctness of the protocol. BAN logic [7], Dolev-Yao model [8], and strand space [9] are examples of these methods.

1.1. Motivation and Goal of This Paper. The goal of this paper is to develop a novel machine learning-based protocol analysis scheme with much better efficiency that can discover more security attacks and vulnerabilities. Previously, the application of machine learning in security analysis has been mainly limited to side-channel attack [10, 11] and symmetric cryptoanalysis [12, 13]. Our motivation for applying machine learning in protocol analysis is described as follows:

(1) The most important limitation of classical methods is the fact that, to a large extent, analysis results rely on the prior knowledge and experience of the analysts. It frequently happens that even if a security protocol is found to be correct by a model-checking or theorem-proving method, another more experienced researcher discovers a new attack against the same protocol. For example, Tingyuan et al. [14] proved the security of the Otway-Rees protocol. Later, Liu et al. [15] also used BAN logic to point out that this protocol is vulnerable to man-in-the-middle attack and typing flaw attack. Therefore, researchers are trying to discover other methods to guarantee security in cyberspace.

(2) Inspired by the astonishing results of the application of machine learning in cybersecurity [16, 17], Ma et al. [18] designed a machine learning-based model to master the machine in the security analysis of protocols. They suggested a multiclassification model in which every protocol is either secure or prone to replay attack, lacks key confirmation, or is prone to other attacks. Although they received remarkable results for the first time, their analysis is limited to only replay attack and key confirmation. In addition, it frequently happens that a protocol is prone to two or three attacks at the same time (e.g., replay attack and lack of key confirmation). Therefore, multiclassification is not an appropriate model for this purpose. Further, their dataset size is so small, i.e., less than 100 instances for each category.

1.2. Contributions and Structure of This Paper. This paper has three main contributions:

(1) We use a machine learning framework to analyze more security properties such as strong entity authentication and Unknown Key Share (UKS) attack, key freshness, key authentication, and resistance to password guessing attack.

(2) To analyze every research problem in machine learning, the features of the problem should be first extracted. Ma et al. [18] suggested three models, namely, LCM, TLM, and SLM, to extract the features of every protocol as a weighted matrix. We propose a new model with much less number of features which improves the convergence speed.

(3) We propose a binary classification model for each category in which each instance of the dataset either violates one security property or is secure against that. Further, we develop more than 1000 datasets for each category, which is 10 times more than the previous work [18]. Inspired by Ma et al.'s scheme, we also use XGBoost [19] to estimate the classification accuracy of the analysis. In addition, a dense neural network (DNN) was deployed to integrate the deep learning approach to the protocol analysis problem.

The rest of the paper is organized as follows. In Section 2, we briefly introduce authentication and key agreement protocol along with their security goals and attacks. Section 3 discusses the application of machine learning in the security analysis of protocols. In this section, we propose our model to analyze more security properties, that is, strong entity authentication and Unknown Key Share (UKS) attack, key freshness, and so on. The experimental results of the analysis are described in Section 4. Finally, a conclusion is given in Section 5.

2. Authentication and Key Establishment Protocols

Authentication and key establishment protocols are the backbone of any secure electronic communication. Cryptographic algorithms such as AES and DES [20, 21] cannot be implemented unless common secret keys are preshared (key establishment) and communication parties know who owns such keys (authentication). Authentication and key establishment protocols achieve these goals by using a set of messages consisting of random numbers, identities, timestamps, hash function, and so on. For example, consider the ISO symmetric key two-pass unilateral authentication and key establishment protocol [22] between two parties like Alice and Bob in Figure 1.

Here, Alice sends a random number N_A to Bob. The secret key K_{AB} is preshared between Alice and Bob. When Bob sends message $E_{K_{AB}}(N_A)$ to Alice, she makes sure that it was sent by Bob because he only has the key K_{AB}. In other words, she authenticates Bob. Authentication protocols are widely used in different applications such as wireless networks [23], smart city [24], and Internet of Things (IoT) [25]. Authentication and key establishment are the two main goals of cryptographic protocols [1]. In the following, we describe more detailed security goals based on authentication and key establishments. Then we present the most common attacks that try to violate these goals. Figure 2 shows the set of security attacks and goals of authentication protocols.

2.1. Authentication Goals. According to ISO security architecture [26], authentication is defined as the "assurance that an entity is the one who claims to be." More precisely, two kinds of authentications can be distinguished as follows.

2.1.1. Entity Authentication. Entity authentication is the process whereby one party is assured of the identity of the second party in the protocol and that the second party has actually participated [27]. This definition assures one party (e.g., A) that the other party (B) has participated in the protocol. It does not provide assurance for A that B also recognized A as his/her peer entity. For example, suppose the protocol of Figure 3(a).

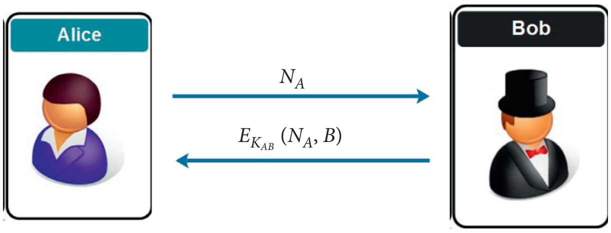


Figure 1: ISO symmetric authentication protocol.

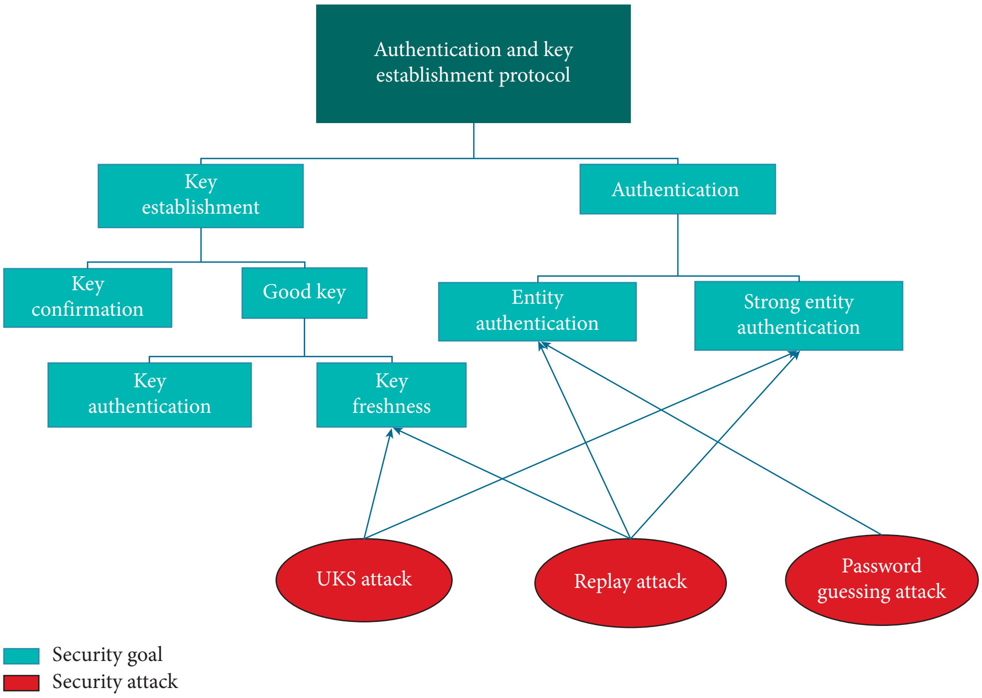


Figure 2: Security goals and attacks in cryptography protocols.

$$\begin{array}{l} 1.A \longrightarrow B : N_A \\ 2.B \longrightarrow A : Sig_B\,(N_A) \end{array}$$

(a)

$$\begin{array}{l} 1.A \longrightarrow C_B : N_A \\ 1'.C \longrightarrow B : N_A \\ 2'.B \longrightarrow C : Sig_B\,(N_A) \\ 2.C_B \longrightarrow A : Sig_B\,(N_A) \end{array}$$

(b)

$$\begin{array}{l} 1.A \longrightarrow B : N_A \\ 2.B \longrightarrow A : Sig_B\,(A, N_A) \end{array}$$

(c)

Figure 3: Strong authentication in security protocols. (a) A security protocol without strong authentication, (b) an attack against strong authentication and (c) improved protocol with strong authentication.

In this protocol, since B signs the nonce N_A, the entity A is assured that B has participated in the same protocol running as entity A. However, the entity B may suppose another entity like C, as his/her peer identity. Figure 3(b) shows an attack against this protocol. In this attack, the adversary C masquerades himself as entity B to A. At the same time, he begins a parallel session with entity B and forwards the response of B to A. As a result of this attack, entity A believes that he/she is contacting with entity B, while B assumes C as his peer entity.

2.1.2. Strong Entity Authentication. Strong entity authentication of A to B is provided if B has a fresh assurance that A has knowledge of B as his/her peer entity [1]. Based on this property, the adversary C has no way to convince B that he/she is in contact with C. Figure 3(c) shows an enhanced version of the protocol of Figure 3(a). In this protocol, the entity B signs his peer identity (ID_A) to make A sure that he recognizes A as his peer entity. As another example, consider protocol SPLICE/AS in Figure 4(a), designed by Yamaguchi et al. [28] to provide mutual authentication between client A and server B. However, Clark and Jacob [29] reported that this protocol cannot provide strong mutual authentication. As shown in Figure 4(b), the attacker C can replace the signature of A with C's signature. As a result, the entity A believes that the protocol has been held with entity B, while B assumes C as his peer entity. To prevent this attack, Clark and Jacob proposed to include the encrypted

$1.A \longrightarrow B : A, B, T_A, E_B(N_A), Sig_A(A, T_A, E_B(N_A))$

$2.B \longrightarrow A : B, A, E_A(B, N_A + 1)$

(a)

$1.A \longrightarrow C_B : A, B, T_A, E_B(N_A), Sig_A(A, T_A, E_B(N_A))$

$1'.C \longrightarrow B : C, B, T_A, E_B(N_A), Sig_C(C, T_A, E_B(N_A))$

$2'.B \longrightarrow C : B, C, E_C(B, N_A + 1)$

$2.C_B \longrightarrow A : B, A, E_A(B, N_A + 1)$

(b)

$1.A \longrightarrow B : A, B, T_A, E_B(ID_A, N_A), Sig_A(A, T_A, E_B(ID_A, N_A))$

$2.B \longrightarrow A : B, A, E_A(B, N_A + 1)$

(c)

Figure 4: Strong authentication in SPLICE/AS protocol. (a) SPLICE/AS protocol. (b) Clark and Jacob's attack against SPLICE/AS protocol. (c) Clark and Jacob's improved protocol.

identity of initiator (ID_A) in the message sent to responder (B). A modified version of this protocol is shown in Figure 4(c).

2.2. Key Establishment. Key establishment is the process whereby a shared secret (session key) becomes available to two or more parties, for subsequent cryptographic use [30]. In this regard, the following goals are assumed for cryptographic protocols.

2.2.1. Good Key. Usually, a session key is only useful if it is known to be fresh and shared to only authenticated and trusted parties. We call it a good key, if it achieves both requirements. More formally, the shared session key is a good key for A to use with B only if A is sure that the following requirements are both satisfied [1]:

(1) Key Freshness. Key freshness is achieved when the communicating parties are able to verify and make sure that the session key they agree with each other is fresh (new) and not replayed from an old session. This is usually achieved by a freshness value. There are two main freshness values used in cryptography protocol: timestamps and nonce [31].

(2) Timestamps. In this method, the current time of the sender is added to the key. As the receiver obtains the message, if there is an acceptable delay, the key is accepted. Otherwise, it aborts. The difficulty of using this method is clock synchronization requirements of sender and receiver. For example, consider the Denning and Sacco protocol [32] depicted in Figure 5. In this protocol, if the timestamp T_S is in a reasonable delay, the parties A and B make sure of the freshness of key K_{AB}.

(3) Nonce. In this method, before the sender sends the key, the recipient, for example, A, generates a nonce, N_A, and transfers it to party B. Then, the nonce, N_A, and the session key K_{AB} are both encrypted and sent to recipient, A. For example, consider the improved MSR protocol of Figure 6. In this protocol, the party B transfers K_{AB} to party A. In addition, it encrypts the nonce N_A with the session key K_{AB}. As the party A decrypts the message with the session key K_{AB} and obtains N_A, it makes sure of the freshness of the key.

$1.A \longrightarrow S : ID_A, ID_B$

$2.S \longrightarrow A : E_{K_{AS}}(ID_A, K_{AB}, T_S)$

$3.S \longrightarrow B : E_{K_{BS}}(ID_A, K_{AB}, T_S)$

Figure 5: Using timestamp as freshness value in Denning and Sacco protocol.

(4) Key Authentication. Key authentication is defined as follows: the key should only be known to A and B and any mutually trusted parties (Gollmann [33] points out that this property can be regarded as the confidentiality of the key). For example, consider the Otway-Rees protocol of Figure 7(a). In this protocol, the server S distributes the session key K_{AB} to A and B. However, as pointed out by Boyd and Mao [34], the attacker can easily mount the attack of Figure 7(b). As a result of this attack, A believes that the key K_{AB} is shared with B, while it is shared with the adversary C. This is a violation of key authentication, as the adversary has access to session key K_{AB}. Abadi and Needham prevented this attack by proposing the protocol shown in Figure 7(c).

2.2.2. Key Confirmation. Key confirmation of A to B is provided if B has assurance that key K is a good key to communicate with A and that principal A has possession of K [1]. Key confirmation provides evidence for a party that his peer partner has received the same key. However, it does not imply entity authentication, as the key may be assumed to be shared with somebody else. In addition, this property cannot be provided for both parties, as one party should finish the protocol.

2.3. Security Attacks. There are many attacks that try to violate the security goals of cryptographic protocols. The most common attacks are described as follows. For more information about other types of attacks, refer to [1].

2.3.1. Unknown Key Share (UKS) Attack. As defined by Blake-Wilson and Menezes [35], an Unknown Key Share (UKS) attack is an attack whereby an entity A ends up believing that she shares a key with B and although this is in fact the case, B mistakenly believes the key is instead shared with entity E ≠ A. This attack targets strong authentication and key freshness of the protocol. For example, consider the Helsinki Protocol in Figure 8(a). A UKS attack on Helsinki's

$$1.A \longrightarrow B : ID_A, Cert_A, N_A$$
$$2.B \longrightarrow A : E_A\,(K_{AB}), E_{K_{AB}}\,(ID_A, Cert_B, N_A)$$

Figure 6: Using nonce as freshness value in MSR protocol.

protocol was published by Horng and Hsu [36]. As shown in Figure 8(b), B ends up believing she shares a session key $f\,(K_{BA}, K'_{AB})$ with A. However, A assumes C as his peer entity whom he shared the key $f\,(K_{AB}, K_{BA})$ with. Mitchell and Yeun [37] proposed to improve this protocol by adding B's identity to message 2 (Figure 8(c)).

2.3.2. Replay Attack. Replay attack occurs whenever the adversary interferes with the protocol run by inserting a message which has been captured in previous sessions of the protocol. Usually, this attack is used to mount other types of attacks, as well. A detailed taxonomy of replay attacks is described by Syverson [38].

2.3.3. Password Guessing Attack. Another common attack to compromise the authentication of legitimate users is through offline guessing of users' passwords. Passwords are generally used to encrypt messages or to authenticate one party to another. In this attack, the adversary needs to access some public parameters and messages, which are usually captured by eavesdropping. If the parameters that are coupled with the password are known to the adversary, he/she can guess the password (as they are usually of low entropy) and check the correctness of his/her guess. For example, if the password of user A is transmitted as $h(\text{Password}_A)$, the attacker can easily guess Password_A and check its correctness by taking a hash of it as $h(\text{Password}_A)$ and see if it is equal to the transmitted message $(h(\text{Password}'_A)? = h(\text{Password}_A))$. However, if the Password_A is coupled with unknown and high entropy parameters like random number N_A, the attacker has no way to check the correctness of his guess [39].

3. Application of Machine Learning in Security Analysis of Authentication and Key Agreement Protocols

The idea of using machine learning to analyze authentication and key agreement protocols was first presented by Ma et al. [18], who suggested training the network by designing a classification problem. Similar to any classification problem in machine learning, we need a set of datasets and their corresponding categories (labels) to train the network. Here, every protocol is an instance of the dataset and the attack that the protocol is vulnerable to is its label. After training the network with a set of protocols and the attacks (categories) that they are vulnerable to, we expect the network to analyze unseen and new protocols and find what kind of attack they are prone to. In this regard, we need a model to map every protocol to an instance of the dataset. In the following, we discuss the dataset model and the categories (labels) of the problem.

3.1. Dataset Construction Model. Dataset construction model is a mapping relation between protocol messages and instances of the dataset. Ma et al. suggested two approaches to convert every protocol to an instance of the dataset. Here, every protocol $P = m_1, m_2, \ldots, m_k$ corresponds to a matrix in the dataset and every message m_i of the protocol corresponds to a vector of the matrix. Before the description of Ma et al.'s dataset models, some definitions are given as follows. Firstly, a message parameter set SP and a parameter property set PP are defined for every message of the protocol:

$$\begin{aligned} \text{SP} &= \{sp_1, sp_2, \ldots, sp_n\}, \\ \text{PP} &= \{pp_1, pp_2, \ldots, pp_n\}. \end{aligned} \tag{1}$$

Here, sp_i denotes any message parameter such as timestamp, participant identity, and random number. Also, pp_i denotes message attributes such as index of parameters, encryption key, and signature key. For example, consider the protocol of Figure 1. This protocol consists of messages m_1 and m_2. For each message, the set of message parameters SP and parameter property sets PP are as follows:

$$\begin{aligned} \text{SP} &= \{N_A, ID_A\}, \\ \text{PP} &= \{K_{AB}\}. \end{aligned} \tag{2}$$

In addition, the lengths of SP and PP are assumed to be fixed (N and M, resp.). If the lengths of the SP and PP are less than N and M, zero values are added to the set. As a result, every message is described by an $N * M$ vector. To reduce the dimension of the message vector, a normalization function is defined as follows:

$$f_n(sp_i) = f_n(PP_i) = f_n(pp_1, pp_2, \ldots, pp_m) = \lambda_i. \tag{3}$$

In the following, after reviewing Ma et al.'s dataset model, we describe our proposed dataset model followed by its comparison with Ma et al.'s dataset model.

3.1.1. Review of Ma et al.'s Model. Ma et al. developed three models, namely, TLM (two-layer model), LCM (literal conversion model), and SLM (single-layer model), to convert every protocol message to a message vector. LCM and SLM models are almost the same. Further, their subtle differences are not clearly explained in [18]. In the following, we only describe TLM and SLM models:

(1) Two-Layer Model (TLM). In TLM, an empty message vector $m_i = sp_{i,1}, sp_{i,2}, \ldots, sp_{i,N}$ is predefined. Here, N is the maximum number of message parameters in the whole dataset. Here, $sp_{i,j}$ is a predefined zero vector of size M. Every dimension of this vector corresponds to a specific property such as plaintext index, encryption key index, and signature key index. For every message parameter sp_i, the property parameters are filled according to the actual protocol. As a result, every message is represented as an $N * M$ vector. Figure 9 shows an example of the TLM conversion model of ISO symmetric authentication protocol in Figure 1.

$1.A \longrightarrow B : A, B, \{N_A, A, B\}\, K_{AS}$

$2.B \longrightarrow S : A, B\, \{N_A, A, B\}\, K_{AS}, \{N_B, A, B\}\, K_{BS}$

$3.S \longrightarrow B : \{N_A, K_{AB}\}\, K_{AS}, \{N_B, K_{AB}\}\, K_{BS}$

$4.B \longrightarrow A : \{N_A, K_{AB}\}\, K_{AS}$

(a)

$1.A \longrightarrow C_B : A, B, \{N_A, A, B\}\, K_{AS}$

$2.C_B \longrightarrow S : A, C, \{N_A, A, B\}\, K_{AS}, \{N_C, A, B\}\, K_{BS}$

$3.S \longrightarrow C_B : \{N_A, K_{AB}\}\, K_{AS}, \{N_B, K_{AB}\}\, K_{BS}$

$4.C_B \longrightarrow A : \{N_A, K_{AB}\}\, K_{AS}$

(b)

$1.A \longrightarrow B : A, B, N_A$

$2.B \longrightarrow S : A, B, N_A, N_B$

$3.S \longrightarrow B : \{N_A, A, B, K_{AB}\}\, K_{AS}, \{N_B, A, B, K_{AB}\}\, K_{BS}$

$4.B \longrightarrow A : \{N_A, A, B, K_{AB}\}\, K_{AS}$

(c)

FIGURE 7: Key authentication in Otway-Rees protocol. (a) Otway-Rees protocol. (b) Boyd and Mao's attack against Otway-Rees protocol. (c) Abadi and Needham's improved protocol.

$1.A \longrightarrow B : E_B\, (A, K_{AB}, N_A)$

$2.B \longrightarrow A : E_B\, (K_{BA}, N_A, N_B)$

$3.A \longrightarrow B : N_B$

(a)

$1.A \longrightarrow C : E_C\, (A, K_{AB}, N_A)$

$1'.C_A \longrightarrow B : E_B\, (A, K_{AB}, N_A)$

$2'.B \longrightarrow C_A : E_B\, (K_{BA}, N_A, N_B)$

$2.C \longrightarrow A : E_B\, (K_{BA}, N_A, N_B)$

$3.A \longrightarrow C : N_B$

$3'.C_A \longrightarrow B : N_B$

(b)

$1.A \longrightarrow B : E_B\, (A, K_{AB}, N_A)$

$2.B \longrightarrow A : E_B\, (B, K_{BA}, N_A, N_B)$

$3.A \longrightarrow B : N_B$

(c)

FIGURE 8: UKS attack in Helsinki protocol. (a) Helsinki's protocol. (b) UKS attack against Helsinki protocol. (c) Mitchell and Yeun's improved protocol resistant against UKS.

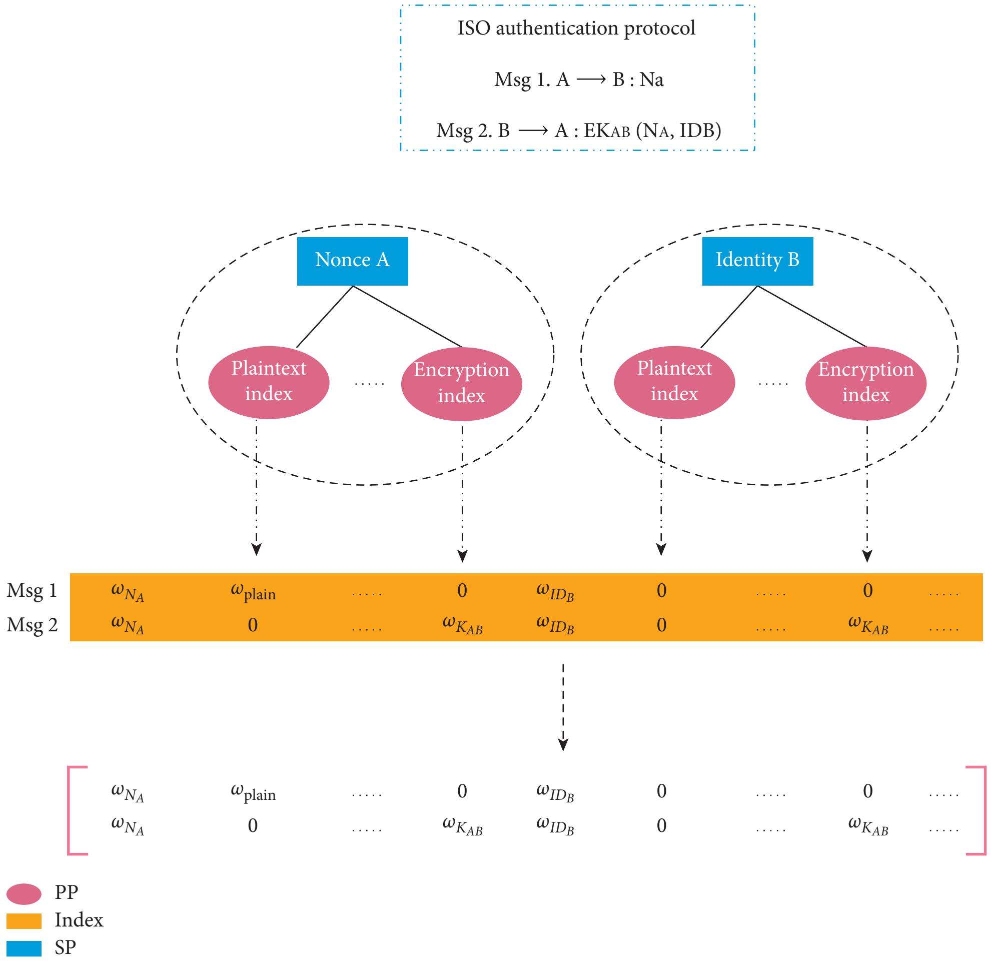

ISO authentication protocol
Msg 1. A ⟶ B : Na
Msg 2. B ⟶ A : EKAB (NA, IDB)
Nonce A
Identity B
Plaintext index
.....
Encryption index
Plaintext index
.....
Encryption index
Msg 1
Msg 2
ω_{N_A} ω_{plain} ..... 0 ω_{ID_B} 0 ..... 0 .....
ω_{N_A} 0 ..... $\omega_{K_{AB}}$ ω_{ID_B} 0 ..... $\omega_{K_{AB}}$ .....
ω_{N_A} ω_{plain} ..... 0 ω_{ID_B} 0 ..... 0 .....
ω_{N_A} 0 ..... $\omega_{K_{AB}}$ ω_{ID_B} 0 ..... $\omega_{K_{AB}}$ .....
PP
Index
SP

FIGURE 9: Conversion process in TLM.

(2) Single-Layer Model (SLM). Similar to TLM model, an empty message vector $m_i = \{sp_{i,1}, sp_{i,2}, \ldots, sp_{i,N}\}$ is predefined, where N is the maximum number of message parameters in the whole dataset. However, in this model, the normalized function f_n is applied to every message parameter $sp_{i,j}$. Thus, every message m_i is represented as follows:

$$m_i = f_n(sp_{i,1}), f_n(sp_{i,2}), \ldots, f_n(sp_{i,N}) = (\lambda_1, \lambda_2, \ldots, \lambda_n). \tag{4}$$

As a result of applying the normalization function, the number of data dimensions is reduced from $N * M$ to N. A schematic of this conversion model is shown in Figure 10.

3.1.2. Our Proposed Model. Despite Ma et al.'s models that consider each message component individually, our proposed model is closer to the real representation of protocols. In this model, every parameter property, pp_i, is represented by a corresponding index, that is, encryption index, signature index, and so on. Then, the indices of the set of message parameters, sp_i, that are in plaintext, encrypted, signed, or hashed together, are put after the plaintext index, encryption index, signature index, and hash index, respectively. The main advantage of this method is that parameters are not modeled individually but alongside other adjacent parameters. Therefore, every message vector would be as follows:

$$m_i = \{pp_{i,1}, sp_{i,1}, sp_{i,2}, \ldots, pp_{i,m}, sp_{i,1}, sp_{i,2}, \ldots\}. \tag{5}$$

As a result, the size of the message vector is reduced to $4 * L$, since the only parameter properties we considered are plaintext, encryption, signature, or hash. Here, L is the maximum number of message parameters that are in plaintext, encrypted, signed, or hashed together. A schematic of this model is depicted in Figure 11.

3.1.3. Comparison of Our Proposed Model with Previous Models. Although Ma et al.'s model could receive remarkable results for the first time, it models each message parameter separately, while in cryptographic protocols, each message parameter is bound to other message parameters. For example, consider the protocol message depicted in Figure 12. In this figure, as shown in red lines, all versions of Ma et al.'s model consider each message component separately. As a result, the machine cannot learn the fact that the set of message parameters are hashed together, while our model considers bound messages together. This is an important point in some attacks such as password guessing attacks, where the adversary exploits the fact that the password is bound to some low entropy parameters (Section 2.3.3). In addition, the dimensions of Ma et al.'s model are so high, which reduces the implementation speed of machine learning models. A comparison of dataset dimensions is shown in Table 1.

3.2. Category. Categories are the labels that we assign to datasets to distinguish the attack in which the protocol is vulnerable to. In Ma et al.'s scheme [18], protocols were labeled based on replay attack and key confirmation. In this paper, we develop more datasets and label them according to more security goals and attacks such as Unknown Key Share attack and strong entity authentication, key freshness, password guessing attack, and key authentication. In this section, after reviewing Ma et al.'s categories, we describe their deficiency and propose our categories to label the datasets.

3.2.1. Review of Ma et al.'s Categories. Ma et al. [18] designed a multiclassification problem to analyze authentication and key agreement protocols with machine learning. Ma et al. suggested that every protocol is either secure or prone to one attack limited to replay attack (Section 2.3.2), lack of key confirmation (Section 2.2.2), or other attacks. Accordingly, Ma et al. associated a category number with every protocol ranging from 1 to 4 (Figure 13). Then, they collected around 500 protocols and divided them according to the attack they are prone to.

3.2.2. Deficiency of Ma et al.'s Categories. Although Ma et al. received remarkable results for the first time, the results are only valid for limited number of protocols. Only around 100 protocols were collected for each category. Limited number of datasets reduces the generalization ability of the analysis tool. Furthermore, most of the protocols are vulnerable to multiple attacks. For example, consider the following protocol in Figure 14.

At the same time, it suffers lack of key confirmation, as neither S nor B is not sure if the other party has received the session key K_{AB}. As a result, the multiclassification problem is not an appropriate framework to analyze protocols with. In the next section, we propose a new framework with a larger number of datasets to analyze security protocols with machine learning.

3.2.3. Our Proposed Categories. Considering the deficiencies of Ma et al.'s categories, we provide more datasets for each category. Further, we design a binary classification problem in which the protocols are either prone to a specific attack or secure against that (Figure 15). In this regard, the following attacks/goals are considered for each problem.

(1) Strong Authentication and Unknown Key Share Attack. As explained in Section 2.1.2, strong entity authentication of A to B is provided if B has a fresh assurance that A has knowledge of B as his/her peer entity. The most common attack that targets this property is UKS attack (Section 2.3.1). For the purpose of analyzing security protocols against this property, we develop around 1000 protocols that are either secure or prone to an attack that violates this property. For example, consider an instance of this dataset in Figure 16. Figure 16(a) shows the SPLICE/AS protocol [28] which is labeled as category 1, since it is prone to the attack presented by Clark and Jacob [29] (Figure 4(b)) and does not achieve strong authentication. An improved version of this protocol

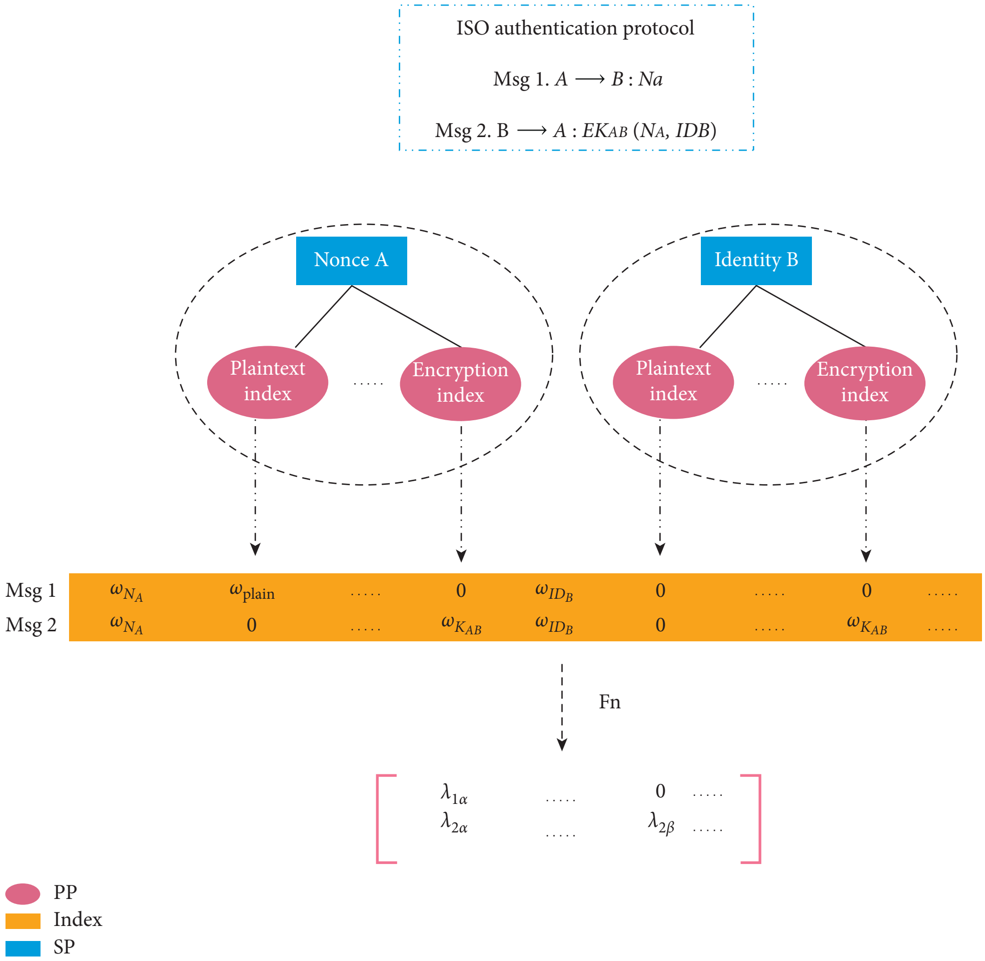


Figure 10: Conversion process in SLM.

is shown in Figure 16(b). As it is secure against this attack, it is labeled as category 0.

As another example, consider the Helsinki protocol [36] in Figure 8(a). As this protocol is vulnerable to UKS attack (Figure 8(b)), it is labeled as category 1 (Figure 17(a)). As suggested by Mitchell et al. [37], a secure version of this protocol is labeled as category 0 (Figure 17(b)).

(2) Key Freshness. As said in Section 2.2.1.1, key freshness is achieved in security protocols if the parties can verify and make sure that the session key they agree with each other is fresh and not replayed from an old session. To analyze security protocols against this property, more than 1500 datasets were developed which are either secure or prone to lack of key freshness. An instance of this dataset is shown in Figure 18. In Figure 18(a), a secure scheme (improved MSR scheme in Figure 6) is shown which is labeled as category 0, while the scheme in Figure 18(b) lacks key freshness, as no freshness value is used to transfer session key K_{AB}.

As another example, consider the key agreement protocol of Figure 19. Here, the session key is a^{xy}. The protocol shown in Figure 19(a) is vulnerable to replay attack which violates the key freshness of the scheme. As the adversary can replay message 2 and convince A to agree on a different session key than party B, thus, it is labeled as category 1. However, the scheme of Figure 19(b) is secure against this attack, as the adversary can no longer replay message 2, since he/she fails to forge the signature of B which includes the freshness parameter a^x.

(3) Key Authentication. According to the definition of Section 2.2.1.1, the key should only be known to A and B and any mutually trusted parties. To analyze security protocols against this property, around 1200 protocols were provided as dataset. Each instance of the dataset is either prone to key authentication or secure against this property. For example, consider Otway-Rees protocol as an instance of dataset in Figure 20(a). As explained in Section 2.2.1.1, this protocol cannot provide key authentication. As a result, it is labeled as category 1. Abadi and Needham's protocol is labeled as category 0 as it achieves key authentication (Figure 20(a)).

(4) Password Guessing Attack. According to the definition of Section 2.3.3, the attacker is able to guess the secret password if it is hashed together with other public parameters. To analyze security protocols against this property, around 1500 protocols were provided as dataset. Each instance of the

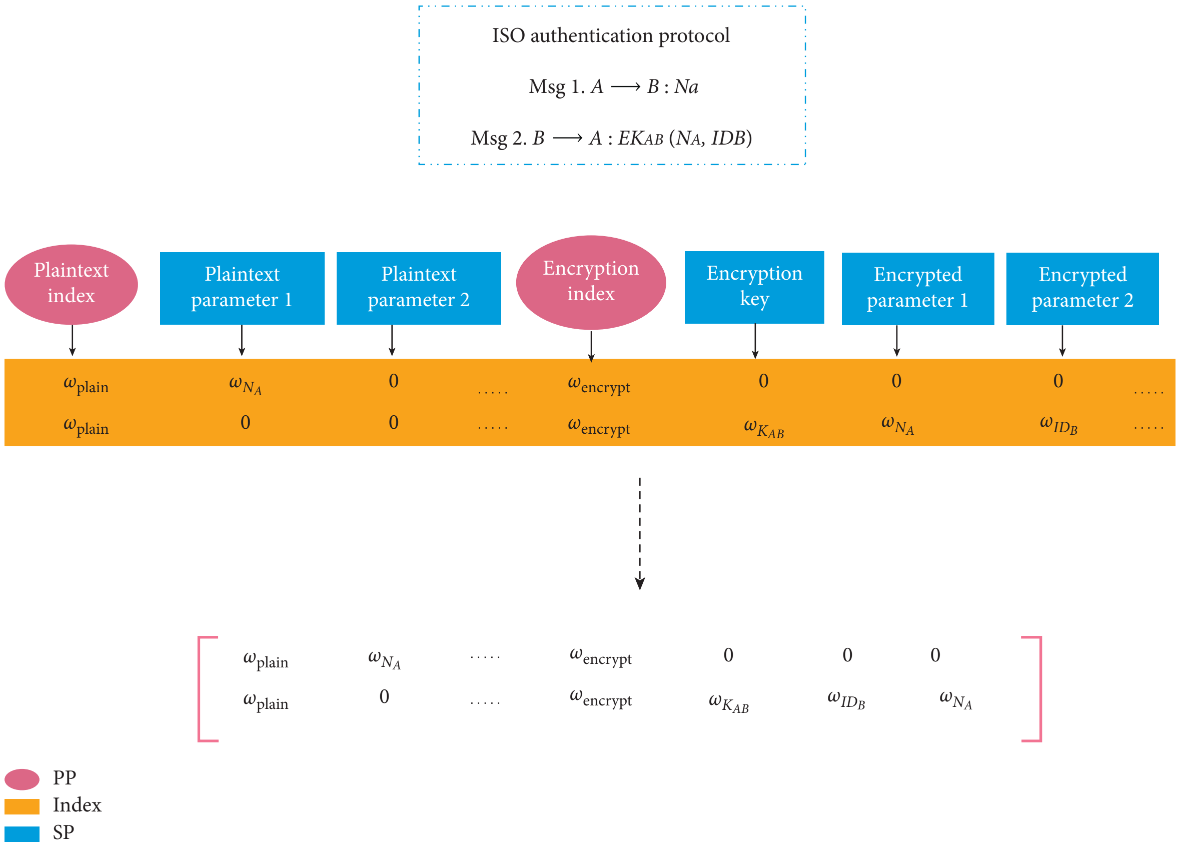

ISO authentication protocol
Msg 1. A ⟶ B : Na
Msg 2. B ⟶ A : EKAB (NA, IDB)
Plaintext index
Plaintext parameter 1
Plaintext parameter 2
Encryption index
Encryption key
Encrypted parameter 1
Encrypted parameter 2
ω_{plain} ω_{N_A} 0 ..... ω_{encrypt} 0 0 0 .....
ω_{plain} 0 0 ..... ω_{encrypt} $\omega_{K_{AB}}$ ω_{N_A} ω_{ID_B} .....
ω_{plain} ω_{N_A} ..... ω_{encrypt} 0 0 0
ω_{plain} 0 ..... ω_{encrypt} $\omega_{K_{AB}}$ ω_{ID_B} ω_{N_A}
PP
Index
SP

Figure 11: Our proposed model.

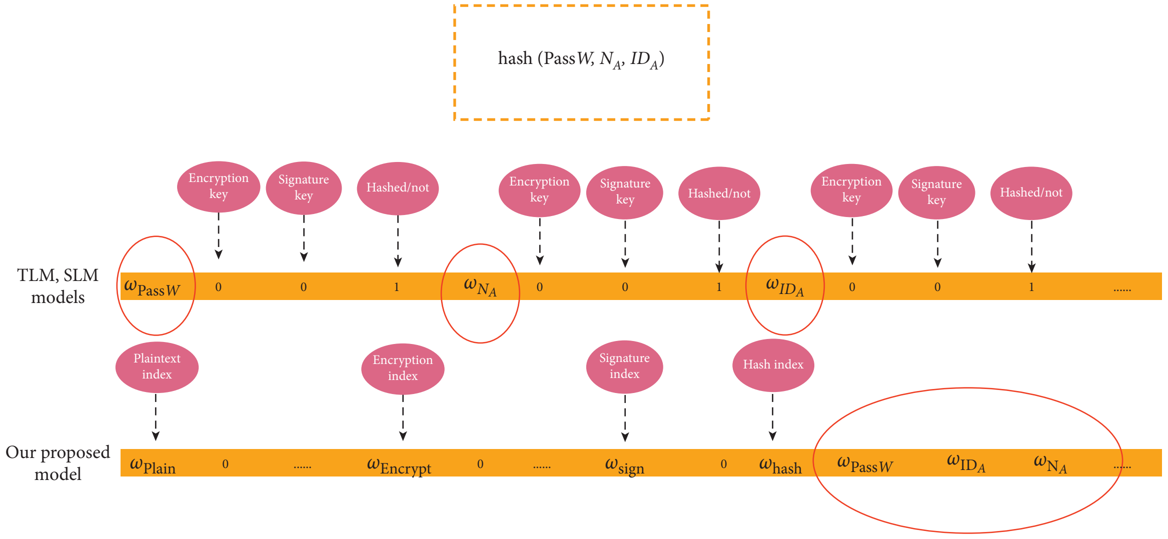

hash (PassW, N_A, ID_A)
Encryption key
Signature key
Hashed/not
Encryption key
Signature key
Hashed/not
Encryption key
Signature key
Hashed/not
TLM, SLM models
$\omega_{\text{Pass}W}$ 0 0 1 ω_{N_A} 0 0 1 ω_{ID_A} 0 0 1 ......
Plaintext index
Encryption index
Signature index
Hash index
Our proposed model
ω_{Plain} 0 ...... ω_{Encrypt} 0 ...... ω_{sign} 0 ω_{hash} $\omega_{\text{Pass}W}$ ω_{ID_A} ω_{N_A} ......

Figure 12: Deficiency of TLM and SLM model in considering message parameters separately.

Table 1: Comparison of dataset dimensions and density.

	TLM	SLM	Our proposed model
Number of features	$N * M$	N	$4 * L$
Data density	High	Low	Low

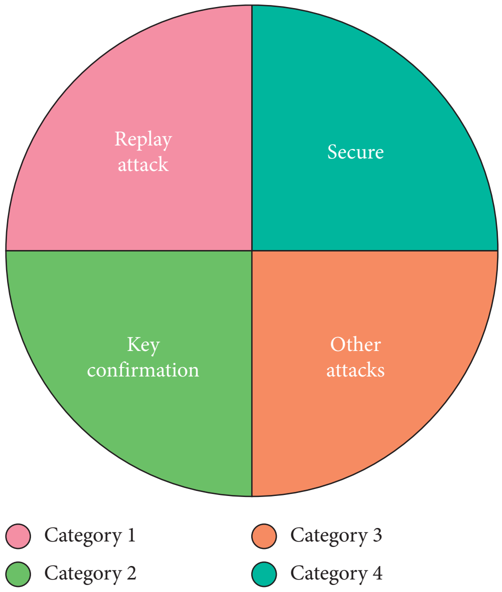


Figure 13: Protocol analysis as a multiclassification problem.

$1.A \longrightarrow B: N_A, ID_A$

$2.B \longrightarrow S: N_A, ID_A, N_B, ID_B$

$3.S \longrightarrow B: E_{K_{AS}}(K_{AB}, ID_B), E_{K_{BS}}(K_{AB}, ID_A)$

$4.B \longrightarrow A: E_{K_{AS}}(K_{AB}, ID_B)$

(a)

$1.A \longrightarrow B: N_A, ID_A$

$1'.A \longrightarrow B: N'_A, ID_A$

$2.B \longrightarrow S: N_A, ID_A, N_B, ID_B$

$2'.B \longrightarrow I_S: N'_A, ID_A, N'_B, ID_B$

$3.S \longrightarrow B: E_{K_{AS}}(K_{AB}, ID_B), E_{K_{BS}}(K_{AB}, ID_A)$

$3'.I_S \longrightarrow B: E_{K_{AS}}(K_{AB}, ID_B), E_{K_{BS}}(K_{AB}, ID_A)$

$4'.B \longrightarrow A: E_{K_{AS}}(K_{AB}, ID_B)$

(b)

Figure 14: An example protocol that suffers both replay attack and lack of key confirmation. (a) A vulnerable security protocol. (b) Replay attack against the protocol.

dataset is either prone to password guessing attack or secure against this attack. For example, consider the Lee-Sohn-Yang-Won password-based protocol as an instance of dataset in Figure 21(a). This protocol is prone to password guessing attack, as the parameters that are hashed together with the password, that is, A and B, are all public and accessible by the adversary. Accordingly, it is labeled as category 1. However, the protocol depicted in Figure 21(b) is secure against password guessing attack thanks to the secret parameter K_{AB}, since the attacker has no way to guess the password and verify its correctness.

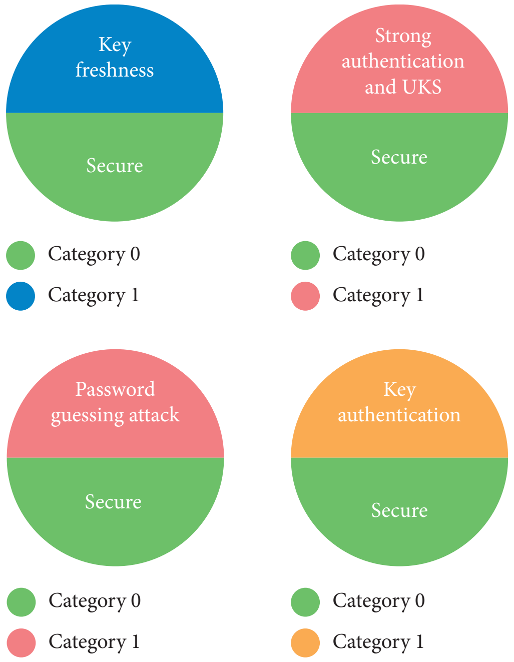


Figure 15: Protocol analysis as a binary classification problem.

4. Experimental Results

In this section, we apply our proposed model along with previous models, namely, TLM and SLM models, to analyze different security properties of authentication and key agreement protocols such as resistance to Unknown Key Share (UKS) attack, key freshness, key authentication, and resistance to password guessing attack. Then, we compare the performance of our proposed model against previous models, namely, TLM and SLM models. The results indicate that our proposed model outperforms the previous models by at least 10–20 percent in all of the machine learning models. In addition, for more complex security properties and attacks such as UKS attack and key authentication, the increase of the dataset size has almost no effect on the classification accuracy, which indicates that these dataset constructions are unable to train the machine. Inspired by the experimental results of Ma et al., we apply XGBoost approach to our classification problem. To improve the accuracy, we modified the default value of the number of gradient boosted trees and maximum depth value of the trees in XGBoost model such that the best results are received with either (10, 3), (10, 15), (20, 15), or (30, 30) where the first component represents the number of gradient boosted trees and the second component represents the maximum depth value of the trees. In addition, to gauge whether deep learning-based approaches are appropriate in this framework, multilayer perceptron (MLP) model is also employed. The results indicate a promising prospect for the integration of deep learning with protocol analysis. The hidden layer size of the MLP model is set to either (15, 15), (20, 20), or (30, 30). In the following, we discuss the experimental results for each analysis, namely, resistance to Unknown Key Share (UKS) attack, key freshness, key authentication, and resistance to password guessing attack.

4.1. Experimental Results of Analyzing UKS Attack. As shown in Figures 22 and 23, in analysis of UKS attack, the classification accuracy of our proposed model rises with the

$1. A \longrightarrow B: A, B, T_A, E_B(N_A), Sig_A(T_A, E_B(N_A))$

$2. B \longrightarrow A: B, A, E_A(B, N_A + 1)$

Category: 1

(a)

$1. A \longrightarrow B: A, B, T_A, E_B(A, N_A), Sig_A(T_A, E_B(A, N_A))$

$2. B \longrightarrow A: B, A, E_A(B, N_A + 1)$

Category: 0

(b)

Figure 16: An instance of dataset in analysis of strong authentication and UKS. (a) SPLICE/AS protocol denoted by category 1. (b) SPLICE/AS improved protocol denoted by category 0.

$1. A \longrightarrow B: E_B(A, K_{AB}, N_A)$

$2. B \longrightarrow A: E_B(K_{BA}, N_A, N_B)$

$3. A \longrightarrow B: N_B$

Category: 1

(a)

$1. A \longrightarrow B: E_B(A, K_{AB}, N_A)$

$2. B \longrightarrow A: E_B(B, K_{BA}, N_A, N_B)$

$3. A \longrightarrow B: N_B$

Category: 0

(b)

Figure 17: Another instance of dataset in analysis of strong authentication and UKS. (a) Helsinki protocol denoted by category 1. (b) Mitchell et al.'s protocol denoted by category 0.

$1. A \longrightarrow B: ID_A, Cert_A, N_A$

$2. B \longrightarrow A: E_A(K_{AB}), E_{K_{AB}}(ID_A, Cert_B)$

Category: 1

(a)

$1. A \longrightarrow B: ID_A, Cert_A, N_A$

$2. B \longrightarrow A: E_A(K_{AB}), E_{K_{AB}}(ID_A, N_A, Cert_B)$

Category: 0

(b)

Figure 18: An instance of dataset in analysis of key freshness. (a) MSR scheme without key freshness denoted by category 1. (b) Improved MSR scheme with key freshness denoted by category 0.

$1. A \longrightarrow B: a^x, Sig_A(ID_A, ID_B, a^x)$

$2. B \longrightarrow A: a^y, Sig_B(ID_A, ID_B, a^x, a^y)$

Category: 0

(a)

$1. A \longrightarrow B: a^x, Sig_A(ID_A, ID_B, a^x)$

$2. B \longrightarrow A: a^y, Sig_B(ID_A, ID_B, a^y)$

Category: 1

(b)

Figure 19: Another instance of dataset in analysis of key freshness. (a) A secure key agreement protocol with key freshness denoted by category 0. (b) A vulnerable key agreement protocol without key freshness denoted by category 1.

$1. A \longrightarrow B: A, B, \{N_A, A, B\} K_{AS}$

$2. B \longrightarrow S: A, B, \{N_A, A, B\} K_{AS}, \{N_B, A, B\} K_{BS}$

$2. S \longrightarrow B \{N_A, K_{AB}\} K_{AS}, \{N_B, K_{AB}\} K_{BS}$

$4. B \longrightarrow A: \{N_A, K_{AB}\} K_{AS}$

Category: 1

(a)

$1. A \longrightarrow B: A, B, N_A$

$2. B \longrightarrow S: A, B, N_A, N_B$

$3. S \longrightarrow B: \{N_A, A, B, K_{AB}\} K_{AS}, \{N_B, A, B, K_{AB}\} K_{BS}$

$4. B \longrightarrow A: \{N_A, A, B, K_{AB}\} K_{AS}$

Category: 0

(b)

Figure 20: An instance of dataset in analysis of key authentication. (a) Otway-Rees's protocol without key authentication denoted by category 1. (b) Abadi and Needham's protocol with key authentication denoted by category 0.

$1. A \longrightarrow B: A, a^x, \text{hash}(A, B, \text{Password}_A)$

$2. B \longrightarrow A: B, a^y, \text{hash}(K_{AB}, a^y, A, B, \text{Password}_A)$

$3. B \longrightarrow A: \text{hash}(K_{AB}, a^y)$

Category: 1

(a)

$1. A \longrightarrow B: A, a^x, \text{hash}(A, B, K_{AB}, \text{Password}_A)$

$2. B \longrightarrow A: B, a^y, \text{hash}(K_{AB}, a^y, A, B, \text{Password}_A)$

$3. B \longrightarrow A: \text{hash}(K_{AB}, a^y)$

Category: 0

(b)

Figure 21: An instance of dataset in analysis of password guessing attack. (a) Lee-Sohn-Yang-Won password-based protocol prone to password guessing attack denoted by category 1. (b) Improved Lee-Sohn-Yang-Won password-based protocol resistant to password guessing attack denoted by category 0.

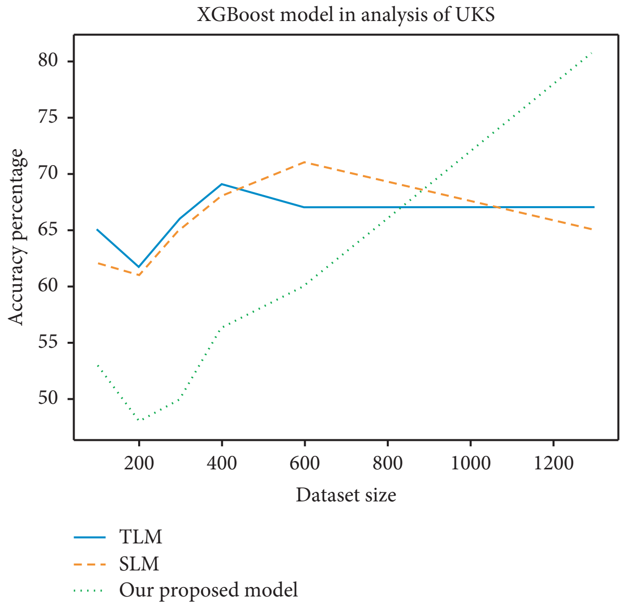


Figure 22: Analysis of UKS with XGBoost model.

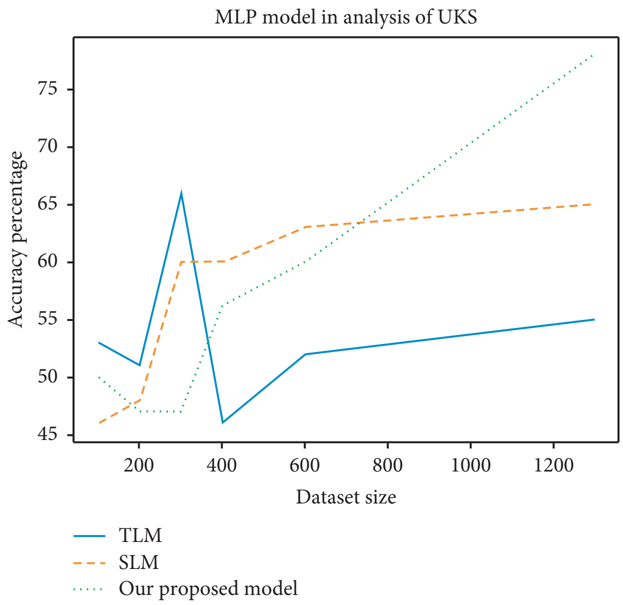


Figure 23: Analysis of UKS with the MLP model.

increase of the dataset size, as opposed to the other two models, namely, TLM and SLM models, in which the dataset size has almost no effect on the classification accuracy. For a large number of datasets, that is, the number of protocols is 1300, the classification accuracy reaches over 80% which is 20% higher than the other two models. The higher classification accuracy of TLM and SLM models for a low number of protocols, that is, 100–600, may be tempting to conclude that our model is unable to train the machine. However, with the increase of the number of datasets, the accuracy of the TLM and SLM models either decreases or remains constant. Further, the performance of TLM model is extremely fluctuating in MLP algorithm.

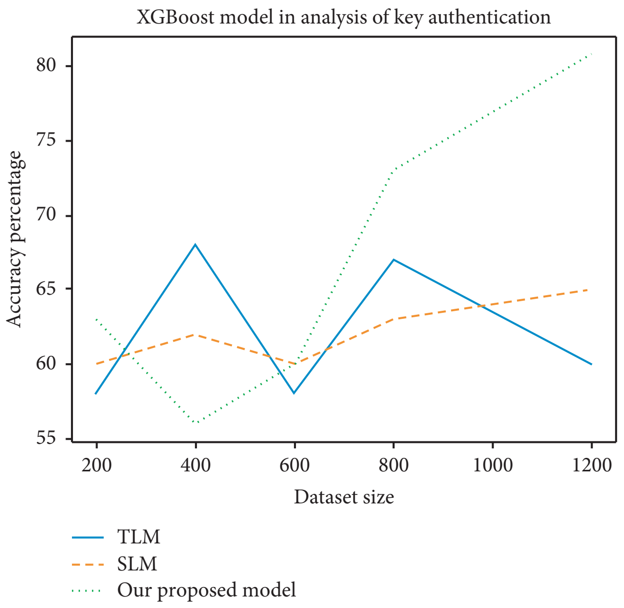


Figure 24: Analysis of key authentication with the XGBoost model.

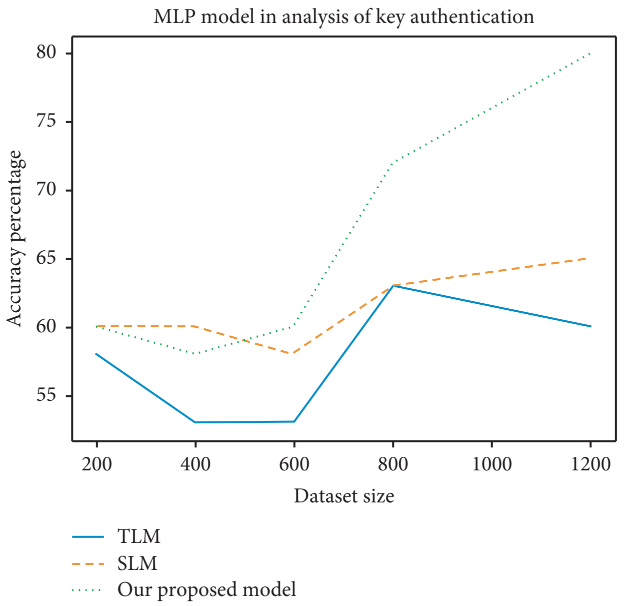


Figure 25: Analysis of key authentication with the MLP model.

4.2. Experimental Results of Analyzing Key Authentication. In the analysis of key authentication, TLM and SLM dataset constructions fail to train the machine. According to Figures 24 and 25, increase of the dataset size not only has no effect on the classification accuracy but also decreases the accuracy in case of TLM model. Meanwhile, the classification accuracy of our proposed model rises with the increase of the dataset size and reaches over 80% for 1200 protocols which is 15–20% higher than the other two models.

4.3. Experimental Results of Analyzing Key Freshness. Key freshness is a simpler security property compared to UKS attack and key authentication, as it affects only a few

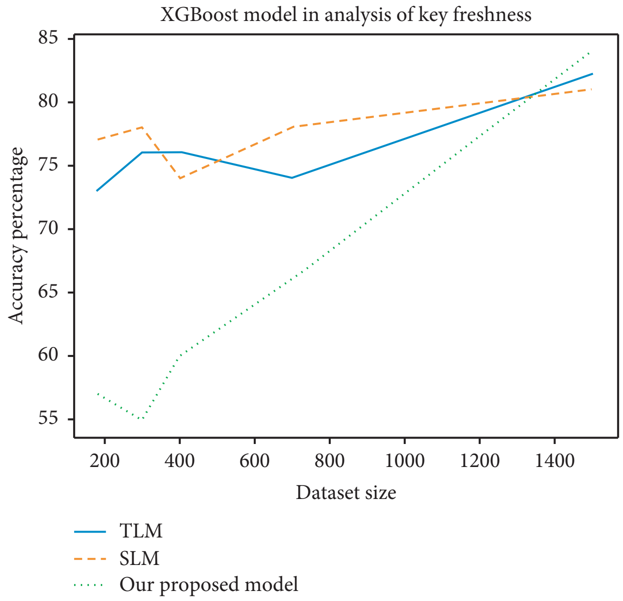


FIGURE 26: Analysis of key freshness with the XGBoost model.

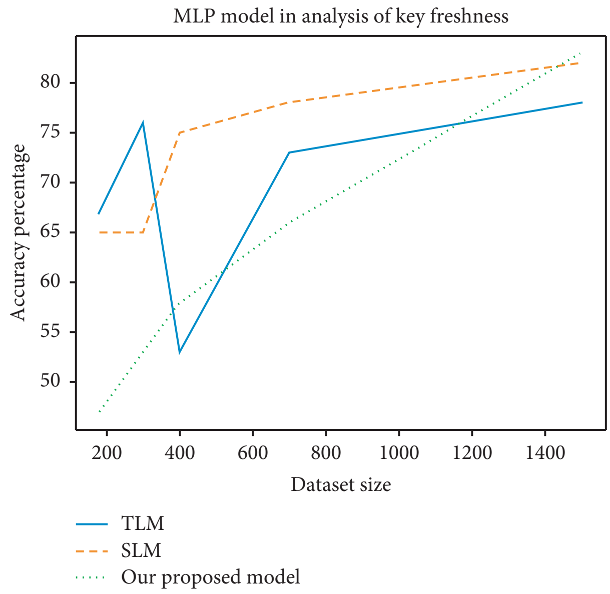


FIGURE 27: Analysis of key freshness with the MLP model.

parameters such as timestamp and nonce. As a result, the classification accuracy of TLM and SLM models still improves with the increase of the dataset size. However, as shown in Figures 26 and 27, the rate of the increase of classification accuracy of our proposed model is almost three times more than the TLM and SLM models. Similar to UKS attack, the performance of TLM model is so fluctuating in MLP algorithm. For a large number of datasets, that is, the number of protocols is 1500, the classification accuracy reaches over 80% which is 10% higher than the TLM and SLM models.

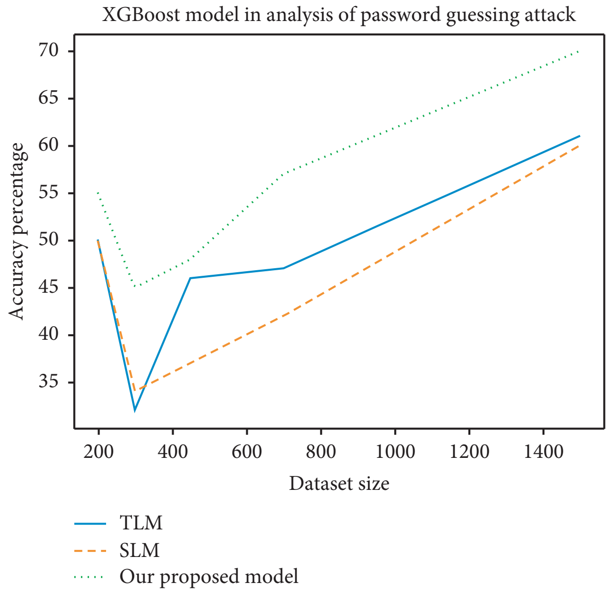


FIGURE 28: Analysis of password guessing attack with XGBoost model.

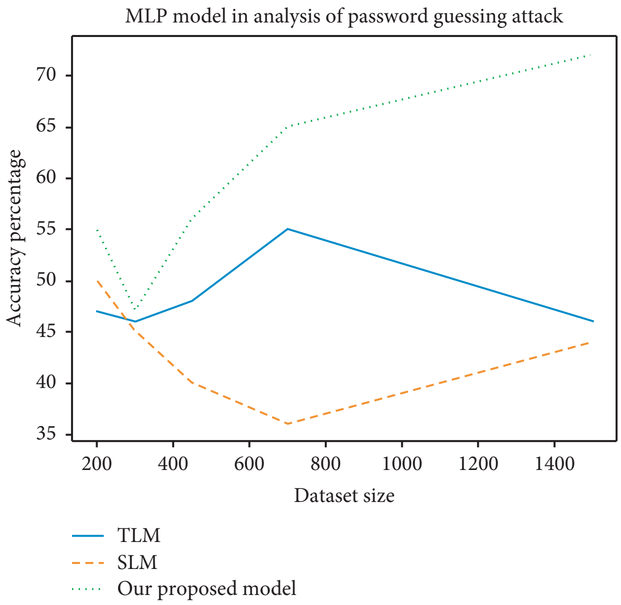


FIGURE 29: Analysis of password guessing attack with MLP model.

4.4. Experimental Results of Analyzing Password Guessing Attack. As depicted in Figures 28 and 29, in MLP approach, the classification accuracy of TLM and SLM dataset constructions extremely decreases with the increase of the dataset size. Although XGBoost solver is able to train the machine using the TLM and SLM dataset constructions, its classification accuracy is still much lower than our proposed dataset construction. For a large number of datasets, that is, the number of protocols is 1200, the classification accuracy reaches 60% which is still 10% lower than our proposed dataset construction.

5. Conclusion, Limitation, and Future Work

Considering the difficulties of formal protocol analysis approaches, researchers have begun to apply machine learning in this area. In this paper, we investigated Ma et al.'s framework as the first attempt in applying machine learning to protocol security analysis. The main limitation of Ma et al.'s framework is that it only considers replay attack and key confirmation. Further, it exploits multiclassification as a security framework for such analysis in which every protocol or dataset is either secure or prone to a security attack such as replay attack, key confirmation, or other attacks. However, we show that multiclassification problem is not an appropriate framework. As a result, we propose binary classification in which every protocol is either prone to a specific attack or secure against that. In addition, more security properties and attacks are considered to analyze protocols against, such as strong authentication and Unknown Key Share (UKS) attack, key freshness, key authentication, and password guessing attack. Despite previous dataset construction models suggested by Ma et al., in our proposed dataset construction model, the classification accuracy increases with the increase of the dataset size, which represents the fact that our proposed dataset construction model is capable of training the machine to analyze security attacks and properties. The most evident limitation of our work is the fact that the accuracy of our scheme is only 80%. However, for a practical analysis scheme, we need an ideal analysis scheme with an accuracy of 100%. As a future work, more datasets can be provided to reach an ideal analysis scheme. In addition, more complex security properties can be analyzed using machine learning techniques such as pretraining and few-shot learning.

Data Availability

Supplementary codes and datasets are available at https://github.com/zahednejad/protocol-analysis-with-machinelearning.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Guangzhou University and the Guangdong Provincial Natural Science Foundation, under Grant no.2018A030310071.

References

[1] C. Boyd, A. Mathuria, and D. Stebila, *Protocols for Authentication and Key Establishment*, Vol. 1, Springer, Heidelberg, Germany, 2003.

[2] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol*, 2008, https://www.hjp.at/doc/rfc/rfc5246.html.

[3] B. Blanchet, "Modeling and verifying security protocols with the applied pi calculus and ProVerif," *Foundations and Trends® in Privacy and Security*, vol. 1, no. 1-2, pp. 1–135, 2016.

[4] C. J. Cremers, "The Scyther Tool: verification, falsification, and analysis of security protocols," in *Proceedings of the International Conference on Computer Aided Verification*, Springer, Berlin, Germany, pp. 414–418, July 2008.

[5] A. Armando, D. Basin, Y. Boichut et al., "The AVISPA tool for the automated validation of internet security protocols and applications," in *Proceedings of the International Conference on Computer Aided Verification*, Springer, Berlin, Germany, pp. 281–285, 2005, July.

[6] B. Blanchet, "CryptoVerif: computationally sound mechanized prover for cryptographic protocols," *Formal Protocol Verification*, vol. 117, p. 156, 2007.

[7] A. Bleeker and L. Meertens, "A semantics for BAN logic," in *Proceedings of the DIMACS Workshop on Design and Formal Verification of Security Protocols*, New Brunswick, NJ, USA, September 1997.

[8] I. Cervesato, "The Dolev-Yao intruder is the most powerful attacker," in *Proceedings of the 16th Annual Symposium on Logic in Computer Science—LICS*, vol. 1, Boston, MA, USA, June 2001.

[9] F. J. T. Fábrega, J. C. Herzog, and J. D. Guttman, "Strand spaces: why is a security protocol correct?" in *Proceedings of the 1998 IEEE Symposium on Security and Privacy*, IEEE, Oakland, CA, USA, pp. 160–171, 1998 May.

[10] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking cryptographic implementations using deep learning techniques," in *Proceedings of the International Conference on Security, Privacy, and Applied Cryptography Engineering*, pp. 3–26, Kharagpur, India, 2016.

[11] S. Picek, I. Petros, and S. Jaehun, "On the performance of convolutional neural networks for side-channel analysis," in *Proceedings of the International Conference on Security, Privacy, and Applied Cryptography Engineering*, Springer, Cham, Switzerland, pp. 157–176, 2018.

[12] A. Gohr, "Improving attacks on round-reduced speck32/64 using deep learning," in *Proceedings of the Annual International Cryptology Conference*, Springer, Cham, Switzerland, 2019.

[13] J. So, "Deep learning-based cryptanalysis of lightweight block ciphers," *Security and Communication Networks*, vol. 2020, Article ID 3701067, 11 pages, 2020.

[14] T. Li, X. Liu, Z. Qin, and X. Zhang, "Formal analysis for security of Otway–Rees protocol with ban logic," in *Proceedings of the First International Workshop on Database Technology and Applications*, pp. 590–593, Hubei, China, 2009.

[15] K. Liu, J. Ye, and Y. Wang, "The security analysis on Otway–Rees protocol based on ban logic," in *Proceedings of the Fourth International Conference on Computational and Information Sciences*, pp. 341–344, Chongqing, China, 2012.

[16] S. Pan, T. Morris, and U. Adhikari, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 3104–3113, 2015.

[17] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," in *Proceedings of the International Conference on Computing, Networking and Communications*, pp. 797–801, Honolulu, Hawaii, 2014.

[18] Z. Ma, Y. Liu, Z. Wang, H. Ge, and M. Zhao, "A machine learning-based scheme for the security analysis of

authentication and key agreement protocols," *Neural Computing and Applications*, pp. 1–13, 2018.

[19] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. Tang, "Xgboost: extreme gradient boosting," *R Package Version*, vol. 4-2, pp. 1–4, 2015.

[20] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-The Advanced Encryption Standard*, Springer Science Business Media, Berlin, Germany, 2013.

[21] D. Coppersmith, "The data encryption standard (DES) and its strength against attacks," *IBM Journal of Research and Development*, vol. 38, no. 3, pp. 243–250, 1994.

[22] ISO Information, "Technology-security techniques-entity authentication-part 2: mechanisms using symmetric encipherment algorithms," *International Standard*, 1999.

[23] B. Zahednejad, M. Azizi, and M. Pournaghi, "A novel and efficient privacy preserving TETRA authentication protocol,"in Proceedings of the 2017 14th International ISC (IranianSociety of Cryptology) Conference on Information Security andCryptology (ISCISC), IEEE, pp. 125–132, Shiraz, Iran, 2017, September.

[24] S. M. Pournaghi, B. Zahednejad, M. Bayat, and Y. Farjami, "NECPPA: a novel and efficient conditional privacy-preserving authentication scheme for VANET," *Computer Networks*, vol. 134, pp. 78–92, 2018.

[25] A. Akbarzadeh, M. Bayat, B. Zahednejad, A. Payandeh, and M. R. Aref, "A lightweight hierarchical authentication scheme for internet of things," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 7, pp. 2607–2619, 2019.

[26] ISO Open, "Systems interconnection-basic reference model-part 2: security architecture," ISO Open, Geneva, Switzerland, 1989.

[27] T. Matsumoto, Y. Takashima, and H. Imai, "On seeking smart public-key-distribution systems," *Transactions of the IECE of Japan*, vol. E69, no. 2, pp. 99–106, 1986.

[28] S. Yamaguchi, K. Okayama, and H. Miyahara, "Design and implementation of an (authentication system in WIDE internet environment," in *Proceedings of the IEEE Region 10 Conference on Computer and Communications Systems*, pp. 653–657, Hong Kong, 1990.

[29] J. Clark and J. Jacob, "On the security of recent protocols," *Information Processing Letters*, vol. 56, no. 3, pp. 151–155, 1995.

[30] J. Alfred, C. Paul, and A. Scott, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, USA, 1997.

[31] L. Gong, "Variations on the themes of message freshness and replay," in *Proceedings of the 6th IEEE Computer Security Foundations Workshop*, IEEE Computer Society Press, Franconia, NH, USA, pp. 131–136, 1993.

[32] D. E. Denning and G. M. Sacco, "Timestamps in key distribution protocols," *Communications of the ACM*, vol. 24, no. 8, pp. 533–536, 1981.

[33] D. Gollmann, "Authentication by correspondence," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 1, pp. 88–95, 2003.

[34] C. Boyd and W. Mao, "On a limitation of BAN logic," in *Advances in Cryptology-Eurocrypt '93*, T. Helleseth, Ed., vol. 765, pp. 240–247, Springer-Verlag, Berlin, Germany, 1994.

[35] S. Blake-Wilson and A. Menezes, "Entity authentication and authenticated key transport protocols employing asymmetric techniques," in *Security Protocols 5th International Workshop*, B. Christianson, Ed., vol. 1361, p. 137, Springer-Verlag, Berlin, Germany, 1998.

[36] G. Horng and C.-K. Hsu, "Weakness in the Helsinki protocol," *Electronics Letters*, vol. 34, no. 4, pp. 354-355, 1998.

[37] J. Chris and C. Y. Yeob, "Fixing a problem in the Helsinki protocol," *ACM Operating Systems Review*, vol. 32, no. 4, pp. 21–24, 1998.

[38] P. Syverson, "A taxonomy of replay attacks," in *Proceedings of the 7th IEEE Computer Security Foundations Workshop*, IEEE Computer Society Press, Franconia, NH, USA, pp. 187–191, 1994.

[39] R. Graham, "How hackers will crack your password," 2009, http://www.darkreading.com/hacked-off/how-hackers-will-crack-your-password/227700892.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8823952, 11 pages
https://doi.org/10.1155/2020/8823952

Research Article

Qualitative Analysis of Commercial Services in MEC as Phased-Mission Systems

Yuhuan Gong[1] **and Yuchang Mo**[2]

[1]*School of Marxism, Huaqiao University, Quanzhou, China*
[2]*Fujian Province University Key Laboratory of Computational Science, School of Mathematical Sciences, Huaqiao University, Quanzhou, China*

Correspondence should be addressed to Yuhuan Gong; myc@hqu.edu.cn

Received 5 August 2020; Revised 10 September 2020; Accepted 23 September 2020; Published 12 October 2020

Academic Editor: Xiaolong Xu

Currently, mobile edge computing (MEC) is one of the most popular techniques used to respond to real-time services from a wide range of mobile terminals. Compared with single-phase systems, commercial services in MEC can be modeled as phased-mission systems (PMS) and are much more complex, because of the dependencies across the phases. Over the past decade, researchers have proposed a set of new algorithms based on BDD for fault tree analysis of a wide range of PMS with various mission requirements and failure behaviors. The analysis to be performed on a fault tree can be either qualitative or quantitative. For the quantitative fault tree analysis of PMS by means of BDD, much work has been conducted. However, for the qualitative fault tree analysis of PMS by means of BDD, no much related work can be found. In this paper, we have presented some efficient methods to calculate the MCS encoding by a PMS BDD. Firstly, three kinds of redundancy relations-inclusive relation, internal-implication relation, and external-implication relation-within the cut set are identified, which prevent the cut set from being minimal cut set. Then, three BDD operations, IncRed, InImpRed, and ExImpRed, are developed, respectively, for the elimination of these redundancy relations. Using some proper combinations of these operations, MCS can be calculated correctly. As an illustration, some experimental results on a benchmark MEC system are given.

1. Introduction

Currently, mobile edge computing (MEC) is one of the most popular techniques used to respond to real-time services from a wide range of mobile terminals [1–4]. Edge computing is a distributed computing topology where information processing is placed closer to the things or people that produce and/or consume that information. It means that MEC provides cloud-computing capabilities at the edge of the mobile network in close proximity to mobile subscribers; it is also considered as one of the key pillars for meeting the demanding KPIs of 5G.

Mobile edge computing integrates cloud computing (CC) into mobile networks, prolonging the battery life of mobile users (MUs). However, this mode may cause significant mission complexity. Compared with single-phase systems, commercial services in MEC can be modeled as phased-mission systems (PMS). A MEC PMS is defined as a commercial service, which is subject to multiple, consecutive, nonoverlapping phases of operation. During each phase, it has to accomplish a specified task. Thus, the MEC configuration, failure criterion, and/or failure behavior can change from phase to phase [5–8].

Much early work has been conducted on the fault tree analysis of PMS. Qualitative analysis of commercial services in MEC as phased-mission systems is much more complex, because of the dependencies across the phases. For instance, the state of a MEC node at the beginning of a new phase is identical to its state at the end of the previous phase [9–11]. Over the past decade, researchers have proposed a set of new algorithms based on the binary decision diagram (BDD) for fault tree analysis of a wide range of PMS with various mission requirements and failure behaviors [12–15]. Due to the nature of the BDD, cancellation of common components among the phases can be combined with the BDD generation, without additional operations, and the sum of disjoint products (SDP) can be implicitly represented by the final PMS BDD. Several experiments show that BDD-based algorithm is more efficient than the algorithm based on SDP, in both computation time and storage space; this efficiency allows the study of some practical, large MEC PMS [16–18].

The analysis to be performed on a fault tree can be either qualitative or quantitative. Qualitative analysis involves calculating the minimal cut set (MCS), that is, listing all possible smallest combinations of basic events, which cause the top event. Quantitative analysis, on the other hand, involves calculating the probability of the top event occurring from the probabilities of the basic events. For the qualitative fault tree analysis of MEC PMS by means of BDD, no much related work can be found. In this paper, we focus on this line of research, that is, calculating the MCS encoding by a PMS BDD. The major contributions of our work are the following ones:

(1) Different kinds of redundancy relations within the cut set encoding by a PMS BDD are identified, and it is these relations that prevent the cut set from being minimal cut set.

(2) Two BDD operations are proposed to eliminate these redundancy relations from a PMS BDD. One operation can only eliminate the external-implication relations relating to one component, and the other operation eliminates all the inclusive relations from the cut set.

(3) MCS calculation method is developed by combining the proposed two BDD operations for a PMS BDD with forward ordering.

The remainder of the paper is organized as follows. Section 2 introduces the fundamentals of fault tree and BDD for the qualitative analysis of MEC PMS. Section 3 proposes three kinds of redundancy relations within a PMS BDD, several BDD operations eliminating these redundancy relations, and MCS calculation methods based on these BDD operations. Section 4 gives some experimental results on a benchmark MEC PMS. Last, conclusions are given in Section 5.

2. Preliminaries

2.1. Fault Tree. Fault tree analysis is an important technique for reliability and safety analysis. Bell Telephone Laboratories developed the concept in 1962 for the US Air Force for use with the Minuteman system. It was later adopted and extensively applied by the Boeing Company.

Fault tree diagrams are logic block diagrams that display the state of a system (top event) in terms of the states of its components (basic events). It uses a graphic "model" of the pathways within a system that can lead to a foreseeable, undesirable loss event (or a failure). Fault trees are built using gates and events. The two most commonly used gates in a fault tree are the AND and OR gates. If both events need to occur to cause the top event to occur, they are connected by an AND gate. Alternatively, if the occurrence of either event causes the top event to occur, then these events are connected using an OR gate. Notice that NOT gate is not considered in this paper. Thus, the fault trees analyzed in the following sections are coherent fault trees.

For MEC PMS, there are two kinds of fault tree models: phase-level fault tree, which represents the causal chain between component failure and phase failure, and system-level fault tree, which represents the causal chain between phase failure and service failure. As an illustration example, Figure 1 depicts a small MEC PMS fault tree.

For the purpose of this paper, MEC PMS fault trees are essentially considered as Boolean functions, that is, terms inductively built over the two constants 0 and 1, a set of variables X, and usual logical connectives ∧ (AND) ∨ (OR). The Boolean function F associated with the fault tree in Figure 1 is $F = (A_1 \vee B_1 \vee C_1) \vee (A_2 \vee (B_2 \wedge C_2)) \vee (A_3 \wedge B_3 \wedge C_3)$.

2.2. BDD. BDD is a compact encoding of the truth tables of Boolean function [19]. The BDD representation is based on the Shannon decomposition. Let F be a Boolean function that depends on the variable x; then the following equality holds:

$$\begin{aligned} F &= (x \wedge F[x \leftarrow 1]) \vee (x \wedge F[x \leftarrow 0]) \\ &= \mathrm{ite}(x, F[x \leftarrow 1], F[x \leftarrow 0]). \end{aligned} \quad (1)$$

BDD has two sink nodes, labeled 0 and 1, representing the two corresponding constants 0 and 1. Each nonsink node is labeled with a Boolean variable x and has two outgoing edges that represent the two corresponding expressions in the Shannon decomposition. These two edges are called E-edge (or 0-edge) and T-edge (or 1-edge), respectively. The node linked by the T-edge represents the Boolean expression $F[x \leftarrow 1]$; E-edge represents the Boolean expression $F[x \leftarrow 0]$. Thus, each nonsink node in a BDD encodes an ite format.

An ordered BDD is a BDD with the constraint that the variables are ordered and every source-to-sink path in the ordered BDD visits the variables in ascending order. A reduced ordered BDD is an ordered BDD where each node represents a distinct Boolean expression.

2.3. Variable Ordering. A variable in a PMS BDD indicates the component that the variable belongs to and the phase in which the component exists. For example, C_i is the state indicator variable of component C at phase i. Variable ordering can be generated in the following steps: component-level ordering and ordering variables on phases.

(1) Component-level ordering: the components in PMS are s-independent from each other. Therefore, the ordering heuristics for ordinary BDD can be applied to component-level ordering for PMS BDD.

(2) Phase ordering: there are two classes of phase ordering: forward ordering and backward ordering. In forward ordering, the variable order is the same as

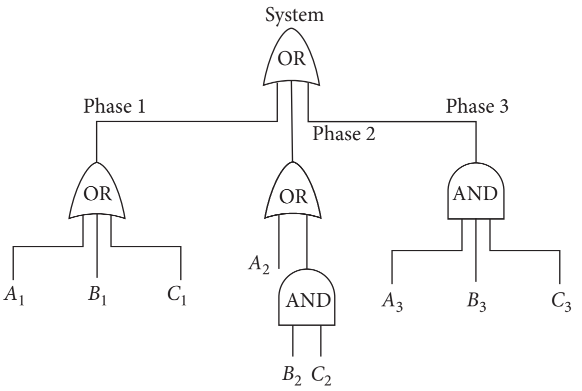


Figure 1: An example MEC PMS fault tree.

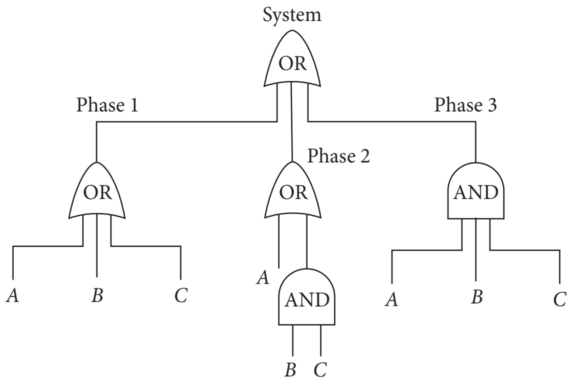


Figure 2: The "do not care" form of the tree in Figure 1.

the phase order. In backward ordering, the variable order is the reverse of the phase order.

To illustrate the above ordering method, the PMS in Figure 1 can be used. Applying DFLM to the fault tree in "do not care" form shown in Figure 2, where phase indexes are unconsidered, the component-level ordering is $A < B < C$. Extending this component-level ordering over different phase indexes, the forward DFLM ordering is $A_1 < A_2 < A_3 < B_1 < B_2 < B_3 < C_1 < C_2 < C_3$, and the backward DFLM ordering is $A_3 < A_2 < A_1 < B_3 < B_2 < B_1 < C_3 < C_2 < C_1$.

2.4. BDD Generation. In order to compute the BDD associated with a Boolean function *F*, the following principle is applied:

(1) If *F* is a constant, then one associates with *F* the corresponding sink node 0 or 1.

(2) If $F = x$, where x is a variable, then one associates with *F* the BDD ite (x, 1, 0).

(3) Finally, if $F = G <> H$, where <> is the binary connective $\wedge$ or $\vee$ and *G* and *H* are functions, then one computes the BDDs associated with *G* and *H* and then performs the operation <> on these two BDDs.

Given two Boolean functions *G* and *H* encoded by the BDDs $G = \text{ite}\ (x, G_1, G_0)$ and $H = \text{ite}\ (y, H_1, H_0)$, it is possible to compute directly on *G* and *H* any logical operation between *G* and *H* by means of the following calculation:

$$G \diamond H = \text{ite}\left(x, G_1, G_0\right) \diamond \text{ite}\left(y, H_1, H_0\right) = \begin{cases} \text{ite}\left(x, G_1 \diamond H_1, G_0 \diamond H_0\right), & x = y, \\ \text{ite}\left(x, G_1 \diamond H, G_0 \diamond H\right), & x < y, \\ \text{ite}\left(y, G \diamond H_1, G \diamond H_0\right), & x > y. \end{cases} \quad (2)$$

Compared with the single-phase system, BDD generation of PMS is much more complex because of the dependencies of component states across the phases. Use *G* and H to represent phase BDD of phase *i* and phase *j*, respectively $(i < j)$. When expanded with regard to x_i and x_j, they, respectively, can be written as $G = \text{ite}\ (x_i, G_1, G_0)$; $H = \text{ite}\ (x_j, H_1, H_0)$.

Following special BDD operation, to deal with the cross-phase dependencies associated with the operation <> on *G* and *H*,

$$G \diamond H = \text{ite}\left(x_i, G_1, G_0\right) \diamond \text{ite}\left(x_j, H_1, H_0\right) = \begin{cases} \text{ite}\left(x_i, G_1 \diamond H_1, G_0 \diamond H_0\right), & x_i = x_j, \\ \text{ite}\left(x_j, G \diamond H_1, G_0 \diamond H_0\right), & x_j < x_i, \\ \text{ite}\left(x_i, G_1 \diamond H_1, G_0 \diamond H\right), & x_i < x_j. \end{cases} \quad (3)$$

With the help of BDD operations (2) and (3), PMS BDD can be generated from PMS fault tree. When different orderings are used, the sizes of PMS BDD are different. As shown in Figure 3, the size of BDD with forward DFLM ordering is 15.

3. Qualitative Analysis

3.1. Cut Set and Minimal Cut Set. In order to introduce formally the notion of minimal cut set, we need the following definitions.

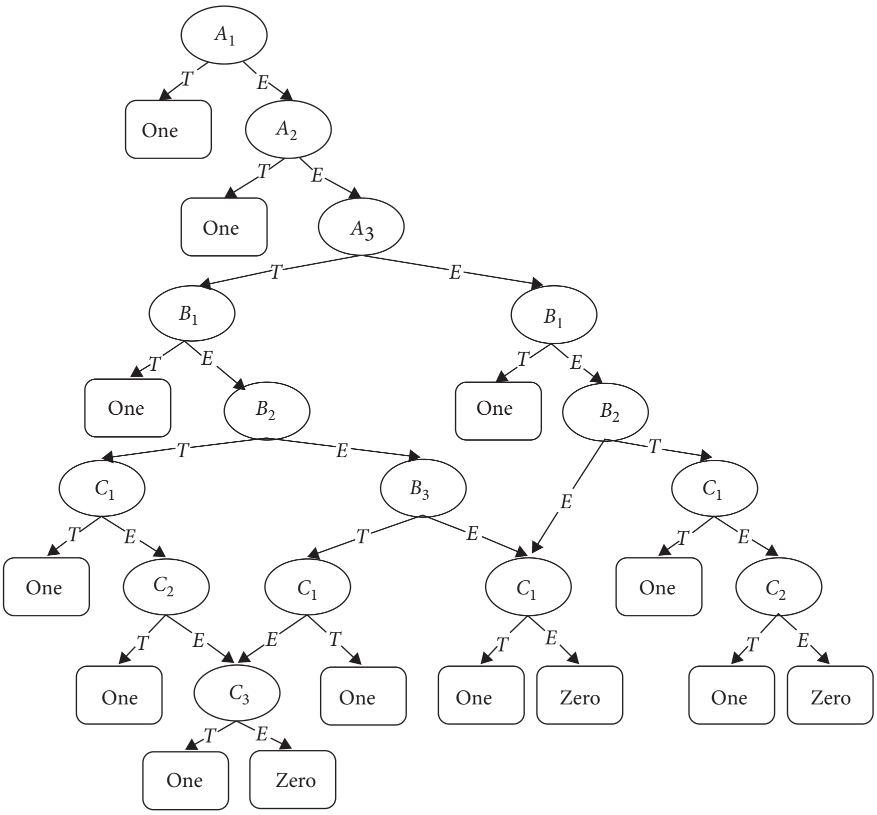


FIGURE 3: PMS BDD with forward DFLM ordering.

Definition 1. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of Boolean variables. An assignment of X is a mapping from X into $\{0, 1\}$.

Assignment and subset of X are equivalent objects. A unique assignment σ corresponds to a subset of X, $X(\sigma)$, where

$$\sigma(x) = 1 \text{iff} x \in X(\sigma). \tag{4}$$

Definition 2. Let σ be an assignment of X and F be a monotonic Boolean function on X. $X(\sigma)$ is a cut if $F(\sigma) = 1$.

The set of cuts of F is denoted by $\mathrm{CS}[F]$.

Definition 3. For any $\rho \in \mathrm{CS}[F]$, ρ is a minimal cut if for any assignment σ,

$$X(\sigma') \subset \rho \mathit{implies} \quad \mathit{that} \quad F(\sigma') = 0. \tag{5}$$

The set of minimal cuts of F is denoted by $\mathrm{MCS}[F]$.

According to Definition 3, the following property holds.

Property 1. Let $\mathrm{MCS}[F]$ be the minimal cut set of F; then

$$\mathrm{MCS}[F] \subseteq \mathrm{CS}[F] \wedge \forall \rho \in \mathrm{CS}[F] - \mathrm{MCS}[F], \exists \delta \in \mathrm{MCS}[F], \delta \subset \rho. \tag{6}$$

Here, set $\mathrm{CS}[F]$-$\mathrm{MCS}[F]$ includes all the elements in $\mathrm{CS}[F]$, but not in $\mathrm{MCS}[F]$.

Let F be the BDD associated with F. Each path from the root of F to sink node 1 corresponds to an assignment σ and defines a cut of F, ρ, as follows: $x \in \rho$ iff the path goes through a node labeled by x and goes out of this node on the T-edge.

The set of cuts encoded by F is denoted by $\mathrm{CS}[F]$. Due to the fact that F compactly encodes the truth table of F in Shannon's form by means of subtree sharing, $\mathrm{CS}[F]$ is a cut set of F.

What is more, PMS BDD F is minimal if for any path from the root of F to sink node 1, the cut identified by it is minimal. The set of cuts encoded by minimal F, denoted by $\mathrm{MCS}[F]$, is still a minimal cut set of F.

Consider the PMS BDD with forward DFLM ordering shown in Figure 3. The cut set encoded by this BDD is

$$\{\{A_1\}, \{A_2\}, \{A_3, B_1\}, \{A_3, B_2, C_1\}, \{A_3, B_3, C_2\}, \{A_3, B_2, C_3\}, \{A_3, B_3, C_1\}, \{A_3, B_3, C_3\}, \{A_3, C_1\}, \{B_1\}, \{B_2, C_1\}, \{B_2, C_2\}, \{C_1\}\}. \tag{7}$$

According to forward ordering, the following property holds.

Property 2. Let F be a PMS BDD with forward ordering. Then,

$$\forall C \in \mathrm{CS}[F], \quad |C \cap X| = 1, \tag{8}$$

where $X = \{x_1, x_2, \ldots, x_n\}$ includes all variables relating to component X and n is the maximum of phase number.

Proof. Due to the fact that PMS considered in this paper is not maintained, a component remains down for the remaining phases if it is down in one phase; that is, x_j will never appear at the T-edge of node x_i.

In the following section, we try to show how to calculate the $\mathrm{MCS}[F]$ from PMS BDD F or $\mathrm{CS}[F]$. □

3.2. Inclusive Relation Elimination. Let $F = \mathrm{ite}\ (x, F_1, F_0)$ be a BDD. If ω is a cut of F_0, then ω does not contain x and ω is also a cut of F. If δ is a cut of F_1, δ augmented of x, $\rho = \delta U\{x\}$, is a cut of F. If ρ includes ω, we say that F has inclusive relation between ω and ρ. Notice that ω will never include ρ. More formally, we have the following definition.

Definition 4. Let F be a PMS BDD; F has inclusive relation across different cuts within $\mathrm{CS}[F]$ if $\exists \rho_1, \rho_2 \in \mathrm{CS}[F], \rho_1 \subset \rho_2$.

From the point of view of MCS, inclusive relation is one kind of redundancy relation. That is to say, in order to make $\mathrm{CS}[F]$ be MCS, ρ_{11} should be dropped.

It is obvious that the cut sets encoded by PMS BDD shown in Figures 4 and 5 have inclusive relations across different cuts. For example, $\{A_3, C_1\}$ is a cut, but its subset $\{C_1\}$ is still a cut.

The following theorem gives an inductive principle to eliminate the inclusive relations across cuts.

Theorem 1. *Let $F = ite\ (x, F_1, F_0)$ be a PMS BDD. Then,*

$$\mathrm{IncRed}(F) = \{\rho \mid (\rho = \delta \cup \{x\}) \wedge (\delta \in \mathrm{IncRed}(F_1) \backslash \mathrm{CS}[F_0])\} \cup \mathrm{IncRed}(F_0), \tag{9}$$

where IncRed (F) is the cut set of F which has no inclusive relation across different cuts.

Proof

$$\mathrm{CS}[F] = \{\rho \mid (\rho = \delta \cup \{x\}) \wedge (\delta \in \mathrm{CS}[F_1])\} \cup \mathrm{CS}[F_0]. \tag{10}$$

For $\mathrm{CS}[F_0]$, by eliminating the inclusive relations within it, we obtain IncRed (F_0).

For $\mathrm{CS}[F_1]$, by eliminating the inclusive relations within it, we obtain IncRed (F_1).

In order to eliminate the inclusive relations between $\{\rho \mid (\rho = \delta \cup \{x\}) \wedge (\delta \in \mathrm{IncRed}[F_1])\}$ and $\mathrm{CS}[F_0]$, $\mathrm{IncRed}(F_1) \backslash \mathrm{CS}[F_0]$ should be calculated. The without operation "\" is defined in [9] and can remove from IncRed (F_1) all those cuts that include a cut of $\mathrm{CS}[F_0]$. For example, if $\{A, B\}, \{B, C\} \in \mathrm{IncRed}(F_1)$ and $\{A\} \in \mathrm{CS}[F_0]$, then $\{A, B\} \notin \mathrm{IncRed}(F_1) \backslash \mathrm{CS}[F_0]$.

According to Theorem 1, the following property holds. □

Property 3. Let F be a PMS BDD. Then,

$$(\mathrm{IncRed}(F) \mathrm{CS}[F]) \wedge (\forall \rho \in \mathrm{IncRed}(F) - \mathrm{CS}[F], \exists \delta \in \mathrm{IncRed}(F), \delta \subset \rho). \tag{11}$$

Using Theorem 1, an operation (IncRed) can be derived to eliminate all the inclusive relations within $\mathrm{CS}[F]$. Reference [20] has implemented a similar operation called "Minsol" under the context of fault tree analysis of single-phase system. Thus, the operation implementation would not be given here.

As an illustration, IncRed operation is used to transform BDD in Figure 3 into the one as shown in Figure 4. The cut set encoded by the PMS BDD shown in Figure 4 is $\{\{A_1\}, \{A_2\}, \{A_3, B_2, C_3\}, \{A_3, B_3, C_3\}, \{B_1\}, \{B_2, C_2\}, \{C_1\}\}$.

3.3. External-Implication Relation Elimination. Let $F = \mathrm{ite}(x_i, F_1, F_0 = \mathrm{ite}(x_j, G_1, G_2)), i < j$, be a PMS BDD with forward ordering. If ρ_1 is a cut of G_1, ρ_1 contains x_j and $\rho_1 U\{x_j\}$ is also a cut of F. If ρ_2 is a cut of F_1, $\rho_2 U\{x_i\}$ is a cut of F. If ρ_1 includes ρ_2, we say that F has external-implication relation between cut $\rho_1 U\{x_j\}$ and cut $\rho_2 U\{x_i\}$. More formally, we have the following definition.

Definition 5. Let F be a PMS BDD with forward ordering. F has external-implication relation across different cuts within $\mathrm{CS}[F]$ *if* $\exists \rho_1, \rho_2 \in \mathrm{CS}[F]$:

$$(\rho_1 \cap \{x_i, x_j\} = \{x_i\}) \wedge (\rho_2 \cap \{x_i, x_j\} = \{x_j\}) \wedge (\rho_1^{<} = \rho_2^{<}) \wedge (\rho_1^{<} \supseteq \rho_2^{<}), \tag{12}$$

where $\rho^{<} = \{y \mid (y \in \rho) \wedge (\mathrm{order}(y) < \mathrm{order}(x_i))\}$. Notice that ρ_1, ρ_2 will never contain variables within $X - \{x_i, x_j\}$ according to Property 2.

From the point of view of MCS, external-implication relation is one kind of redundancy relation. Notice that x_i implies x_j, and $\{x_i, x_j\}$ and $\{x_i\}$ are equivalent objects. In order to make $\mathrm{CS}[F]$ be MCS, ρ_2 should be dropped. Consider the PMS BDD shown in Figure 5. The cut set encoded by this BDD is $\{\{A_1\}, \{A_2\}, \{A_3, B_2, C_3\}, \{A_3, B_3, C_3\}, \{B_1\}, \{B_2, C_2\}, \{C_1\}\}$. Due to the external-implication relation, $\{A_1\}$ should be eliminated from the cut set, and so does $\{A_2, B_2, C_3\}$.

The following theorem gives an inductive principle to eliminate the external-implication relation across different cuts.

Theorem 2. *Let $F = ite\ (x_i, F_1, F_0 = ite\ (y_j, G_1, G_0))$, $i < j$, be a PMS BDD with forward ordering. Then,*

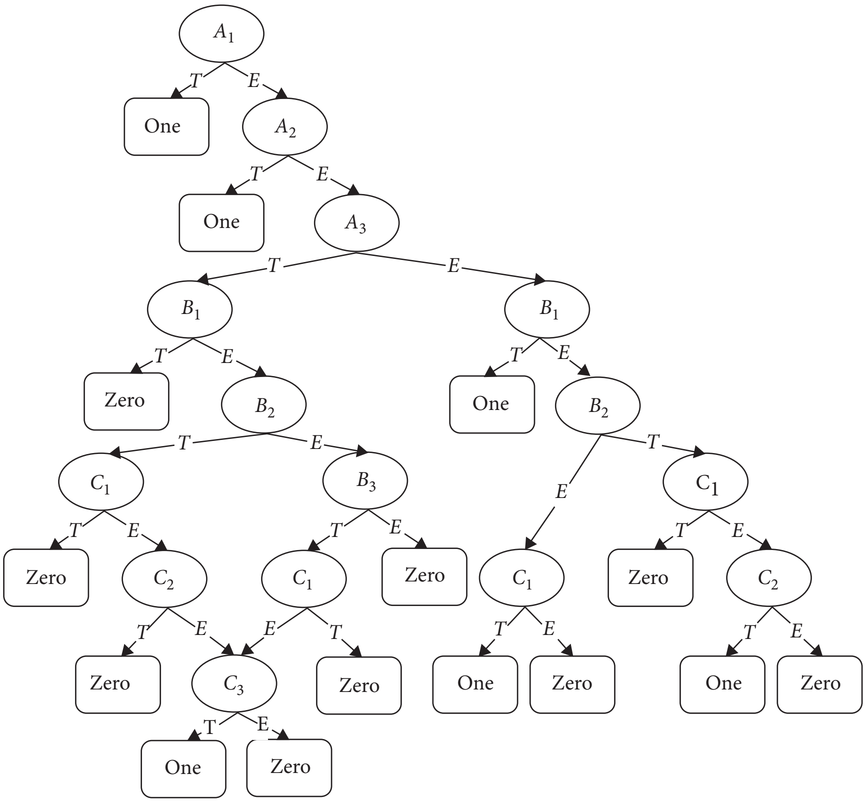


Figure 4: Results of IncRed operation for the PMS BDD in Figure 3.

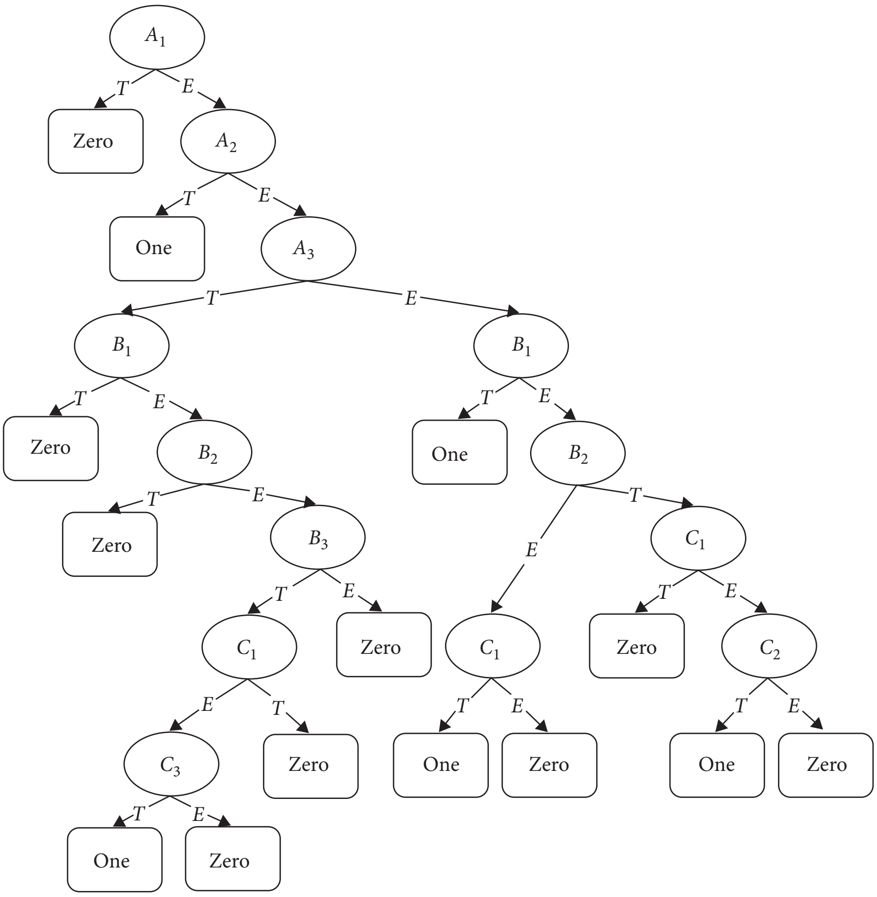


Figure 5: Results of ExImpRed operation for the PMS BDD in Figure 4.

$$\operatorname{ExImpRed}(F)=\begin{cases}\left\{\rho\,\middle|\,(\rho=\delta\cup\{x_i\})\wedge\delta\in\operatorname{ExImpRed}(F_1)\right\}\cup\operatorname{ExImpRed}(F_0), & x\neq y,\\ \left\{\rho\,\middle|\,(\rho=\delta\cup\{x_i\})\wedge\delta\in\operatorname{ExImpRed}(F_1)\backslash\operatorname{CS}[G_1]\right\}\cup\operatorname{ExImpRed}(F_0), & x=y,\end{cases}\tag{13}$$

where ExImpRed (F) is the cut set of F which has no external-implication relation across different cuts.

Proof. Two cases are possible:

(1) $x\neq y$. F_0 does not contain variables relating to component X.

If ρ is a cut of F, ρ is either a cut of F_0 (ρ does not contain x_i) or a cut of F_1 augmented of x_i (ρ only contains one variable x_i relating to component X). Thus, there is no external-implication relation relating to x_i within the cut set, and we obtain that

$$\left\{\rho\,\middle|\,(\rho=\delta\cup\{x_i\})\wedge\delta\in\operatorname{ExImpRed}(F_1)\right\}\cup\operatorname{ExImpRed}(F_0).\tag{14}$$

(2) $x=y$. There might be some cuts that have external-implication relation relating to x_i.

If ρ is a cut of F_0 and ρ contains x_j, ρ is also a cut of F. If δ is a cut of F_1, δ augmented of x_i is a cut of F. If $(\delta\supseteq\rho/\{x_j\})$, ρ and $\delta U\{x_i\}$ have external-implication relation. That is to say, δ should be eliminated. According to the fact that ρ contains x_j, $(\rho/\{x_j\})$ is a cut of G_1. Thus, we obtain that

$$\begin{gathered}\left\{\rho\,\middle|\,(\rho=\delta\cup\{x_i\})\wedge(\delta\in\in\operatorname{ExImpRed}(F_1)\backslash\operatorname{CS}[G_1])\right\}\\ \cup\operatorname{ExImpRed}(F_0).\end{gathered}\tag{15}$$

According to Theorem 2, the following property holds. □

Property 4. Let F be a PMS BDD with forward ordering. Then,

$$\begin{gathered}(\operatorname{ExImpRed}(F)\subseteq\operatorname{CS}[F])\wedge(\forall\rho\in\operatorname{ExImpRed}(F)-\operatorname{CS}[F],\\ \exists\delta\in\operatorname{ExImpRed}(F),\delta\subset\operatorname{Extend}(\rho)),\end{gathered}\tag{16}$$

where extend operation means adding variables with higher index to a cut. For example, extend $(\{A_2,B_2,C_3\})=\{A_3,A_2,B_3,B_2,C_3\}$.

As an illustration, the ExImpRed operation is used to transform BDD in Figure 4 into the BDD shown in Figure 5. The cut set encoded by this BDD is $\{\{A_2\},\{A_3,B_3,C_3\},\{B_1\},\{B_2,C_2\},\{C_1\}\}$. This cut set has no external-implication relation across different cuts.

3.4. MCS Calculation

Theorem 3. *Let F be a PMS BDD with forward ordering. Then,*

$$\operatorname{IncRed}(F)\cap\operatorname{ExImpRed}(F)=\operatorname{MCS}\,[F].\tag{17}$$

Proof. We have known that $\operatorname{CS}[F]$ might have inclusive relations and external-implication relations across the included cuts. According to Property 2, we know that x_i and x_j will not be simultaneously included by a cut. Thus, there is no internal-implication relation.

According to Theorem 2, we know that ExImpRed operation can only eliminate the external-implication relations relating to one component. However, $\operatorname{CS}[F]$ might have external-implication relations relating to more than one component. For example, consider a $\operatorname{CS}[F]$ has two cuts (one is $\{A_2,B_2\}$ and the other is $\{A_1,B_1\}$. Notice that x_i implies x_j and $\{A_1,B_1,A_2,B_2\}$ and $\{A_1,B_1\}$ are equivalent objects. Thus, $\{A_1,B_1\}$ should be dropped. However, our ExImpRed operation cannot use $\{A_2,B_2\}$ to eliminate $\{A_1,B_1\}$. Now, we show that the ExImpRed operation can do this elimination indirectly. If both $\{A_2,B_2\}$ and $\{A_1,B_1\}$ belong to $\operatorname{CS}[F]$, then F must have a structure as shown in Figure 6. There are two points: (1) according to the BDD generation process, B_1 will appear at the T-edge of node A_2 under the condition that the T-edge of node A_1 has a B_1 and the T-subBDD of node A_2 has a B_2; (2) the T-edge of this B_1 is 1 due to the fact that $\{A_2,B_2\}$ is a cut. Thus, $\{A_2,B_1\}$ must be included by $\operatorname{CS}[F]$. Now, the ExImpRed operation can use $\{A_2,B_1\}$ to eliminate $\{A_1,B_1\}$ and then use $\{A_2,B_2\}$ to eliminate $\{A_2,B_1\}$. Thus, our ExImpRed operation can eliminate all kinds of external-implication relations within $\operatorname{CS}[F]$.

ExImpRed operation eliminates all the external-implication relations from the cut set $\operatorname{CS}[F]$. However, it will disturb some inclusive relations. For example, consider the PMS BDD F shown in Figure 7, where cut set $\operatorname{CS}[F]=\{\{A_1\},\{A_2,B_1,C_1\},\{A_2,B_2,C_1\},\{B_1,C_1\}\}$. After the ExImpRed operation, we get $\operatorname{ExImpRed}(F)=\{\{A_1\},\{A_2,B_2,C_1\},\{B_1,C_1\}\}$. Here, the inclusive relation between cut $\{B_1,C_1\}$ and cut $\{A_2,B_1,C_1\}$ is destroyed. Thus, after another IncRed operation, we get IncRed $(\operatorname{ExImpRed}(F))=\{\{A_1\},\{A_2,B_2,C_1\},\{B_1,C_1\}\}$ instead of $\{\{A_1\},\{B_1,C_1\}\}$, the $\operatorname{MCS}[F]$.

On the other hand, IncRed operation eliminates all the inclusive relations from the cut set $\operatorname{CS}[F]$. However, it will disturb some external-implication relations. For the PMS BDD F shown in Figure 7, after the IncRed operation, we get $\operatorname{IncRed}(F)=\{\{A_1\},\{A_2,B_2,C_1\},\{B_1,C_1\}\}$. Here, the external-implication relation between cut $\{A_2,B_1,C_1\}$ and cut $\{A_2,B_2,C_1\}$ is destroyed. Thus, after another ExImpRed operation, we get ExImpRed $(\operatorname{IncRed}(F))=\{\{A_1\},\{A_2,B_2,C_1\},\{B_1,C_1\}\}$ instead of $\{\{A_1\},\{B_1,C_1\}\}$, the $\operatorname{MCS}[F]$. □

4. Application

With edge, compute and storage systems reside at the edge as well, as close as possible to the component, device,

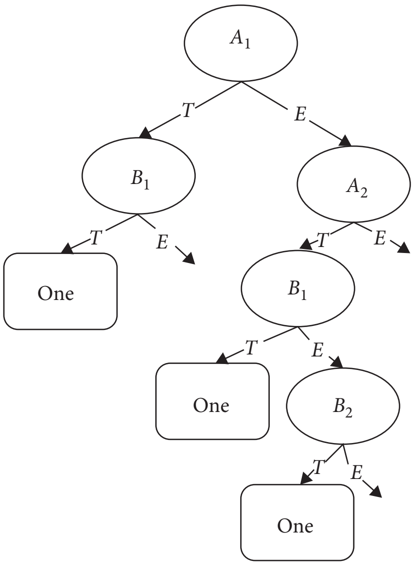


Figure 6: The PMS BDD encoding $\{A_2, B_2\}$ and $\{A_1, B_1\}$.

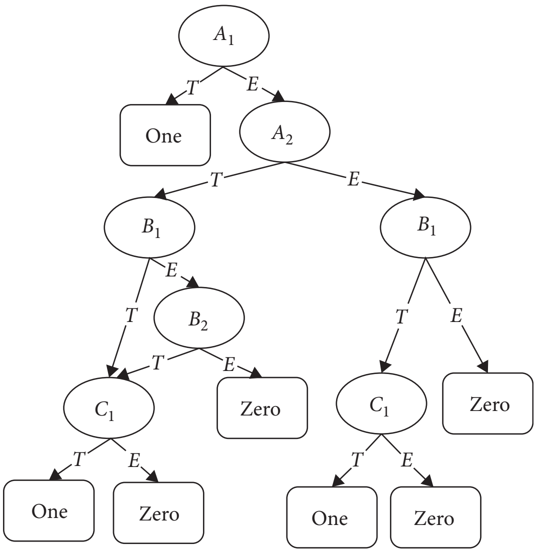


Figure 7: An example PMS BDD with forward ordering.

application, or human that produces the data being processed. The purpose is to remove processing latency, because the data need not be sent from the edge of the network to a central processing system and then back to the edge.

The applications for edge make sense: Internet of things-connected devices are a clear use for edge computing architecture. With remote sensors installed on a machine, component, or device, they generate massive amounts of data. If that data is sent back across a long network link to be analyzed, logged, and tracked, that takes much more time than if the data is processed at the edge, close to the source of the data.

There are patterns of failure. (1) If you place too much at the edge, it is easy to overwhelm the smaller processor and storage platforms that exist there. In some cases, storage could be limited to a few gigabytes and processing using a single CPU. Power and size restrictions are really what set the limits. (2) Another pattern is failure to integrate security from concept to production. Security is systemic to edge computing architectures and centralized processing. Security needs to span both and use mechanisms such as identity and access management. Encryption is not a nice-to-have, but rather a requirement for device safety.

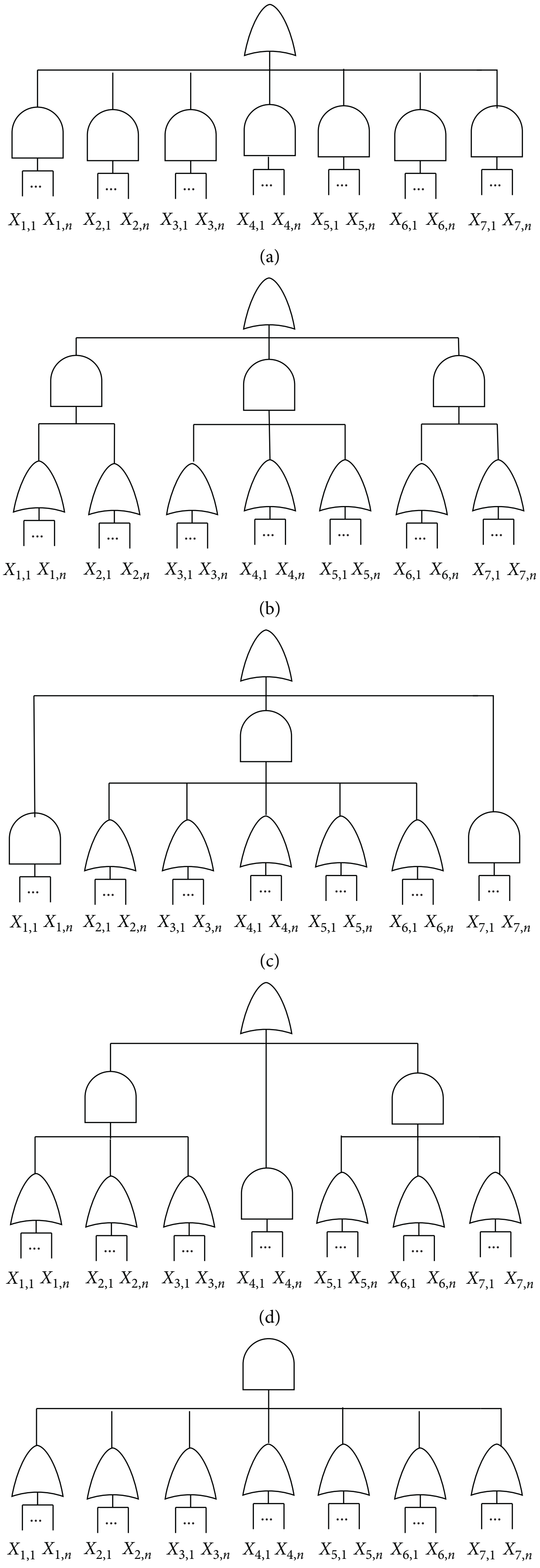
Consider a benchmark MEC PMS borrowed from [21]. This MEC PMS has the following:

(1) It has 7 groups of components.

(2) Each group has 5 components.

(3) Five MEC node configurations are shown in Figure 8.

(4) Four MEC service configurations are listed in Table 1.


$X_{1,1}$ $X_{1,n}$ $X_{2,1}$ $X_{2,n}$ $X_{3,1}$ $X_{3,n}$ $X_{4,1}$ $X_{4,n}$ $X_{5,1}$ $X_{5,n}$ $X_{6,1}$ $X_{6,n}$ $X_{7,1}$ $X_{7,n}$
...

(a)

(b)

(c)

(d)

(e)

Figure 8: Five MEC node configurations.

Table 1: Four MEC service configurations.

Name	Phase number	Configuration
M1	2	A, E
M2	3	A, B, E
M3	4	A, B, C, E
M4	5	A, B, C, D, E

Table 2: Results of MCS calculation for PMS BDD with forward ordering.

Mission	|CS [Mi]|	|ExImpRed (Mi)|	|IncRed (Mi)|	|IncRed (Mi)| ∩ |ExImpRed (Mi)|
M1	3399661	104492	279943	78132
M2	109574034	5802547	1944295	78307
M3	1616724670	60025676	8222890	81432
M4	1224665026	121374302	17953442	81682

Using Theorem 4, MCS is calculated with forward ordering. The results are shown in Table 2. Here, we only present the numbers of cuts in the set CS [Mi], IncRed (Mi), ExImpRed (Mi), and IncRed (Mi) ∩ ExImpRed (Mi). From the results in Table 2, the following conclusions can be derived: (1) Different phase combination will result in different sizes of cut sets. Some more complex combination may have smaller cut sets as illustrated by M3 and M4. (2) For the MCS calculation IncRed ∩ ExImpRed of this PMS, forward ordering can generate extremely smaller cut sets for the intermediate BDD operations compared with the original CS.

5. Conclusion

Based on the mobile edge computing techniques, commercial service providers, such as video content providers, can benefit from low-latency edge resources to provide their users with more efficient service acquisition, thereby improving the quality of experience [22–24]. Compared with single-phase systems, fault tree analysis of MEC PMS is much more complex, because of the dependencies across the phases.

The analysis to be performed on a fault tree can be either qualitative or quantitative. For the qualitative fault tree analysis of PMS by means of BDD, no much related work can be found. In this paper, we have presented some efficient methods to calculate the MCS encoding by a PMS BDD. The basic idea is to eliminate different kinds of redundancy relations from a cut set encoding by a PMS BDD. These methods are based on several BDD operations, such as IncRed and ExImpRed. The IncRed can eliminate all inclusive relations. The ExImpRed can eliminate all external-implication relations. Using some proper combinations of these operations, MCS can be calculated correctly.

Smart city is a fast-developing system enabled by the Internet of things (IoT) with massive collaborative services (e.g., intelligent transportation and collaborative diagnosis) [25, 26]. Therefore, one direction of our future work is to consider the qualitative analysis of commercial services in MEC smart city. As an extension of this research work, we will also improve the presented BDD-based analysis methodology for more generalized MEC service systems [27].

Data Availability

The MCS calculation data used to support the findings of this study may be released upon application to the Fujian Province University Key Laboratory of Computational Science, who can be contacted at CSlab@hqu.edu.cn.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the NSF of China under Grant no. 61972156 and Program for Innovative Research Team in Science and Technology in Fujian Province University, Quanzhou High-Level Talents Support Plan under Grant no. 2017ZT012.

References

[1] X. Zhou, W. Liang, K. I.-K. Wang, H. Wang, L. T. Yang, and Q. Jin, “Deep-Learning-enhanced human activity recognition for Internet of healthcare things,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6429–6438, 2020.

[2] X. Zhou, Y. Li, and W. Liang, “CNN-RNN based intelligent recommendation for online medical pre-diagnosis support,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2020.

[3] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Bhuiyan, “Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[4] X. Xu, S. Fu, W. Li, F. Dai, H. Gao, and V. R. Chang, “Multi-objective data placement for workflow management in cloud infrastructure using NSGA-II,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 5, 2020.

[5] H. Yu and M. Tang, “Reliability assessment for systems suffering competing degradation and random shocks under fuzzy environment,” *Science Progress*, vol. 103, no. 1, Article ID 003685041988108, 2020.

[6] K. Yan, J. Huang, W. Shen, and Z. Ji, "Unsupervised learning for fault detection and diagnosis of air handling units," *Energy and Buildings*, vol. 210, Article ID 109689, 2020.

[7] K. Yan, W. Shen, Q. Jin, and H. Lu, "Emerging privacy issues and solutions in cyber-enabled sharing services: from multiple perspectives," *IEEE Access*, vol. 7, pp. 26031–26059, 2019.

[8] Q. Zhai, L. Xing, R. Peng, and J. Yang, "Aggregated combinatorial reliability model for non-repairable parallel phased-mission systems," *Reliability Engineering & System Safety*, vol. 176, pp. 242–250, 2018.

[9] R. Peng, Q. Zhai, L. Xing, and J. Yang, "Reliability analysis and optimal structure of series-parallel phased-mission systems subject to fault-level coverage," *IIE Transactions*, vol. 48, no. 8, pp. 736–746, 2016.

[10] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Transactions on Network Science and Engineering*, 2020.

[11] Y. Liu, Y. Chen, and T. Jiang, "Dynamic selective maintenance optimization for multi-state systems over a finite horizon: a deep reinforcement learning approach," *European Journal of Operational Research*, vol. 283, no. 1, pp. 166–181, 2020.

[12] A. Shrestha, L. Xing, and Y. Dai, "Reliability analysis of multistate phased-mission systems with unordered and ordered states," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 4, pp. 625–636, 2011.

[13] Y. Mo, L. Xing, and S. V. Amari, "A multiple-valued decision diagram based method for efficient reliability analysis of non-repairable phased-mission systems," *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 320–330, 2014.

[14] Y. Mo, "A multiple-valued decision-diagram-based approach to solve dynamic fault trees," *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 81–93, 2014.

[15] A. Shrestha, L. Xing, and Y. Dai, "Decision diagram-based methods, and complexity analysis for multistate systems," *IEEE Transactions on Reliability*, vol. 59, no. 1, pp. 145–161, 2010.

[16] Y. Mo, L. Xing, and J. B. Dugan, "Performability analysis of k-to-l-out-of-n computing systems using binary decision diagrams," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 1, pp. 126–137. in press, 2015.

[17] X. Zhou, W. Liang, S. Huang, and M. Fu, "Social recommendation with large-scale group decision-making for cyber-enabled online service," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1073–1082, 2019.

[18] X. Zhou, W. Liang, K. Wang, R. Huang, and Q. Jin, "Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data," *IEEE Transactions on Emerging Topics in Computing*, 2018.

[19] R. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Transactions on Computers*, vol. C-35, no. 8, pp. 677–691, 1986.

[20] A. Rauzy, "New algorithms for fault trees analysis," *Reliability Engineering & System Safety*, vol. 40, no. 3, pp. 203–211, 1993.

[21] Z. Xu, Y. Mo, Y. Liu, and T. Jiang, "Reliability assessment of multi-state phased-mission systems by fusing observation data from multiple phases of operation," *Mechanical Systems and Signal Processing*, vol. 118, pp. 603–622, 2019.

[22] C. Zhou, A. Li, A. Hou, Z. Zhang, Z. Zhang, and F. Wang, "Modeling methodology for early warning of chronic heart failure based on real medical big data," *Expert Systems with Applications*, 2020.

[23] K. Yan, L. Liu, Y. Xiang, and Q. Jin, "Guest editorial: AI and machine learning solution cyber intelligence technologies: new methodologies and applications," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6626–6631, 2020.

[24] X. Zhou, W. Liang, K. I. Wang, and L. T. Yang, "Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations," *IEEE Transactions on Computational Social Systems*, 2020.

[25] X. Xu, Q. Huang, X. Yin, M. Abbasi, M. Khosravi, and L. G. Qi, "Intelligent offloading for collaborative smart city services in edge computing," *IEEE Internet of Things Journal*, 2020.

[26] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, "Joint optimization of offloading utility and privacy for edge computing enabled IoT," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2622–2629, 2020.

[27] L. Wang, X. Zhang, R. Wang, C. Yan, H. Kou, and L. Qi, "Diversified service recommendation with high accuracy and efficiency," *Knowledge-Based Systems*, vol. 204, Article ID 106196, 2020.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8839047, 11 pages
https://doi.org/10.1155/2020/8839047

Research Article

Rational Protocols and Attacks in Blockchain System

Tao Li,[1,2,3] Yuling Chen ,[1,2] Yanli Wang,[3] Yilei Wang,[3] Minghao Zhao,[4] Haojia Zhu,[3] Youliang Tian,[1,2] Xiaomei Yu,[5] and Yixian Yang[1,2]

[1]*State Key Laboratory of Public Big Data, Guizhou University, Guizhou 550025, China*
[2]*College of Computer Science and Technology, Guizhou University, Guizhou 550025, China*
[3]*School of Information Science and Engineering, Qufu Normal University, Rizhao 276825, China*
[4]*School of Software, Tsinghua University, Peking 100084, China*
[5]*School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China*

Correspondence should be addressed to Yuling Chen; ylchen3@gzu.edu.cn

Received 13 July 2020; Revised 25 August 2020; Accepted 15 September 2020; Published 24 September 2020

Academic Editor: Xiaolong Xu

Blockchain has been an emerging technology, which comprises lots of fields such as distributed systems and Internet of Things (IoT). As is well known, blockchain is the underlying technology of bitcoin, whose initial motivation is derived from economic incentives. Therefore, lots of components of blockchain (e.g., consensus mechanism) can be constructed toward the view of game theory. In this paper, we highlight the combination of game theory and blockchain, including rational smart contracts, game theoretic attacks, and rational mining strategies. When put differently, the rational parties, who manage to maximize their utilities, involved in blockchain chose their strategies according to the economic incentives. Consequently, we focus on the influence of rational parties with respect to building blocks. More specifically, we investigate the research progress from the aspects of smart contract, rational attacks, and consensus mechanism, respectively. Finally, we present some future directions based on the brief survey with respect to game theory and blockchain.

1. Introduction

To facilitate data processing, clients prefer to host them to a trusted party. However, the security problems have proliferated due to lack of trust parties. In 2017, 180,000 patient records were stolen by Hackers from clinics such as Aesthetic Dentist in the United States [1]. In September 2017, 140 million clients' personal information in Equifax, an American credit reporting company, was stolen [2]. In April 2018, 87 million customers' data information was leaked on Facebook platform [3]. All these information leakage incidents derive from the heavy dependence upon the centralized trusted parties. Once trusted parties are compromised, system securities are in grave danger. Therefore, it is challenging to discharge the dependence on trusted parties and enhance the nodes' independence of distributed computing [4–7].

Blockchain technology, a "decentralized" underlying support technology, can establish trust among distributed nodes. Thus, it may solve the problems mentioned above. In January 2016, the British Government released a thematic study on blockchain to actively promote the application of blockchain in financial and government affairs. In that same year, the People's Bank of China held a seminar on digital currency to explore the feasibility of using blockchain technology to issue virtual currency in order to improve the efficiency, convenience, and transparency of financial activities. The white paper, issued in 2016 by China Blockchain Technology and Application Development, defines blockchain as a new application framework for computer technology such as distributed data storage, point-to-point transmission, consensus mechanism, encryption algorithm, etc. However, there has also been a tremendous amount of work in the underlying technology of blockchain. The basic framework of blockchain, combined with game theory [8, 9], complex networks, and rational protocols [10, 11], is presented in Figure 1. Note that Figure 1 is different from that of the figure in [12], although it inherits from it, since the combined parts are the focus of this survey.

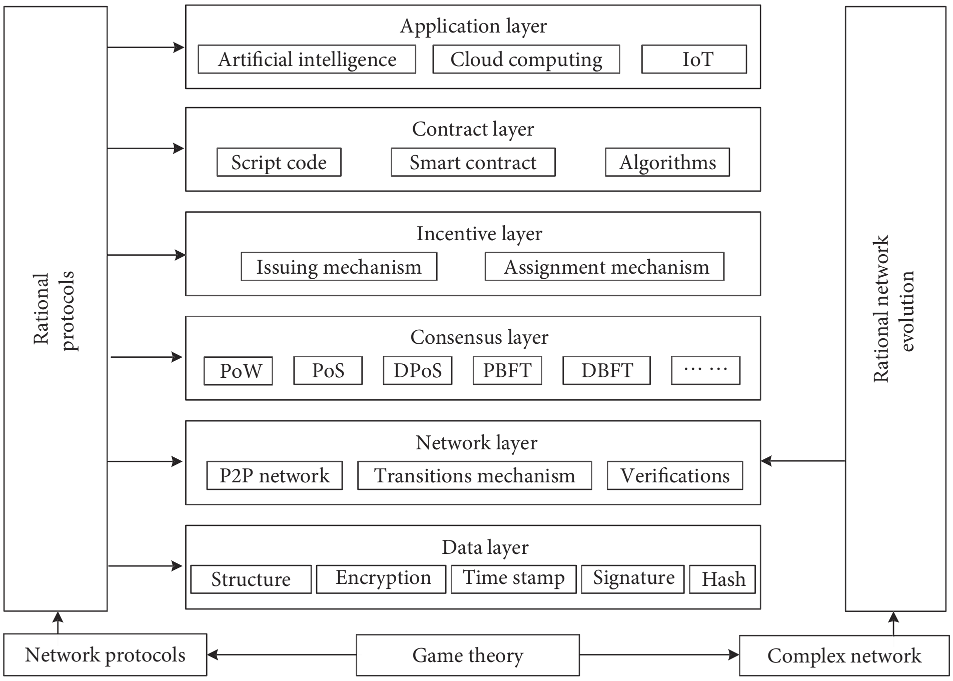


Figure 1: The framework of blockchain combined with game theory.

The underlying technology of blockchain covers a wide range of contents, which can be divided into data layer, network layer, consensus layer, incentive layer, contract layer, and application layer according to their functions. The nodes in the network layer, without mutual trust, achieve verification through a certain consensus mechanism such as proof of work (PoW), proof of stake (PoS), and the practical Byzantine fault tolerance algorithm (PBFT). These mechanisms provide solid foundations for the smooth implementation of the contract layer and the application layer. In effect, the blockchain network is a complex network, which exhibits the character of small world. Therefore, we may leverage the theory of complex network to investigate the topological structure of blockchain network. On the other hand, the nodes in blockchain network are social, whose behaviours may be influenced by the topological structure. The incentive layer connects the bottom data with the upper application through the network layer and the consensus layer. The incentive mechanism (IC) is the core driving source of the technical value of the blockchain [13].

Generally speaking, the main function for the nodes in network layer mechanism is to maximize their economic benefits. Therefore, the incentive layer needs to design the economic incentive mechanism reasonably such that the nodes maximize their economic benefits. Meanwhile, the security and effectiveness of blockchain system should be ensured in order to form a stable consensus on blockchain history. In economics, this kind of behaviours is called rational behaviours. In fact, rationality is a basic concept of economic theory, which mainly focus on the interaction among incentive structures and analyses the optimal strategy for rational participants. In consequence, rational protocols, participated by rational nodes, can describe the characteristics of nodes and provide an efficient solution for some practical problems.

In Figure 1, rational behaviours intersect each layer in blockchain, which can be utilized to achieve some desired conclusions. In this paper, we address the rational behaviours in blockchain, which are common concepts in game theory. That is, we may probe rational behaviours toward the view of game theory since it is a powerful tool for rational behaviours. More specifically, we first investigate the rational behaviours in smart contracts. For example, attackers may adopt rational strategies in rational smart contracts and criminal smart contracts in order to maximize their incomes (refer to Section 3). Since the inspiration of the rational parties is economic incentives, they have enough motivation to sponsor rational attacks by leveraging incentive mechanisms in game theory (refer to Section 4). The main function of the consensus layer is responsible for assigning rewards. It is crux for the whole blockchain system to maintain a sound economic and ecological environment. Therefore, we outline some rational consensus mechanisms (refer to Section 5). Finally, we highlight the future directions with respect to the combination of game theory and blockchain (refer to Section 6).

2. Classical Games Implemented in Blockchain

Parties, in blockchain, manage to maximize their economic incomes, so they have incentives to adopt rational strategies when they participate in protocols. It is not in conflict when the strategies are incompatible with the security requirements of the protocols. Put differently, rational parties in blockchain can achieve maximum incomes by just honestly following the protocols. However, honest behaviours, in most cases, are not harmonious with these rational strategies. That is, rational parties can always get around the secure protocols to maximize their incomes, which breach the secure protocols to some extent. From the above, the interaction between rational parties and secure protocols can be constructed as games. On the other hand, strategies which are incentive compatibility can be solved by mechanism design. Both fall into the field of game theory. Therefore, it is necessary to survey on game theory in blockchain.

In effect, various works (aka rational protocols) analyse the security of protocols with respect to blockchain toward the view of game theory. The basic idea is to establish protocols as games, where an equilibrium exists such that rational parties achieve maximum incomes and meanwhile protocols are secure according to the definitions. Basically, these rational protocols derive from and are constructed as classical games, like prisoner's dilemma game and tragedy of the commons game. In this paper, we just investigate these two games [14].

2.1. Prisoner's Dilemma Game.

The derivation of prisoner's dilemma game is as follows: two prisoners (say Alice and Bob) are thrown in jail without confession in collusion. They have two choices: defect and collude. There are altogether four outcomes according to different choices. (1) If both defect, each gets eight years in prison. (2) If both collude, each gets one year in prison. (3) If Alice defects while Bob colludes, Alice is acquitted of a charge while Bob gets ten years in prison. (4) If Bob defects while Alice colludes, Bob is acquitted of a charge while Alice gets ten years in prison.

This is a classical game in game theory, and there is one Nash equilibrium (defect, defect). Put differently, two prisoners have incentives to defect since it is optimal for them. However, it is obvious that mutual collusion may maximize their utilities. Thus, prisoners run into dilemma, where they have to settle for the second-best solution. Normally, we outline it in matrix form as shown in Table 1.

Prisoner's dilemma game applies to such scenarios where parties cannot collude beforehand, and they do not trust each other. Blockchain is rightly such a distributed scenario, where distrustful parties are distributed all over the world. All parties try to maximize their utilities while they may involve deeply the dilemma, where the conflict exists between the security requirements of the protocols and the utility requirements of the rational parties. In Sections 3–5, strategic attacks and rational protocols based on game theory highlight how these blockchain techniques utilize them to solve some vexing problems in their respective fields.

2.2. Tragedy of the Commons Game.

The security issues in blockchain come from the conflict between the individual and the collective rationality. The blockchain system is secure if all parties follow the protocols therein, which are incentive compatible with the utility requirements. Otherwise, the security of the protocols is breached since rational parties always manage to maximize their utilities by violating the protocols.

However, if rational parties achieve their utility endlessly at the expense of the security of the whole blockchain system. The blockchain will disappear someday, which lead to undesirable results that rational parties gain nothing. This scenario is similar to the tragedy of the common game, where the security of the blockchain system is the common goods for rational parties.

The basic model of tragedy of the commons game in blockchain is as follows. Say there are n parties in blockchain system and each of them has m strategies $q_1, q_2, \ldots, q_m$ to choose. The revenue $V(i) = Q(q_1, q_2, \ldots, q_m)$ of each strategy is a decreasing function $Q(\cdot)$ of the whole strategies. On the other hand, there exist cost $c_1, c_2, \ldots, c_m$ with respect to each strategy. Therefore, the utility for party i when he chooses strategy q_j is

$$U(i) = q_i \cdot V(i) - q_i \cdot c_i. \quad (1)$$

In fact, the tragedy of the commons game is a social trap for rational parties. They must leverage their strategies such that they may maximize their utilities in the long run. As mentioned above, the interaction between rational parties and protocols in blockchain can be modelled as a tragedy of the commons game, especially in pool mining (refer to Section 4).

3. State of the Art of Smart Contracts

In 1996, Nick Szabo put forward the concept of smart contract. "A smart contract is a set of promises defined in digital form, including agreements on which contract participants can implement these promises." Before the implementation of smart contracts, participants make a commitment. Then the smart contract will be executed automatically when the conditions are triggered, where no participants can bias the contract commitments. Smart contract can be automatically executed without prior review, which may avoid intractable issues such as contract disputes [12]. However, smart contract has not been applied to the actual industry due to the lack of effective technical support and trust platform. Ethereum provides an implementable development platform for smart contract by drawing on the characteristics of blockchain technology such as decentralization, nontamperability, process transparency, traceability, and so on [15]. Ethereum offers complete scripting language of Turing, which can embed more additional information. Therefore, any smart contract, once precisely defined, can be constructed and implemented automatically on Ethereum. The white paper of smart contracts: 12 Use Case for Business & Beyond, published by the Chamber of Digital Commerce and Smart Contracts Alliance in December 2016, points out that smart contract can be applied in various fields like mortgage loans, Internet of things, medical research, etc.

The security problem of smart contract can be summarized into two aspects: internal security and external attack. Luu et al. [16] put forward a new security problem with respect to smart contract and present the corresponding solutions to strengthen the robustness of smart

Table 1: The matrix of prisoner's dilemma game.

Alice, Bob	Defect	Collude
Defect	(−1, −1)	(0, −10)
Collude	(−10, 0)	(−8, −8)

contracts. Kosba et al. [17] propose a decentralization system, Hawk, which addresses the privacy of the content of smart contracts. Bhargavan et al. [18] compile the smart contracts into F^* language to verify the correctness of the smart contract. Atzei et al. [19] summarize the smart contracts on the Ethereum. They focus on the robustness of contracts and categorize programming pitfalls in the programs. Dika categorizes the vulnerabilities of smart contracts [20]. Furthermore, they analyse code security issues in smart contracts on the Ethereum, such as Oyente, Security, and SmartCheck. Recently, Nikolic et al. [21] analyse about a million smart contracts, among which 34,200 are inherently vulnerable to hackers. In addition, they also utilize MAIAN as a tool to analyse the validity of smart contracts. Based on a sample of 3,759 smart contracts, 3,686 of them contain loopholes, with an 89% probability of vulnerability. A loophole in the smart contract will also cause the customer's electronic property to be locked up in Ethereum. In November 2017, about $300 million were permanently frozen in the Ethereum for the maloperation of some users of Ethereum smart contract.

In addition to the inherent security problems in smart contract, there are also many attacks specifically targeting smart contract. Velner et al. [22] introduce an attack model based on smart contract, through which attackers can destroy the normal work of the mine pool. Juels et al. [23] have proposed the concept of criminal smart contract. Criminals can use smart contracts to carry out some illegal activities, such as illegally selling pirated films. They focus on discussing the feasibility and harmfulness of criminal smart contracts. Finally, they appeal to the introduction of relevant laws and policies to improve technical prevention measures. Brunoni and Beaudet-Labrecque also have in-depth discussions on how to use smart contracts to commit cybercrimes [24]. Alharby and van Moorse even think there are no countermeasures to criminal smart contracts at present [25]. The current research has increased people's concern about the harm caused by smart contracts. Furthermore, scholars are trying to resist the negative effects caused by smart contracts. Wang et al. analyse the validity of the criminal smart contracts by setting some parameters reasonably. They prove that, given proper parameters, the successful probability of criminal intelligence contract is extremely low [26]. Zhang et al. further impair the threshold of criminal smart contracts by utilizing Q-learning [27]. Bigi et al. combine game theory and formal method to verify the validity of smart contracts [28]. They mainly analyse the uncertainty brought to the system by the deposit introduced by the smart contract. Consequently, they address the validity of smart contracts. Although smart contracts are not perfect and secure, they still have wide application. For example, when clients outsource some tasks in cloud computing, there are two problems: (1) the client's cost is higher and (2) once most of the clouds collude, the principal still cannot learn the correct value. Dong et al. [29] propose a smart contract to verify anticollusion in cloud computing by combining smart contract with game theory. They assume that both the client and cloud are rational participants, where the former strives to maximize their benefits by providing the correctness of the calculation results.

Figure 2 presents the future directions for smart contracts based on the above related works. The environment for smart contracts must be more complex and practical.

4. Rational Mine Pool Attack and Incentive Mechanism

Toward the economic point of view, the incentive mechanism in bitcoin has solved the problem of miners' motivation. However, the application of game theory in the field of economics has become extraordinarily mature. Therefore, it is natural to analyse some problems in bitcoin and the blockchain from the perspective of game theory. As we all know, the most important mechanism in bitcoin is mining. Tschorsch and Scheuermann provide the basic concepts and workflows of bitcoin [30]. If miners want to get bitcoin, they need to solve a specific mathematical problem. That is, miners, who solve the problem and find a bitcoin block, can get 12.5 bitcoins. By February 5, 2020, the value of a bitcoin is 9217.61 USD, which provides enough motivation for miners. However, it requires a certain amount of computing power to solve these problems. It usually takes several months or even years for a single miner to find a bitcoin block. Normally, a new block will appear on the bitcoin network in about 10 minutes. Therefore, most miners work in vain. For this reason, some miners constitute a mining pool by taking their computing power as a whole. If they find an effective block within a proper time, they will share the rewards according to their computing power. However, toward the view of game theory, miners may achieve additional rewards by leveraging the security vulnerabilities in the incentive mechanism. Therefore, rational miners have incentives to deviate from the honest strategy, which is similar to the noncooperative game in game theory.

Schrijvers et al. [31] define a pool payment function in a single pool from the perspective of game theory. In the mining pools, miners can strategically choose the time to report once they find a new block. They present three properties of the payment function: (a) incentive compatibility, (b) proportional payments, and (c) budget balanced (BB). Schrijvers et al. analyse whether the payment functions based on allocation strategies satisfy these characteristics. Proportional reward function, denoted as *R prop*, refers to the allocation of rewards according to the computing power of each miner, which is an earlier allocation strategy for the mine pool. However, *R prop* only meets the properties of (b) and (c). The Pay-Per-Share (PPS) reward function only meets the property of (a). Eyal and Sirer highlight the incentive compatibility of bitcoin protocols [32], where miners are allowed to collude. They prove that rational miners will

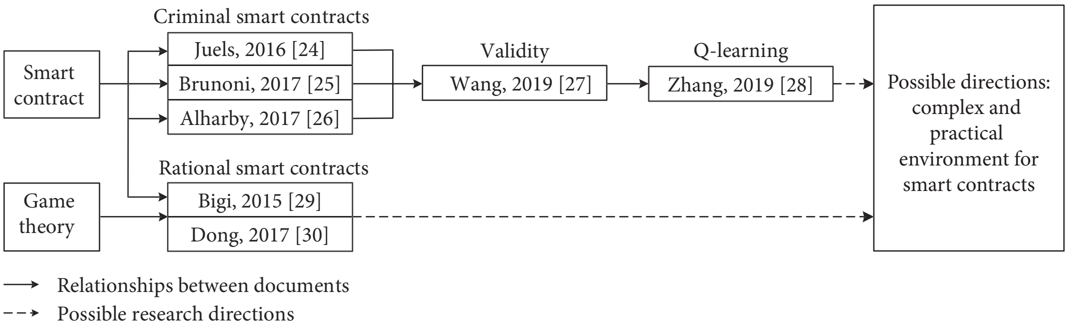


Figure 2: The possible directions for smart contracts.

eventually turn into selfish miners who collude to form a selfish pool. On the other hand, the pool can attract an increasing number of selfish miners. Therefore, the pool may develop into a super pool consisting most miners in the whole system. As a consequence, bitcoin becomes a centralized system again, which goes against the original intention of bitcoin. Put differently, any selfish mining strategy can form a super pool that controls the majority miners. Such selfish mining strategy is known as selfish mining attack. In order to resist such attacks, Eyal and Sirer propose an improved version of the bitcoin protocol which is backwards-compatible.

Nayak et al. [33] expand the space for mining strategies, including "stubborn" strategies. They prove that selfish mining is not a good strategy for a larger strategic space. Nayak et al. mainly survey two types of mining attacks: "selfish mining" style and Eclipse attack. Miners combine eclipse attacks based on the network layer to increase their rewards. In other words, the victims of some eclipse attacks, given the optimal strategy, can benefit from the attack. Kroll et al. [34] regard bitcoin mining as a game between miners and bitcoin holders. Furthermore, they address the impact of 51% attacks on the game and manage to reach an equilibrium. They also propose a new attack: goldfinger attack, where the attackers' incentives derive from exterior motivations other than bitcoin. For example, attackers may be a law enforcement agency or intelligence unit hoping to impair the bitcoin's ability. Finally, they feature the influence of goldfinger attack on the game and point out the necessity of government supervision.

Heilman introduces the freshness preferred (FP) mechanism [35], which punishes selfish miners who do not release blocks in time by using unforgeable timestamps. They increase the threshold, compared with [32], from 0.25 to 0.32. However, FP mechanism is neither incentive compatible nor robust to forgeable timestamps. That is, the implementation of FP mechanism depends on unforgeable timestamps, which are difficult to implement [36, 37]. Therefore, the implementation of FP mechanism has certain limitations. Solat and Potop-Butucaru propose a solution for selfish mining attacks and withholding attack, Zero Block [38], which does not depend on timestamps. In Zero Block scheme, the honest miners will refuse to accept the new block if a selfish miner holds a block for more than a mat interval.

Sapirshtein et al. [39] extend the work of literature and propose an efficient algorithm, which can calculate ε-optimal ($\varepsilon \geq 0$) selfish mining strategy. They prove the correctness of the algorithm and analyse its error bounds. Miners can increase their rewards relying on the efficient algorithm mentioned above. Furthermore, they launch selfish mining attacks. They also prove that if we consider the delay of the block in the network transmission, the threshold value becomes zero again. That is, no matter how many resources the attacker controls, there will always be a selfish mining strategy, which will bring more profits than honest mining. Finally, they summarize the interaction between selfish mining and double spending.

Eyal discusses the withholding attack between two mining pools [40]. There is a contradiction between individual rationality and collective rationality with respect to two mining pools, which is similar to the tragedy of the commons game. Eyal proposes a withholding attack between two mining pools. More specifically, the manager of one attacking pool first registered as a normal miner in the other victim pool. He accepts several tasks from the victim pool and assigns them to infiltrating miners in the attacking pool. The ratio of infiltrating miners in the attacking pool is called infiltration rate. The attacking pool submits part of the work capacity of the infiltrating miners to the victim pool, allowing the victim pool to evaluate the capacity of infiltrating miners. When the infiltrating miners submit full-work certificate, the attacking pool will ignore this work. The drawback of the withholding attack is that the overall calculation capacity of the victim pool has not increased (infiltrating miners do not work), but its average budget has decreased. In effect, it weakens itself for the attacking pool to split the computing power to the victim pool. As a result, withholding attack generally reduces the computing power of the entire network. For two mining pools, withholding attack is the only Nash equilibrium [41]. However, it is better for both mining pools not to conduct withholding attack. Toward the perspective of game theory, it is a miner's dilemma for the pool whether or not to conduct withholding attack. The miners' constant mining process is similar to a repeated prisoner's dilemma game. Rosenfeld suggests modifying the block structure to solve this problem [42].

In the process of mining, there are still various detailed problems to be explored besides constructing selfish mining pools. Miners earn transaction fees once they find a new block, which contain several transactions in the block. However, the number of the transactions is a difficult issue, which should balance the propagation speed and transaction fees. That is, a block with fewer transactions has higher propagation speed while less transaction fees. A block with more transactions has more transactions fees while lower propagation speed. To solve this issue, Houy defines a bitcoin mining game among miners [43], assuming that the number of transactions contained in a block is the outcome of a game. Houy discusses the impact of transaction fees on the block size toward an economic point of view [44]. Any case with fixed transaction fees is equivalent to setting the maximum block size allowed for a block. Moreover, imposing a fixed transaction fee on transactions is equivalent to imposing a compulsive tax on each transaction, which will undoubtedly impair bitcoin's economic and ecological environment. However, the transaction fees will reduce to zero again if the maximum size of each block is not restrained. In this case, the miner is similar to the follower in Stackelberg game. In order to maximize his rewards within the limited block size, the miners manage to include as many transactions as possible in the block. Figure 3 presents the future directions for rational mining attack based on the related works.

5. Consensus Mechanism

The main role of the blockchain consensus layer is to reach consensus in a decentralized system with highly decentralized decision-making power. The consensus issue is a research hotspot with great academic value for a long time. The core indicators are fault-tolerant proportions and convergence speed. The commonly used consensus mechanism is proof of work (PoW) [45] (refer to Algorithm 1). Each node in the PoW calculates SHA256 of an ever-changing block header. The result of the consensus mechanism is to find a hash value that is less than a certain value. A new block is generated if the new hash value is found. In bitcoin, the node that calculates the hash value is called miner and the process of consensus is called mining. Although PoW can implement decentralization and distributed accounting, PoW heavily relies on nodes' power, which are low throughput and of poor scalability. In fact, the PoW mechanism in bitcoin introduces the game theory without forcing the number of faulty (malicious) nodes. PoW assumes that each node in the network is rational and utilizes economic incentives to maintain the operation within PoW. Recently, researchers propose other consensus mechanisms without relying on computing power, such as the proof-of-stake (PoS) mechanism and the delegated-proof-of-stake (DPoS) mechanism [46].

The basic idea of PoS mechanism is to prove ownership of their stakes. PoS is more reliable than PoW since the more stakes the nodes hold, the less likely they will attack the system. Peercoins, for the first time, achieve a true equity certificate, which is based on the age of the coins, and decide who creates the next block. In DPoS mechanism, the equity owner selects the representative to generate the next block. The above consensus mechanisms are based on the game of economic benefits. The nodes will lose some economic benefits once a malicious node destroys the mechanism. Therefore, most nodes have incentives to maintain the mechanism. Meanwhile, the achievement of consensus must be guaranteed probabilistically after the generation of multiple blocks. In distributed systems, the classical Byzantine algorithm can solve the deterministic problem. Castro and Liskov propose a practical Byzantine fault tolerance (PBFT) [47], which solve the problem of low efficiency of the classic Byzantine fault-tolerant algorithm. More specifically, say there are N nodes in the blockchain network, and the number of Byzantine malicious nodes is $f = ((N - 1)/3)$. Then PBFT can ensure that at least $2f + 1$ nodes reach consensus before adding information to the distributed shared ledger. The advantages of PBFT include the following: (1) it is a complete theoretical proof system, (2) each block is generated by a unique master node, and (3) there are no forks.

However, the complexity of the network is $O(N^2)$, which greatly increases network overhead and reduces system efficiency. Therefore, the performance of the system with PBFT is not high, which is more suitable for a blockchain system with a smaller number of nodes, e.g., consortium blockchain. Currently, Hyperledger Fabric 1.0 is developing a consensus module such as BFT-Smart, simplified Byzantine fault tolerance (SBFT), and HoneyBadgerBFT (refer to Algorithm 2) based on plug-in [48, 49]. In addition, it is also a problem needed to be solved regarding how to support the nodes to dynamically join and quit in the consensus mechanism. Miller et al. study the possibility of the PoW mechanism to solve the problem of single-point Byzantine consistency in the presence of a few Byzantine nodes in an asynchronous network [50]. Garay et al., based on the PoW mechanisms, propose two consensus mechanisms for multiple-instance setting [51]. Their mechanism meets all the properties (a), (b), and (c), but it does not consider the asynchronous network and the problem of participation of honest nodes.

Garay et al. denominate the consensus mechanism as Bitcoin's Backbone protocol (refer to Algorithm 3). They first portray three parameters γ, β, and f. γ and β represent the hashing power of honest participants and adversaries in each round. f represents the PoW value that all participants in the bitcoin network expect to achieve in each round. Some basic properties of the two backbone protocols are proposed. (1) The common prefix property: the blockchain maintained by honest participants will have the largest common prefix when $\gamma > \lambda\beta$, where $\lambda \in [1, \infty)$ and $\lambda^2 - f\lambda + 1 \geq 0$. It requires that most participants be honest in order to satisfy the common prefix characteristics when $f \longrightarrow 0$ and λ tends to be 1. (2) The chain-quality property: the proportion of blocks maintained by any honest participant that are contributed by an honest participant is at least $1 - (1/\lambda)$ when $\gamma > \lambda\beta (\lambda \in [1, \infty))$. For example, when $\lambda \longrightarrow 1$, the block contributed by honest participants is only a minority of the blockchain. Garay et al. prove that if the backbone protocol satisfies the common prefix property and the block quality property, it can meet the two basic properties: agreement and validity of the Byzantine protocol with great probabilities. However, the work [49] does

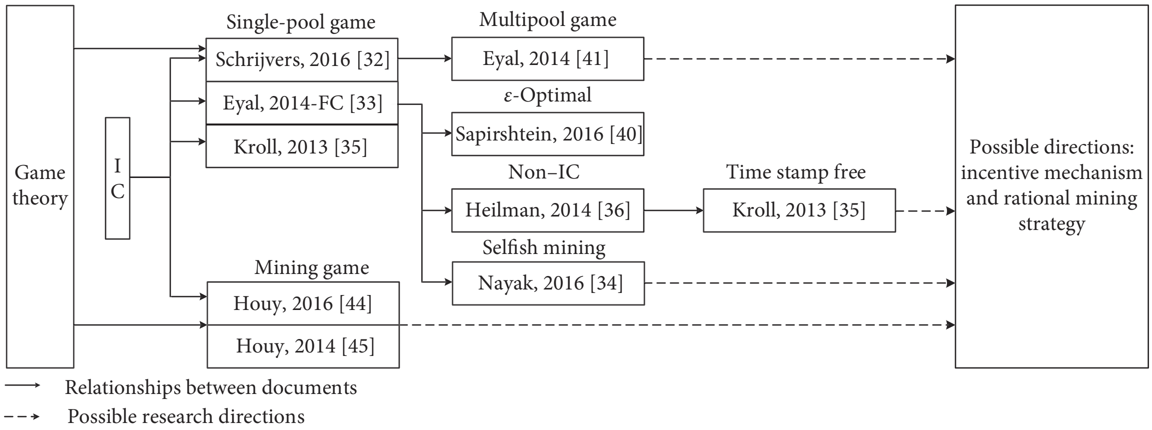


Figure 3: The possible directions for rational mining attacks.

Step 1. The initial voting set V is empty, and each party i has a proposal p_i by solving a hard problem
Step 2. Each party broadcasts p_i and others update their voting set after they verify the validity of the proposal
Step 3. They vote for the proposal p_{most} with the most votes
Step 4. p_{most} is recorded to the blockchain and the one who proposes it wins the rewards

Algorithm 1: The PoW consensus protocol.

Step 1. One proposal $p_i$consisting of $\lfloor B/N \rfloor$ transactions is randomly selected and encrypted to be c_i, where B is the batch size parameter and N is the number of parties
Step 2. Parties agree on these ciphertexts
Step 3. Parties first decrypt c_i if it has passed the verification

Algorithm 2: The HoneyBadgerBFT protocol.

Step 1. Each party i maintains a local chain C and updates it by invoking PoW algorithm (refer to Algorithm 1).
Step 2. However, party i does not update it immediately when C has any change. Instead, i first checks if there are any other "better" chain by verifying its communication tape.
Step 3. An input x is determined by some functions therein and local chain C is updated to be C' by invoking PoW.
Step 4. The updated chain C' is broadcasted to other parties. Note that Backbone consensus protocol is depended on input contribution function $I(\cdot)$ and the chain reading function $R(\cdot)$. Readers may find more details in [51].

Algorithm 3: The Backbone protocol.

not consider the delay in the message delivery process. Before the adversary sends its own information, it can see the information of all honest participants, which will bring privacy issues. Sompolinsky and Zohar propose a longest-chain rule called GHOST [52] in order to reach the consensus of the blockchain. When there is a fork in the blockchain, GHOST selects the subtree with the largest weight at the fork. A variant of GHOST has been adopted by the Ethereum project. However, GHOST does nothing for attacks such as selfish mining. Moreover, [52] only considers limited delays and specific attacks instead of withholding block attacks. Lewenberg et al. improve GHOST [53] to reduce the priority of heavyweight miners and further increase system throughput.

Decker et al. propose PeerCensus [54] to allow nodes dynamically joining and quitting. They implement PeerCensus protocol to Discoin, which achieves strong consistency. Note that Discoin relies not on the blockchain but on the Byzantine protocol. Pass et al. consider the blockchain consensus mechanism in asynchronous networks in a formal model, where new nodes can join the network any time and the adversary can adaptively corrupt the honest nodes [55]. Pass et al. extend the nature of the blockchain consensus protocol: consistency, future self-consistency, g-chain growth, and μ-chain quality. In [52], Sompolinsky and Zohar point out the nature of chain growth, while they only consider the expected growth of the chain. The chain-growth characteristics are also mentioned in one of their documents

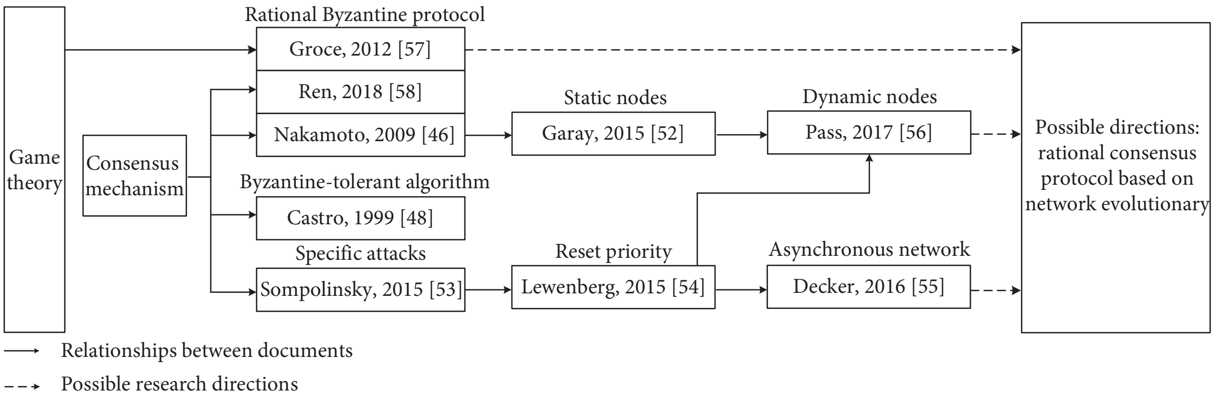


Figure 4: The possible directions for consensus mechanism.

[51]. The nature of chain quality is first proposed and discussed in the Bitcoin forum [32]. To the best of our knowledge, [49] first defines the notion of "chain quality." The study in [55] proves that, given certain conditions, the blockchain consensus protocol satisfies all properties.

Rational behaviours are also implemented in Byzantine agreement protocols. For example, Groce et al. propose a Byzantine agreement based on rational adversaries [56]. They combine game theory and cryptography, assuming that some of the participants in the network are honest and some are rational. Rational participants are controlled by rational adversaries, who try to maximize their rewards by biasing the final outcome. Groce and Katz focus on rational broadcast and Byzantine agreement. They prove that many self-evident conclusions in the classic Byzantine agreement do not establish in a rational adversary environment. For example, a consensus cannot be reached when $t \geq n/2$ and the consensus problem does not stipulate the broadcast problem when $t < n/2$. Finally, they classify the outcomes with respect to rational Byzantine agreement into three categories: (1) agreement on 0, (2) agreement on 1, and (3) inconsistency. Participants have different preferences for outcomes, which depends on the utility function. They highlight the influence of adversary preferences on rational Byzantine agreements: (1) the adversary's preferences are fully known, (2) only the adversary's preference between consistency and inconsistency is known, and (3) it is not known that in consistency, the opponent prefers 0 or 1.

In game theory, these scenarios are called complete information games or incomplete information games. Groce et al. prove that if the traditional Byzantine consistency problem is safe, then the rational adversary strategy in rational Byzantine consistency is the Nash strategy. Ren et al. consider the message type when discussing the Byzantine consensus [57]. They also introduce the concept of rational behaviour. That is, rational issuers of transactions either mine on their own or hire others to mine. Ren et al. define the characteristics of value propagation in the value-transfer ledgers model (VTL model): (1) the rational sender proves the authenticity of the transaction to the recipient, (2) rational recipients should verify the authenticity of the transaction after receiving it, and (3) the authenticity of these transactions is not important if certain transactions have no effect on the transactions that the rational recipient has received. They also prove that if all nodes are rational, then effective transactions in the system can resist double-flower attacks. Figure 4 presents the future directions for consensus mechanism.

6. Future Research Directions

Game theory permeates all levels of blockchain technology, and the behaviours of the entire blockchain rely heavily on the motivation to maintain the blockchain, which is usually an economic incentive. Therefore, it is a hot topic to consider the influence of rational behaviours on key technologies in blockchain technology. At the same time, the blockchain network constructed by the participants has complex network characteristics, and the behaviour of the participants affects the topology of the network, which is closely related to the consensus mechanism. In summary, the future development direction of blockchain technology is mainly concentrated in the following aspects:

(1) The smart contracts automatically run according to the code agreed in advance, which are not affected by the outside world during the execution process. Therefore, it is widely used in various fields in real life. In addition to its own vulnerabilities and external attacks, it becomes a research hotspot to survey on the impacts of the rational behaviours on the effectiveness. Current researches on rational smart contracts focus on the scenario of complete information games. That is, the utility functions are common knowledge. However, there are lots of scenarios with asymmetry information in real life. These asymmetry scenarios result in uncommon knowledge for the utility functions. Therefore, the implementation of incomplete information games in blockchain is one of the future research directions.

(2) The incentive mechanism is one of the core mechanisms of the blockchain technology, which is crucial

to blockchain. At present, game theory is used to discuss the incentive compatibility strategy in single- or multiple-mine pools. So, the miners have enough mining incentives to promote the efficiency of the blockchain. In order to maximize their benefits, rational miners must maintain competition and cooperation with other miners and selectively release the information they have. Therefore, it is one of the future research directions to weigh the information of all parties and reasonably develop the optimal incentive compatibility strategies.

(3) At present, most of the consensus mechanisms based on Byzantine fault-tolerant algorithm assume that there are Byzantine nodes and honest nodes. Most works discuss the proportion of Byzantine nodes and the fault tolerance of consensus mechanism. Some works consider the influence of joining and quitting of nodes under the asynchronous network on the consensus mechanism. Under the assumption that the nodes are rational, the choices of neighbour nodes, when new nodes join the asynchronous network, will also consider the impact on future revenue. Therefore, the phenomenon of small worlds in the process of network evolution is affected by the rational behaviours. Meanwhile, rational Byzantine protocols have certain impacts on information broadcasting and consistency compared to traditional Byzantine protocols. It is one of the future research directions to integrate rational behaviours into network evolution and Byzantine protocols.

Data Availability

The authors claim that all figures are drawn by the tool of Visio. No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the Foundation of National Natural Science Foundation of China Grant nos. 62072273, 61962009, 61832012, 61771231, 6150028, 61672321, and 61771289, Major Scientific and Technological Special Project of Guizhou Province (20183001), Foundation of Guizhou Provincial Key Laboratory of Public Big Data (no. 2019BDKFJJ009), Talent Project of Guizhou Big Data Academy, and Guizhou Provincial Key Laboratory of Public Big Data ([2018]01).

References

[1] http://www.360doc.com/content/17/0519/20/30350201655373279.shtml.

[2] https://www.wosign.com/Info/equifax.htm.

[3] http://www.dsj365.cn/front/article/6019.html.

[4] X. Zheng and H. Liu, “A scalable coevolutionary multi-objective particle swarm optimizer,” *International Journal of Computational Intelligence Systems*, vol. 3, no. 5, pp. 590–600, 2010.

[5] X. Yu, H. Wang, Zheng, X. Zheng, and Y. Wang, “Effective algorithms for vertical mining probabilistic frequent patterns in uncertain mobile environments,” *International Journal of Ad-Hoc Ubiquitious Computing*, vol. 23, no. 3-4, pp. 137–151, 2016.

[6] X. Yu, W. Feng, H. Wang, Q. Chu, and Q. Chen, “An attention mechanism and multi-granularity-based Bi-LSTM model for Chinese Q&A system,” *Soft Computing*, vol. 24, no. 8, pp. 5831–5845, 2019.

[7] Y. Tian, Z. Wang, J. Xiong, and J. Ma, “A blockchain-based secure key management scheme with trustworthiness in DWSNs,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.

[8] X. F. Ding and H. C. Liu, “A new approach for emergency decision-making based on zero-sum game with Pythagorean fuzzy uncertain linguistic variables,” *International Journal of Intelligent Systems*, vol. 34, no. 7, pp. 1667–1684, 2019.

[9] R. Ureña, G. Kou, J. Wu, F. Chiclana, and E. Herrera-Viedma, “Dealing with incomplete information in linguistic group decision making by means of interval type-2 fuzzy sets,” *International Journal of Intelligent Systems*, vol. 34, no. 6, pp. 1261–1280, 2019.

[10] Y. Wang, M. Zhao, Y. Hu, Y. Gao, and X. Cui, “Secure computation protocols under asymmetric scenarios in enterprise information system,” *Enterprise Information Systems*, pp. 1–21, 2019.

[11] Y. Wang, C. Zhao, Q. Xu, Z. Zheng, Z. Chen, and Z. Liu, “Fair secure computation with reputation assumptions in the mobile social networks,” *Mobile Information Systems*, vol. 2015, Article ID 637458, 8 pages, 2015.

[12] Y. Ren, Y. Liu, S. Ji, A. K. Sangaiah, and J. Wang, “Incentive mechanism of data storage based on blockchain for wireless sensor networks,” *Mobile Information Systems*, vol. 2018, Article ID 6874158, 10 pages, 2018.

[13] Z. Chen, Y. Tian, and C. Peng, “An incentive compatible rational secret sharing scheme using blockchain and smart contract,” *SCIENCE CHINA Information Sciences*, Springer, Berlin, Germany, 2020.

[14] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press Books, Cambridge, MA, USA, 1994.

[15] B. V. Ethereum, “A next-generation smart contract and decentralized application platform,” 2014, https://github.com/ethereum/wiki/wiki/English-WhitePaper.

[16] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, “Making smart contracts smarter,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 254–269, Vienna, Austria, October 2016.

[17] A. E. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, “Hawk: the blockchain model of cryptography and privacy-preserving smart contracts,” in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 839–858, San Jose, CA, USA, May 2016.

[18] K. Bhargavan, A. Delignat-Lavaud, C. Fournet et al., “Formal verification of smart contracts: short paper,” in *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, pp. 91–96, Vienna, Austria, October 2016.

[19] N. Atzei, M. Bartoletti, and T. Cimoli, “A survey of attacks on ethereum smart contracts (SoK),” in *Proceedings of the 6th*

International Conference on Principles of Security and Trust, POST 201, pp. 164–186, Uppsala, Sweden, April 2017.

[20] A. Dika, "Ethereum smart contracts: security vulnerabilities and security tools," Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2017.

[21] I. Nikolic, A. Kolluri, I. Sergey, P. Saxena, and A. Hobor, "Finding the greedy, prodigal, and suicidal contracts at scale," in *Proceedings of the 34th Annual Computer Security Applications Conference*, pp. 653–663, San Juan, PR, USA, December 2018.

[22] Y. Velner, J. Teutsch, and L. Luu, "Smart contracts make bitcoin mining pools vulnerable," in *Proceedings of the FC 2017 International Workshops*, pp. 298–316, Sliema, Malta, April 2017.

[23] A. Juels, A. E. Kosba, and E. Shi, "The ring of gyges: investigating the future of criminal smart contracts," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 283–295, Vienna, Austria, October 2016.

[24] L. Brunoni and O. Beaudet-Labrecque, "Smart contracts and cybercrime: a game changer?" *Mathematical Structures and Modeling*, vol. 4, no. 44, 2017.

[25] M. Alharby and A. van Moorsel, "Blockchain-based smart contracts: a systematic mapping study," 2017, https://arxiv.org/abs/1710.06372.

[26] Y. Wang, A. Bracciali, T. Li, F. Li, X. Cui, and M. Zhao, "Randomness invalidates criminal smart contracts," *Information Sciences*, vol. 477, pp. 291–301, 2019.

[27] L. Zhang, Y. Wang, F. Li, Y. Hu, and M. H. Au, "A game-theoretic method based on Q-learning to invalidate criminal smart contracts," *Information Sciences*, vol. 498, pp. 144–153, 2019.

[28] G. Bigi, A. Bracciali, G. Meacci, and E. Tuosto, "Validation of decentralised smart contracts through game theory and formal methods," *Programming Languages with Applications to Biology and Security*, Springer, Berlin, Germany, pp. 142–161, 2015.

[29] C. Dong, Y. Wang, A. Aldweesh, P. McCorry, and A. van Moorsel, "Betrayal, distrust, and rationality: smart-counter-collusion contracts for verifiable cloud computing," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 211–227, Dallas, TX, USA, October 2017.

[30] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: a technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[31] O. Schrijvers, J. Bonneau, D. Boneh, and T. Roughgarden, "Incentive compatibility of bitcoin mining pool reward functions," in *Proceedings of the FC 2016*, pp. 477–498, Christ church, Barbados, February 2016.

[32] I. Eyal and E. G. Sirer, "Majority is not enough: bitcoin mining is vulnerable," in *Proceedings of the FC 2014*, pp. 436–454, Christ church, Barbados, March 2014.

[33] K. Nayak, S. Kumar, A. Miller, and E. Shi, "Stubborn mining: generalizing selfish mining and combining with an eclipse attack," in *Proceedings of the IEEE European Symposium on Security and Privacy*, pp. 305–320, Saarbrücken, Germany, March 2016.

[34] J. A. Kroll, I. C. Davey, and E. W. Felten, "The economics of bitcoin mining, or bitcoin in the presence of adversaries," *Proceedings of the WEIS*, vol. 2013, pp. 11–32, 2013.

[35] E. Heilman, "One weird trick to stop selfish miners: fresh bitcoins, a solution for the honest miner (poster abstract)," in *Proceedings of the WAHC 2014*, pp. 161-162, Christ Church, Barbados, March 2014.

[36] B. Cohen, "An attack on the timestamp semantics of bitcoin," 2014.

[37] A. Boverman, "Timejacking & bitcoin," 2011.

[38] S. Solat and M. Potop-Butucaru, "Zeroblock: preventing selfish mining in bitcoin," https://arxiv.org/abs/1605.02435v1, 2016.

[39] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Proceedings of the FC 2016*, pp. 515–532, Christ church, Barbados, February 2016.

[40] I. Eyal, "The miner's dilemma," in *Proceedinsg of the 2015 IEEE Symposium on Security and Privacy*, pp. 89–103, San Jose, CA, USA, May 2015.

[41] W. Jiang, C. Huang, and X. Deng, "A new probability transformation method based on a correlation coefficient of belief functions," *International Journal of Intelligent Systems*, vol. 34, no. 6, pp. 1337–1347, 2019.

[42] M. Rosenfeld, "Analysis of bitcoin pooled mining reward systems," 2011, https://arxiv.org/abs/1112.4980.

[43] N. Houy, "The bitcoin mining game," *Ledger*, vol. 1, pp. 53–68, 2016.

[44] H. Nicolas, "The economics of bitcoin transaction fees," *SSRN Electronic Journal*, pp. 1407–1420, 2014.

[45] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2019.

[46] D. Larimer, C. Hoskinson, and S. Larimer, "Bitshares: a peer to-peer polymorphic digital asset exchange," 2017.

[47] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 173–186, New Orleans, LO, USA, February 1999.

[48] A. N. Bessani, J. Sousa, and E. A. P. Alchieri, "State machine replication for the masses with BFT-SMART," in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 355–362, Edinburgh, UK, June 2014.

[49] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of BFT protocols," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 31–42, Vienna, Austria, October 2016.

[50] A. Miller and J. J. LaViola Jr., *Anonymous Byzantine Consensus from Moderately-Hard Puzzles: A Model for Bitcoin*, University of Central Florida Tech, Oviedo, FL, USA, 2014.

[51] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: analysis and applications," in *Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 281–310, Sofia, Bulgaria, April 2015.

[52] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin," in *Proceedings of the FC 2015*, pp. 507–527, San Juan, PR, USA, January 2015.

[53] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, "Inclusive block chain protocols," in *Proceedings of the FC 2015*, pp. 528–547, San Juan, PR, USA, January 2015.

[54] C. Decker, J. Seidel, and R. Wattenhofer, "Bitcoin meets strong consistency," in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, Singapore, January 2016.

[55] R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *Proceedings of the 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 643–673, Paris, France, April 2017.

[56] A. Groce, J. Katz, A. Thiruvengadam, and V. Zikas, “Byzantine agreement with a rational adversary,” in *Proceedings of the ICALP 2012*, pp. 561–572, Warwick, UK, July 2012.

[57] Z. Ren, K. Cong, T. Aerts, B. de Jonge, A. Morais, and Z. Erkin, “A scale-out blockchain for value transfer with spontaneous sharding,” in *Proceedings of the CVCBT 2018*, pp. 1–10, Zug, Switzerland., June 2018.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8891664, 15 pages
https://doi.org/10.1155/2020/8891664

Research Article

A Novel IM Sync Message-Based Cross-Device Tracking

Naixuan Guo, Junzhou Luo, Zhen Ling, Ming Yang, Wenjia Wu, and Xiaodan Gu

School of Computer Science and Engineering, Southeast University, Nanjing, China

Correspondence should be addressed to Naixuan Guo; guonaixuan@seu.edu.cn

Received 23 June 2020; Revised 17 August 2020; Accepted 31 August 2020; Published 22 September 2020

Academic Editor: Xiaolong Xu

Cybercrime is significantly growing as the development of internet technology. To mitigate this issue, the law enforcement adopts network surveillance technology to track a suspect and derive the online profile. However, the traditional network surveillance using the single-device tracking method can only acquire part of a suspect's online activities. With the emergence of different types of devices (e.g., personal computers, mobile phones, and smart wearable devices) in the mobile edge computing (MEC) environment, one suspect can employ multiple devices to launch a cybercrime. In this paper, we investigate a novel cross-device tracking approach which is able to correlate one suspect's different devices so as to help the law enforcement monitor a suspect's online activities more comprehensively. Our approach is based on the network traffic analysis of instant messaging (IM) applications, which are typical commercial service providers (CSPs) in the MEC environment. We notice a new habit of using IM applications, that is, one individual logs in the same account on multiple devices. This habit brings about devices' receiving sync messages, which can be utilized to correlate devices. We choose five popular apps (i.e., WhatsApp, Facebook Messenger, WeChat, QQ, and Skype) to prove our approach's effectiveness. The experimental results show that our approach can identify IM messages with high F_1-scores (e.g., QQ's PC message is 0.966, and QQ's phone message is 0.924) and achieve an average correlating accuracy of 89.58% of five apps in an 8-people experiment, with the fastest correlation speed achieved in 100 s.

1. Introduction

According to a report of CyberEdge Group, 80.7% of surveyed organizations were affected by a successful cyberattack in 2019 (https://cyber-edge.com/wp-content/uploads/2020/03/CyberEdge-2020-CDR-Report-v1.0.pdf). To defend against the cyberattacks, the law enforcement usually adopts network surveillance technology to track a suspect and analyzes the network traffic of his device to derive his online profile. However, the traditional network surveillance using the single-device tracking method can only obtain the online activities of a suspect's single device. As different devices play different roles in the MEC environment, one suspect can employ multiple devices to carry out a cybercrime. For example, a suspect may launch a cyberattack on a personal computer and communicate with his accomplices on a mobile phone. If only tracking his personal computer, the law enforcement cannot capture the suspect's accomplices. In this paper, we propose a novel cross-device tracking approach which is able to correlate one suspect's different devices (e.g., personal computer and mobile phone), which can help the law enforcement monitor a suspect's online activities more comprehensively.

Cross-device tracking approaches are mainly divided into two categories, i.e., deterministic tracking and probabilistic tracking [1]. The former approach relies on deterministic identifiers to correlate one user's different devices. For example, since one user logs in the same YouTube account on two devices, the YouTube website can achieve his login information directly to realize cross-device tracking. However, this approach can only be utilized by those companies which need users to log in. At present, most researchers focus on probabilistic tracking. This kind of approach is based on the similarity of user preference and behavior when one user operates different devices. For instance, Kane et al. [2] found that there is a certain overlap between the sites that users visit on their personal computer and mobile phone. Therefore, it is possible to perform cross-device tracking based on users' web logs on distinct devices. By providing users' web logs of different devices, ICDM2015

[3] and CIKM2016 [4] held two cross-device tracking competitions. These datasets provided were collected by commercial companies within about one month, and these two competitions mainly focused on designing linking algorithms based on the datasets. This approach has two drawbacks. First, this kind of cross-device approach often requires users' active participation, i.e., software installing and log records, which is difficult in practical application. Second, it is not suitable for fine-grained correlation with a specific range of users who have similar behaviors, while our approach in this paper is better to deal with this situation.

IM applications are typical CSPs in MEC, and they are widely used in daily life, e.g., chatting online with friends, transmitting files, and conducting video conferences. Google initially proposed an IM app called "Hangouts" that enables one user account to be logged in on multiple devices at the same time and sync messages automatically across devices. Therefore, if one user starts a chat on his computer, he can continue his chat on his phone (https://support.google.com/hangouts/answer/2944865?hl=en&ref_topic=6386410). In this way, instant messages are exchanged from one-to-one to one-to-multiple. For example, when you receive a message from a friend, your IM apps simultaneously used on two devices can receive this message almost at the same time. Then, many other IM apps such as WhatsApp, Facebook Messenger, WeChat, QQ, and Skype have added this functionality that is referred to as the cross-device message sync mechanism in this paper. The cross-device sync message can be leveraged to correlate two devices belonging to the same user so as to achieve the goal of cross-device tracking.

In this paper, we put forward a new cross-device tracking approach based on the IM sync message detection. We assume the law enforcement can capture target devices' network traffic from the gateway. Then, we classify the network traffic by IPs and identify the device type according to the user-agent field of HTTP packets. After that, we filter out the retransmitted packets which may reduce our correlation accuracy. We analyze a set of ground-truth IM application network traffic and find that sync messages are contained in the gate server's flow. Thus, we employ domain names matching to identify those flows of gate servers, which have specific domain names and summarize several filtering rules to identify the other flows. In order to identify sync messages in the gate server's flow, we extract the features of sync messages in advance and employ machine learning and rule matching as classification methods. According to the identification result, we extract sync messages' timestamps to form each device's message-receiving time list. Then, we calculate the time interval between each timestamp in two devices' time lists. It is regarded that when the interval is less than a threshold, it is one successful sync message match. According to the matching result, we employ the SPRT (sequential probability ratio testing [5]) algorithm to determine whether two devices are correlated or not. SPRT algorithm fits our scenario very well and gives theoretical support to our approach. It can make a decision as fast as possible when we observe the sequence of sync messages matching results.

To evaluate our cross-device tracking approach, we choose five popular IM applications (https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/) in our experiments: WhatsApp, Facebook Messenger, WeChat, QQ, and Skype. In our experiments, we first test the ability of our approach to identify messages. The best performance is PC's message of QQ, whose F_1-score reaches 0.966, while the worst is PC's message of Skype which also has 0.820. Then, we carry out device correlation experiments by designing a scenario that includes 8 people with 16 devices. The results show that our approach achieves a 97.9% matching accuracy of WeChat and an 83.3% accuracy of Skype. When the number of participants increases to 16, the matching accuracy of WeChat also exceeds 86%. Besides, we also analyze the factors, i.e., different users, message frequency, and user amount. We find that different users have little effect on correlating results, and the increase of message frequency or user amount reduces the correlation accuracy. Last but not the least, we count our matching time of each app, finding that most of the matching time length of QQ and WeChat is less than 400 seconds. This result indicates that our approach is faster than those approaches [6,7] which are based on long-time records of weblogs.

The major contributions of our paper are summarized as follows:

(i) First, we propose a novel approach based on network traffic analysis to realize cross-device tracking. Our approach just sniffs the network traffic silently which does not require users' active participation. Also, we do not analyze the deterministic identifiers such as IM numbers, usernames, and network IDs.

(ii) Second, we extract the features of five popular IM apps' messages to identify sync messages and employ rule matching and machine learning to identify different apps' messages. We deal with real users' network traffic which includes other apps' network traffic and background network traffic interference.

(iii) Third, we utilize the SPRT algorithm to accelerate the speed of judging whether the devices are correlated or not, and we perform experiments to prove that our approach is effective and fast, which can solve the problem of cross-device tracking in a fine-grained scenario.

The rest of this paper is organized as follows. We introduce the architecture of an IM system and the cross-device message sync mechanism in Section 2. In Section 3, we present our cross-device tracking approach, including the tracking scenario, basic idea, and the detailed workflow of our approach. Section 4 shows the experimental results of our approach, which is followed by discussing future research directions and corresponding countermeasures in Section 5. The existing related work is introduced in Section 6, and a conclusion is drawn in Section 7.

2. Background

In this section, we introduce the architecture of an IM system and the cross-device message sync mechanism. Then, we present the observation of IM application network traffic.

2.1. Architecture of an IM System. Figure 1 shows a typical architecture of an IM system. An IM system consists of IM clients and different types of servers, e.g., authentication servers, file servers, gate servers, and route servers [8]. The IM clients are installed on devices (e.g., mobile phones and personal computers) by users. The authentication servers are used to verify the user identity. The route servers act as a message relay center to concatenate the connection between users and relay their messages. The gate servers are edge servers that maintain a persistent chatting connection with the IM clients and mainly relay messages on behalf of the IM client. To communicate with each other, an original IM client sends a request to the route server, requiring the latter to establish a connection to the target IM client. Since the route server knows the gate servers of these two IM clients, it can concatenate connections of the gate servers of the two IM clients and relay their messages. In addition, the file servers are used to store and relay the files shared between users as most of the IM systems support file sharing functionality.

2.2. Cross-Device Message Sync Mechanism. The modern IM system supports the cross-device message sync mechanism so that a user can keep an IM conversation after he switches from one device to another in a hurry. We take the scenario in Figure 1 as an example to illustrate how an IM system works. As shown in this figure, user A logs in the same IM account on two distinct devices (i.e., a mobile phone C_1 and a personal computer C_2), and user B employs a device C_3 to communicate with user A. To keep an IM client online after a user logs in the device, a gate server is responsible for maintaining a persistent connection between the gate server and the IM client and relaying IM messages. As a result, C_1, C_2, and C_3 establish persistent connections to the gate servers S_1, S_2, and S_3, respectively. Once user B intends to communicate with user A, a route server S_4 is used to establish connections to the corresponding gate servers so as to concatenate the connections among C_1, C_2, and C_3 and then forward messages on behalf of user A and user B. To achieve the cross-device message sync functionality, the route server is used to create and send cross-device sync messages to ensure that C_1 and C_2 can receive the same message almost at the same time. In particular, if a message m_1 is sent from C_3 and arrives at the route server S_4, the message is copied and forwarded to gate servers (i.e., S_1 and S_2) by S_4. In this way, user A can receive the message, referred to as the sync message, on both C_1 and C_2.

2.3. Observation of IM Application Network Traffic. We collect the network traffic of IM applications to observe the pattern of sync messages. In this paper, we mainly focus on 5 popular IM applications around the world, including WhatsApp, Facebook Messenger, WeChat, QQ, and Skype. We install these IM clients on two types of devices (i.e., mobile phones and personal computers) and require users to send and receive chatting messages so that we can capture the IM chatting traffic. We find that the messages are transmitted in the gate servers' flow one by one as the user sends them sequentially. As mentioned in [9, 10], one message consists of a series of successive packets with short time interval and similar packet number in one application. In addition, since network traffic is encrypted between clients and gate servers, we cannot derive the plaintext of messages from the network traffic.

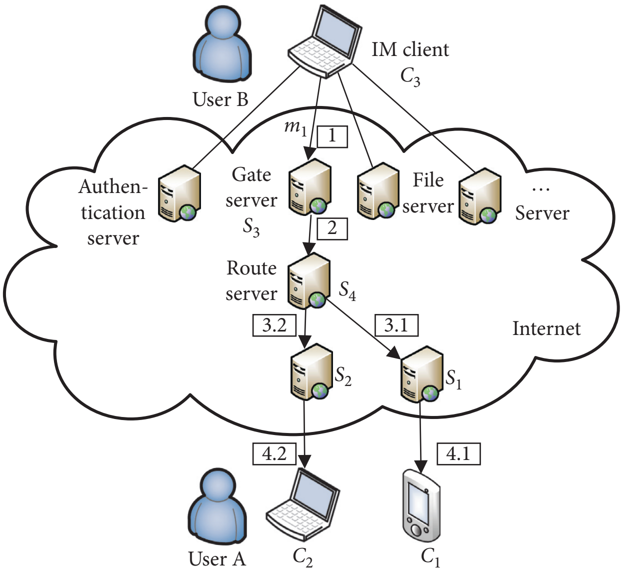


FIGURE 1: Architecture of an IM system.

3. Methodology

In this section, we first describe the assumption and basic idea of our cross-device tracking approach and then introduce the detailed approach, step by step. Table 1 summarizes the notations used throughout this paper.

3.1. Overview of the Cross-Device Tracking Approach. The law enforcement agency aims to correlate two devices used by a suspect so as to monitor his online behaviors. It is assumed that the suspect employs one personal computer (e.g., a desktop or a laptop) and one mobile phone to access the internet. It should be noted that we mainly focus on two kinds of devices in this paper, i.e., personal computers running Windows systems and mobile phones running Android systems. As Figure 2 shows, the suspect can operate his devices to access the internet via wired or wireless networks. The law enforcement controls the gateways of the network used by the suspect in order to passively record incoming and outgoing network traffic of the suspect. We also assume that the IP addresses of these devices do not change during a short period of time. In addition, we assume the suspect logs in the same IM application with one account

Table 1: Summary of notations.

Symbol	Meaning
$\uparrow$, $\downarrow$	An outgoing packet, an incoming packet
D, U	The set of devices, the set of users
X, Y	The set of personal computers, the set of mobile phones
X^k	The set of sync message timestamps of the kth PC
Y^ℓ	The set of sync message timestamps of the ℓth mobile phone
$Z^{k\ell}$	The set of sync message timestamps of the kth PC and the ℓth mobile phone successfully matched
$M_i^{k\ell}$	The result of matching sync messages in the ℓth mobile phone with the ith message of the kth PC
$\wedge_n$	The likelihood ratio
θ_0	The probability of sync message matching when two devices are not correlated
θ_1	The probability of sync message matching when two devices are correlated
η_u, η_l	The upper boundary, the lower boundary
α, β	The false-positive rate, the false-negative rate
T_m	The average time interval between user receiving sync messages
T_{interval}	The packet interval threshold
T_{offset}	The time offset of the network environment

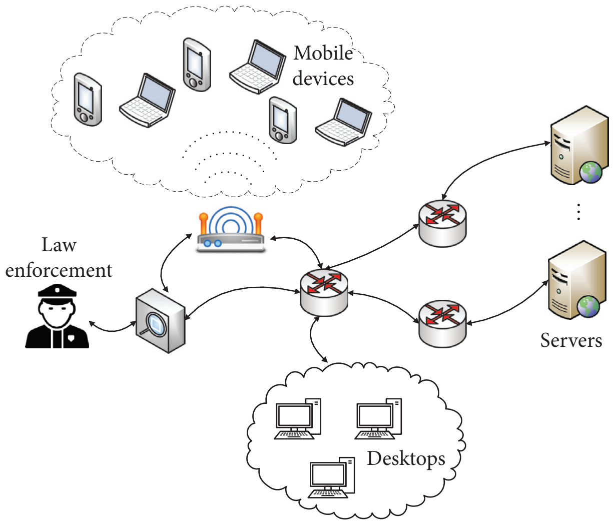


Figure 2: Cross-device tracking scenario.

on both the mobile device and the personal computer. As a result, when a user sends to the suspect a message, the IM applications on both distinct devices of the suspect will receive the message nearly at the same time due to the cross-device message sync mechanism. Since the sync messages are transmitted through the network, the law enforcement can passively identify two sync messages sent to these devices so as to link them. Last but not the least, we do not rely on the explicit identifiers, such as user IDs, in the IM messages to achieve cross-device tracking.

Figure 3 shows the workflow of our cross-device tracking approach. First, we capture and preprocess users' network traffic of mobile devices and personal computers. The network traffic is classified by IP addresses, and the device type (i.e., Windows personal computers and Android mobile phones) is identified according to the user-agent field of HTTP packets. Second, we collect ground-truth IM application (i.e., WhatsApp, Facebook Messenger, WeChat, QQ, and Skype) network traffic and analyze the features of gate server's flows. After that, we propose several filtering rules to identify the gate server's flows that contain sync messages. Third, we design experiments to analyze the attributes of sync messages and employ the rule matching and machine learning techniques to identify different applications' sync messages. After the identification of sync messages, we record their receiving timestamps to form the device sync message time list. At last, we compare the timestamps of sync messages between two devices and define the time gap that is less than 0.3 seconds as a successful match. Based on the matching results of sync messages of two devices, the sequential probability ratio testing (SPRT) algorithm is leveraged to make a quick decision of whether two devices are correlated.

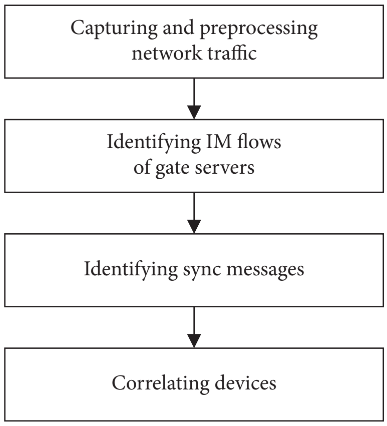

Capturing and preprocessing network traffic
Identifying IM flows of gate servers
Identifying sync messages
Correlating devices

Figure 3: Workflow of the cross-device tracking approach.

3.2. Capturing and Preprocessing Network Traffic. As shown in Figure 2, we can capture the network traffic from the gateway and wireless routers, respectively. After capturing network traffic, we first classify them into several groups in terms of the IP addresses of the devices. Then, we identify the devices as either an Android mobile phone or a Windows personal computer based on the HTTP traffic transmitted from these devices. Since HTTP packets are common in network traffic from both mobile phones and personal computers, we can use the value of the user-agent field in the HTTP packets as keywords to identify the device types. The user-agent field of HTTP packets from a Windows system often contains "Windows NT," while that from an Android system often includes "Android" (http://useragentstring.com/pages/useragentstring.php/). In addition, to reduce the noise, we remove the retransmitted packets from the captured traffic.

3.3. Identifying IM Flows of Gate Servers. We intend to identify the flows of IM gate servers that include the sync messages transmitted between gate servers and IM clients. To this end, we first collect a set of ground-truth IM network flows and analyze the communication mechanism of IM gate servers. We require a group of senders to use different IM clients to transmit messages to the other group of receivers who employ two devices to receive the sync messages. In particular, the senders use each IM client to transmit 20 messages with a time interval (e.g., 10 seconds). The receivers record the message timestamp upon obtaining a sync message. According to the timestamps, we can locate packets carrying the sync messages in network flows and treat these flows as the IM flows of gate servers. Then, we use the ground-truth flows to analyze the IP addresses and DNS request packets of the flows of gate servers. We find that the sync messages produced by the Windows version of QQ are transmitted by OICQ protocol, which is a kind of UDP packet that can be identified by using the DPI (deep packet inspection) technique. Meanwhile, the Android version of QQ and other IM applications employ TCP protocols to transmit sync messages between IM clients and gate servers. We find that gate servers of some IM applications own domain names, and others do not own. Therefore, there are two types of IM gate server flow recognition methods.

For the IM gate servers that have domain names, we collect a set of domain names of gate servers using the ground-truth flows as shown in Table 2. In the identification phase, we derive the DNS traffic of IM clients and use the domain name set to identify the flows of the gate server. It should be noted that the domain names in Table 2 are collected in our network within around 7 months. However, they may be changed due to the IM application update. In practice, the law enforcement can collect sufficient domain names in different network environments in advance.

For the IM gate servers that do not have domain names, we leverage reverse DNS lookup tools (e.g., Whois (https://www.whois.com/whois/)) to choose the candidate flow set and propose traffic filtering rules to identify the IM gate servers based on the IP addresses in the network flows. We discover that only gate servers of QQ and WeChat do not use domain servers, and both IM applications are developed by the Tencent company. Therefore, if the IP addresses of the servers belong to the Tencent company, we can save these flows. In addition, since gate servers of Tencent may use the network from different internet service providers (ISPs), we also save the traffic from the servers that belong to ISPs. Then, we summarize some features of the flows of gate servers to perform further traffic filtering:

(1) Since the persistent flows between IM clients and gate servers last longer than the others, we can discover the persistent flows in terms of the time period of the flows. Commonly, a message flow can last a few minutes or even longer, compared with ordinary flows that only last a few seconds. In practical, we use an empirical threshold (i.e., 1 minute) to identify potential IM persistent flows.

(2) According to the ground-truth traffic, we do not find any HTTP flows between IM clients and gate servers. Therefore, we can exclude the HTTP traffic that is the major traffic generated by user devices.

After employing these two filtering rules, there may be some candidate gate server's flows that we cannot uniquely identify. In this case, we can move to the next step to identify the sync messages for all candidate flows.

3.4. Identifying Sync Messages. After identifying IM flows of gate servers, we further analyze the ground-truth dataset of collected IM flows and extract the features of sync messages in order to identify them. According to our observation, each sync message transmitted from gate servers to the IM client corresponds to a series of successive packets with short packet time interval, while the interval between two messages is longer. Then, we can use an empirical packet interval threshold (i.e., T_{interval}, we conduct statistical analysis to derive the value of T_{interval} in Section 4) to automatically segment the IM flows to derive groups of packet sequences; these groups contain sync message packet group and background network traffic group. As mentioned before, in order to locate the sync message packet group, we record the timestamps when receiving the sync messages. We find that the packet pattern of sync messages from QQ, WeChat, WhatsApp, and Facebook Messenger can be extracted for sync message identification. However, the packet pattern of a sync message of Skype is different from the other four IM applications. Therefore, we try to employ a machine learning method to identify this kind of message (i.e., Skype).

We can use the packet direction pattern and packet length to identify the sync messages from the four IM gate servers (i.e., QQ, WeChat, WhatsApp, and Facebook Messenger). After the flow segmentation, the segmented packets from the PC client of QQ as well as PC and mobile clients of WeChat can be directly used to extract the packet direction pattern as there is no noise packet. However, we need to preprocess the segmented packets from the rest of the clients to remove the noise packets and derive the packet direction pattern. For the QQ mobile clients, we remove the packets, whose length is less than 100 bytes. For the traffic from clients of WhatsApp and Facebook Messenger, we apply the longest common subsequence (LCS) algorithm to sequences of packets so as to extract the common subsequence of the packet direction pattern for their sync messages. Then, the longest common subsequence can be used to detect the sync messages.

After deriving the packet direction patterns, we perform statistical analysis of the packet length and derive a specific

Table 2: Domain names of gate servers.

Application	Device type	Domain names
QQ	PC	*
	Mobile phone	*
WeChat	PC	long.weixin.qq.com
	Mobile phone	*
Skype	PC Mobile phone	(i) ip.wusmw1-client-s.msnmessenger.msn.com.akadns.net (ii) ip.ea2hk2-client-s.msnmessenger.msn.com.akadns.net
WhatsApp	PC	mmx-ds.cdn.whatsapp.net
	Mobile phone	(i) chat.cdn.whatsapp.net (ii) Whatsapp-chatd-edge-shv-02-hkg3.facebook.com
Facebook Messenger	PC (web)	star.c10r.facebook.com
	Mobile phone	(i) mqtt.c10r.facebook.com (ii) edge-mqtt-mini-shv-02-hkg3.facebook.com

packet length or a length range for each packet in the packet direction pattern to further reduce the false-positive rate of sync message identification. It is important to note that we discover that the packet direction patterns extracted from the PC client of WeChat as well as the mobile clients of WeChat and WhatsApp are useful enough to identify the sync message. Therefore, we do not use the packet length to identify the sync messages from these clients. For the WhatsApp PC clients, the range of the first packet length is used to decrease the false detection. Moreover, since the OICQ protocol used by the QQ PC clients is not encrypted, the deep packet inspection technique can be applied to further improve sync message identification. In particular, the fourth and fifth bytes of the two packets in the sequence are 0x17 and 0xce in hex.

Table 3 shows the features used to identify the sync messages. We denote "↑" and "↓" as an outgoing packet and an incoming packet, respectively. Then, packets of a sync message can be represented as a sequence of "↑" and "↓." The packet length range is denoted as (x, y). Then, we use "–" to concatenate the length range of each packet in order. For example, $(200, \text{MTU}) - 97$ indicates that the packet length of the first packet in the sequence is between 200 and MTU (maximum transmission unit), and the second packet length is 97. In addition, we denote "FPL" as the length of the first packet in the packet sequence.

As the packet pattern of the sync message of Skype is different from the other four applications, we employ machine learning methods to identify it. After segmenting the flows of the gate server into different bunches of the packet sequence, we process each bunch of the packet sequence to obtain statistical features which are shown in Table 4. There are four categories which contain 25 features in this table. We count the number of continuous packet subsequences whose directions are "↑↑," "↓↓," and so on. We also derive the statistical data (i.e., mean, standard deviation (STD), and maximum) of the packet length and packet time interval. In addition, we perform statistical analysis of the mean of the length and time interval of the first 1/3 packets, the second 1/3 packets, and the rest 1/3 packets, respectively. Finally, we choose to employ these features and the machine learning methods (i.e., XGBoost [11] and random forests [12]) to identify the packet sequence of the Skype sync message.

Once the sync message is identified, we choose the timestamp of the first packet in the preprocessed packet sequence as the sync message timestamp. After identifying all sync messages, we can derive a sequence of the timestamps of the sync messages for each device.

3.5. Correlating Devices.

We formalize the cross-device tracking problem in this section and elaborate on the theory of correlating devices. We denote two different types of devices (i.e., the mobile device and the personal computer) as D_1 and D_2 and denote the hypothesis "two devices are correlated" as H_1 and "two devices are not correlated" as H_0. Then, we can formalize H_1 and H_0 as

$$\begin{aligned} H_1&: D_1 \in U_a \text{ and } D_2 \in U_a,\\ H_0&: D_1 \in U_a \text{ and } D_2 \in U_b, \end{aligned} \tag{1}$$

where U_a and U_b represent different users. If two devices belong to the same user, we accept H_1; otherwise, we accept H_0.

We correlate two devices (i.e., a personal computer and a mobile phone) by matching their sequences of the timestamps of the sync messages. We assume that there are several users and they each have a personal computer and a mobile phone, respectively. We denote sequences of the sync message timestamps from the kth personal computer and the ℓth mobile phone as $X^k = \{t_1^k, \ldots, t_i^k, \ldots\}$ and $Y^\ell = \left\{t_1^\ell, \ldots, t_j^\ell, \ldots\right\}$, respectively. To match the sync message transmitted to two different types of devices, we compare the time difference between each identified sync message from a personal computer and all identified sync messages from all of the mobile phones. The ith sync message of the kth personal computer and the jth sync message of the ℓth mobile phone are matched if $|t_i^k - t_j^\ell| \leq \Delta t$, where Δt is an empirical value (i.e., 0.3 second) derived by our statistical analysis in Section 4. If we

Table 3: Features of four applications' sync messages.

Application	Device type	Packet direction	Packet length
QQ	PC	↓↑	(200, MTU)-97
	Mobile phone	↓↑↓	(200, 1000)-(200, 1000)-(100, MTU)
WeChat	PC	↓↑↓↑ or ↓↑↓↓↑	
	Mobile phone	↓↑↑↓	
WhatsApp	PC	↓↑ or ↓↑↓↓↑↓↑	FPL > 200
	Mobile phone	↓↑↑↓ or ↓↑↑↑	
Facebook Messenger	PC (web)	↓↑↑↓ or ↓↑↓	(100,MTU)-(0,100)-(0,100)-(0,100) or (100,MTU)-(87 or 94)-54
	Mobile phone	↓↑↑↓ or ↓↑↓	(200,MTU)-66-(54,MTU)-66 or (200,MTU)-(101 or 99)-66

Table 4: Features of Skype's sync messages.

Category	Features
Packet number	Total packet number, ratio of outgoing to incoming packet number, number of ↑, ↓, ↑↑, ↓↓, ↑↑↑, ↓↓↓, ↓↑↓, ↑↑↑↑, ↓↓↓↓, ↓↑↑↓
Packet direction	Direction of the first packet
Packet length	Mean, STD, max, mean (low 1/3), mean (mid 1/3), mean (high 1/3)
Packet time interval	Mean, STD, max, mean (low 1/3), mean (mid 1/3), mean (high 1/3)

have enough samples that match the formula above, we can correlate the two devices.

Next, we use the sequential probability ratio testing (SPRT) algorithm to determine how many sync messages are matched so that we can accept H_1 and reject H_0. We use a binary decision variable $M_i^{k\ell}$ to represent whether the ith message of the kth PC is matched with a sync message of the ℓth mobile phone. If matched, $M_i^{k\ell}$ is equal to 1. Otherwise, $M_i^{k\ell}$ is set as 0. After each sync message from the kth PC and the ℓth mobile phone is compared, we can derive a message-matching sequence, i.e., $\{M_1^{k\ell},\ldots,M_n^{k\ell}\}$. Then, we define $P(M_i^{k\ell}=1\mid H_1)=\theta_1$ and $P(M_i^{k\ell}=1\mid H_0)=\theta_0$, i.e., the probability of the ith message of the kth PC matched with a sync message of the ℓth mobile phone when the hypothesis of two devices correlated is true and false, respectively. Intuitively, the probability of two messages matched (i.e., θ_1) is high if the two devices are correlated. On the contrary, if the two devices are not correlated, the probability (i.e., θ_0) is low. Then, we assume $M_i^{k\ell}$ are independent and identically distributed (i.i.d.). We can calculate the likelihood ratio $\wedge_n$:

$$\begin{aligned}\wedge_n &= \ln\frac{\Pr\left[M_1^{k\ell},\ldots,M_n^{k\ell}\mid H_1\right]}{\Pr\left[M_1^{k\ell},\ldots,M_n^{k\ell}\mid H_0\right]}\\ &= \ln\frac{\prod_{i=1}^{n}\Pr\left[M_i^{k\ell}\mid H_1\right]}{\prod_{i=1}^{n}\Pr\left[M_i^{k\ell}\mid H_0\right]}=\sum_{i=1}^{n}\ln\frac{\Pr\left[M_i^{k\ell}\mid H_1\right]}{\Pr\left[M_i^{k\ell}\mid H_0\right]}.\end{aligned}\tag{2}$$

According to the SPRT algorithm, we calculate the likelihood $\wedge_i$ sequentially according to the values of $M_i^{k\ell}$ one by one until it reaches stopping boundaries [13, 14]. Then, we have

$$\wedge_i=\begin{cases}\wedge_{i-1}+\ln\dfrac{\theta_1}{\theta_0}, & M_i^{k\ell}=1,\\[2ex] \wedge_{i-1}+\ln\dfrac{1-\theta_1}{1-\theta_0}, & M_i^{k\ell}=0,\end{cases}\tag{3}$$

where $1\le i\le n$ and $\wedge_0=0$. It means that when $M_i^{k\ell}=1$, we add $\ln(\theta_1/\theta_0)$; otherwise, we add $\ln((1-\theta_1)/(1-\theta_0))$. The stopping rule is

$$\begin{cases}\wedge_i\ge\eta_u, & \text{stop and accept } H_1,\\ \wedge_i\le\eta_l, & \text{stop and accept } H_0,\\ \eta_l<\wedge_i<\eta_u, & \text{calculate } \wedge_{i+1},\end{cases}\tag{4}$$

where η_u and η_l represent an upper boundary and a lower boundary, respectively. This rule means that we need to calculate $\wedge_i$ until it reaches η_u or η_l. According to the theory of SPRT, η_u and η_l can be defined by

$$\begin{aligned}\eta_u &= \ln\frac{1-\beta}{\alpha},\\ \eta_l &= \ln\frac{\beta}{1-\alpha},\end{aligned}\tag{5}$$

where α and β are the user-chosen false-positive rate and false-negative rate, respectively. We use $\alpha\le 0.01$ and $\beta=0.01$ [14].

We need to calculate θ_0 and θ_1, respectively, to derive the likelihood $\wedge_n$. To compute θ_1, we collect the ground-truth dataset and conduct statistical analysis to derive the value of θ_1 in Section 4. We collect the ground-truth data of sync messages from several pairs of correlated devices and calculate the ratio of successful matching sync messages to all sync messages. Then, we set $\theta_1=0.9$ in our paper.

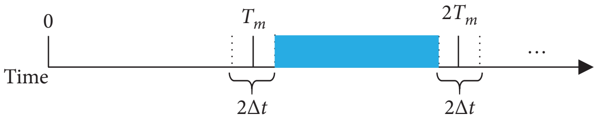


Figure 4: Sketch map of message matching.

We build a model to analyze the factors which impact θ_0 and then compute it. We can take time as a coordinate axis and the sync messages' timestamps of the devices as the coordinates. According to our statistical analysis, the average time interval between the user receiving sync messages equals T_m (i.e., $T_m \geq 10$), and the time gap between two matched sync messages is less than Δt. As shown in Figure 4, if one sync message timestamp of the mobile phone is located near the sync message timestamps of the PC (e.g., $[T_m - \Delta t, T_m + \Delta t]$), these two sync messages are matched. Otherwise, if it is located within the blue area, it does not match with a sync message of the PC. Then, we can calculate the probability of one mobile phone's message not matching with the personal computer's message, the probability being equal to the ratio of the blue area's length to the average time interval (i.e., $((T_m - 2\Delta t)/T_m)$). We denote the number of mobile phones as $|Y|$, and there are $|Y| - 1$ mobile phones which are uncorrelated with the PC. The probability of $|Y| - 1$ phones' message not matching with the personal computer's message is

$$\left(\frac{T_m - 2\Delta t}{T_m}\right)^{|Y|-1}. \tag{6}$$

Furthermore, we can calculate the maximum value of θ_0 which means the probability of at least one mobile phone's message matching with the personal computer's message:

$$\theta_{0\_\max} = 1 - \left(\frac{T_m - 2\Delta t}{T_m}\right)^{|Y|-1}. \tag{7}$$

According to this equation, if T_m is the minimum value (i.e., 10), $\theta_{0\_\max}$ gets the maximum value. And we have $\theta_0 \leq \theta_{0\_\max}$ normally.

In addition, as different networks have different latencies, we also find that most of time gaps between two sync messages may be bigger than Δt in some network environments. In order to deal with this situation, we define a time offset T_{offset}, which equals the average time delay of the sync messages between two devices. We require to collect the ground-truth dataset in the native network to measure T_{offset} in advance. Then, the sync message-matching interval converts to $[T_{\text{offset}} - \Delta t, T_{\text{offset}} + \Delta t]$. Specifically, T_{offset} has no effect on the calculation of $\theta_{0\_\max}$.

To show the matching rounds in one correlation, we can compute the expected number of sync message-matching with reference to Gu et al. [14] and Wald [5]. When we want to make a decision that two devices are correlated, the expected number of sync message-matching rounds we need to observe is

$$E[N \mid H_1] = \frac{\beta \ln(\beta/(1-\alpha)) + (1-\beta)\ln((1-\beta)/\alpha)}{\theta_1 \ln(\theta_1/\theta_0) + (1-\theta_1)\ln((1-\theta_1)/(1-\theta_0))}. \tag{8}$$

If we want to make a decision that two devices are not correlated, the expected number of matching rounds is

$$E[N \mid H_0] = \frac{(1-\alpha)\ln(\beta/(1-\alpha)) + \alpha \ln((1-\beta)/\alpha)}{\theta_0 \ln(\theta_1/\theta_0) + (1-\theta_0)\ln((1-\theta_1)/(1-\theta_0))}. \tag{9}$$

The detailed calculation process of these two formulas is in [5]. We analyze the possible values of the four parameters and substitute them into the formula and then we plot Figure 5 to show how $E[N \mid H_1]$ changes with the parameter change. When we fix $\beta = 0.01$ and vary α, θ_0, and θ_1, we can see that if θ_0 increases or θ_1 decreases, it demands more messages to make a decision of whether two devices are correlated. We assume that there are 8 people in one correlation (i.e., $|Y| = 8$), and we have $\theta_{0\_\max} \approx 0.35$. Therefore, when we set $\theta_1 = 0.9$, $\theta_0 = 0.35$, and $\alpha = 0.01$, we can get $E[N \mid H_1] \approx 7$ and $E[N \mid H_0] \approx 5$. It means that, after observing an average of 7 rounds of sync message-matching, we can make a decision H_1 (correlated); otherwise, we can accept H_0 (uncorrelated) after observing 5 rounds of sync message-matching.

Although the SPRT algorithm can help us to make a quick decision, there are still two cases that cannot be solved by employing it. One case is that if we detect more than one mobile phone correlated with a computer, then how to distinguish the right mobile phone? The other case is that the number of the sync messages of one device is not sufficient enough to make a decision (i.e., correlated or not). In order to deal with these cases, we employ the Jaccard index [15]. We denote the matched sync message set of the kth PC and the ℓth mobile phone as $Z^{k\ell}$ and the size of this set as $|Z^{k\ell}|$. As mentioned before, the sync message set of the kth personal computer and the ℓth mobile phone is denoted as X^k and Y^ℓ, respectively, and their size is denoted as $|X^k|$ and $|Y^\ell|$, respectively. According to the definition of the Jaccard index, we have

$$J\left(X^k, Y^\ell\right) = \frac{\left|Z^{k\ell}\right|}{\left|X^k\right| + \left|Y^\ell\right| - \left|Z^{k\ell}\right|}. \tag{10}$$

In order to ensure that the two devices have sufficient correlation, we only calculate the value of the Jaccard index if $Z^{k\ell} \geq 10$ and choose the biggest one as the final matching result.

4. Evaluation

In this section, we introduce the settings of our experiments and evaluate the performance of our cross-device tracking approach. We also perform the experiments to discuss the factors which can affect the performance of our approach.

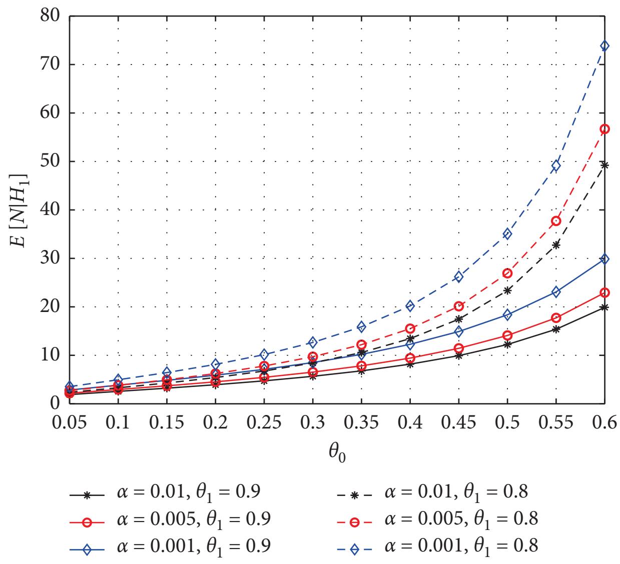


FIGURE 5: Values of $E[N \mid H_1]$ and $\beta = 0.01$.

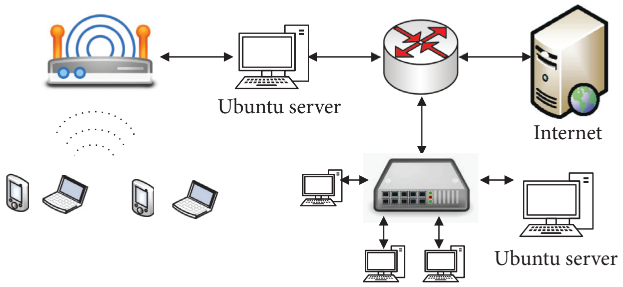


FIGURE 6: Experimental setup.

4.1. Experimental Setup. We recruit 23 participants and set up an experimental platform to connect their devices to the internet and capture the traffic. The experimental setup is shown in Figure 6. Desktops used by the participants are connected to a switch that connects to the internet via our campus network. We configure a mirror port of the switch which is connected by the first Ubuntu server used to capture traffic of the desktops. The mobile phones or laptops of the participants are connected to a wireless route that is set as a bridge mode. Then, the wireless route is connected to another Ubuntu server which has two network cards configured with a bridge mode. Since the traffic of the wireless devices passes through the second Ubuntu server, the server can capture all traffic of the wireless devices. The IP addresses of all the devices are assigned from a Dynamic Host Configuration Protocol (DHCP) server in the campus network. Therefore, we can use the IP addresses to identify each device in advance so as to collect the ground-truth traffic. We also install a Network Time Protocol (NTP) server [16] in the first Ubuntu server and synchronize the second Ubuntu server's time with that of the first one.

We require the 23 participants to use the five IM applications on their own devices so as to derive the ground-truth and testing traffic. All of the mobile phone OSes used by the participants are Android OSes, and all of the PC OSes are Windows OSes. The participants are required to chat with each other by logging in the same IM application both on a mobile phone and a computer. The IM applications include WhatsApp, Facebook Messenger, WeChat, QQ, and Skype. Four of the applications, except Facebook Messenger, have both Android and Windows versions. The participants use the Facebook Messenger chat widget on the Facebook website through a browser on their PCs. We capture the network traffic and save it in the pcap format.

4.2. Experimental Results. We first collect ground-truth data to derive the parameters which are mentioned in our approach. Then, we perform experiments to evaluate the effectiveness of different apps' sync message identification. After that, we implement device correlation experiments to verify our approach's ability in cross-device tracking. We also discuss the factors which can affect the results.

4.2.1. Parameter Settings. We first collect sync message ground-truth data and conduct experiments to derive the optimally empirical packet interval time threshold T_{interval} so as to accurately segment the sync message from the traffic. We invite 10 participants who are divided into five groups, and each participant has the same IM account logged in on two devices (e.g., a mobile phone and a personal computer). We let each participant employ each app to send 10 messages to their partner and let the receiver record the timestamps of the messages. At last, we obtain a ground-truth dataset of 100 pairs of IM sync messages with receiving time for each IM application. We locate the IM sync message packet sequences based on the recording timestamps and employ the time threshold to segment the traffic flow. We denote the correct segmentation as the sync message packet sequence is not separated, and the background traffic is separated from the sync message packet sequence. Figure 7 illustrates the relationship between different time intervals and the accuracy of the sync message segmentation. The accuracy is calculated by the number of correctly segmented sync messages divided by the number of all sync messages. As we can see from the figure, the accuracy approaches around 100% when T_{interval} is equal to 1.2 s. Therefore, we choose $T_{\text{interval}} = 1.2$.

We perform statistical analysis on the ground-truth data and derive an appropriate threshold Δt which is used to determine whether a sync message of a mobile phone is matched with a specific sync message of a PC and then derive $\theta_1 = 0.9$ in terms of Δt. To evaluate the value of Δt, we calculate the time gap between the timestamps of a pair of two sync messages from two devices according to the messages' timestamp in the network traffic. Figure 8 shows the cumulative distribution function (CDF) of 500 pairs of sync messages' time gaps collected from five IM applications. Since large Δt can lead to a number of false matches, we choose $\Delta t = 0.3$. Then, about 90% of the sync message pairs can be matched using this threshold, and we, therefore, can derive $\theta_1 = 0.9$.

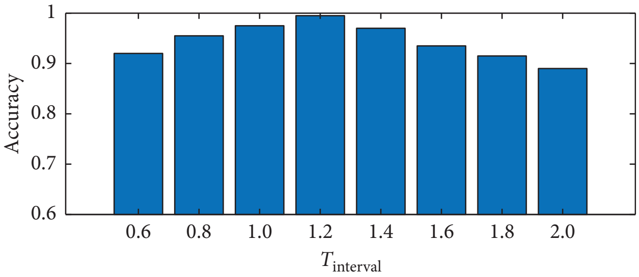


FIGURE 7: The result of segmenting sync messages at different time intervals.

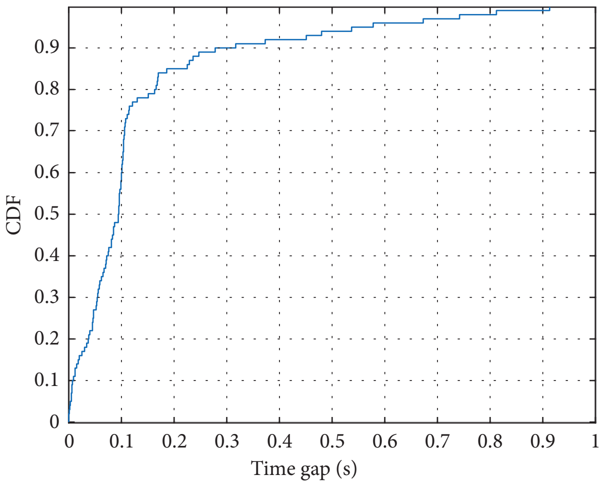


FIGURE 8: The CDF of the time gap between sync messages.

4.2.2. Sync Message Identification. In the next step, we collect testing data and evaluate the performance of sync message identification. In this set of experiments, we divide 10 participants into 5 groups and let them chat with his partner in the same group. We ask them to send 10 messages to their partner with a time interval (e.g., 10 seconds), which ensures that the network traffic of two sequential messages does not interfere with each other. The receivers record the timestamps of the received messages as the ground truth. Then, we obtain a dataset of 100 pairs of IM sync messages with receiving time for each IM application. At last, we evaluate our sync message identification method mentioned in Section 3.

To evaluate the performance of our sync message identification method, we define some evaluation indicators (i.e., precision, recall, and F_1-score). Precision is equal to the number of correctly identified sync messages divided by the number of all identified sync messages. Recall is equal to the number of correctly identified sync messages divided by the actual number of sync messages. F_1-score is

$$F_1 - \text{score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{11}$$

TABLE 5: Sync message identification result of four applications.

Device type	Indicator	QQ	WeChat	WhatsApp	Facebook Messenger
PC	Precision	94.3%	81.4%	95.3%	86.8%
	Recall	99.0%	96.0%	82.0%	92.0%
	F_1-score	0.966	0.881	0.882	0.893
Mobile phone	Precision	88.2%	90.5%	80.3%	88.3%
	Recall	97.0%	95.0%	94.0%	98.0%
	F_1-score	0.924	0.927	0.866	0.929

TABLE 6: Sync message identification result of Skype.

F_1-score	PC	Mobile phone
XGBoost	0.820	0.892
Random forest	0.786	0.796

Table 5 illustrates the F_1-score of the sync message identification method of QQ, WeChat, WhatsApp, and Facebook Messenger. As we can see from this table, QQ and WhatsApp have better performance on the PC version, while WeChat and Facebook Messenger have better performance on the phone version. From the perspective of the application type, the sync message identification performance of QQ is the best due to its special protocol and stable features. On the contrary, we find that the sync messages of WhatsApp are more difficult to be distinguished from background traffic which leads to a lower F_1-score. The other two apps have similar identification performance.

Table 6 depicts the F_1-score of the Skype sync message identification method. In order to identify Skype sync messages, we collect 400 sync messages of the PC and 400 sync messages of the mobile phone to train models. In addition, we preprocess our data used for XGBoost by a data standardization function "scale" in Python scikit-learn library. Then, we employ XGBoost and random forest as the classification methods. The performance of XGBoost is better than that of random forest, and we therefore choose XGBoost as the Skype sync message identification method in device correlation experiments.

4.2.3. Correlation Results. We evaluate our cross-device tracking approach and compare several scenarios to analyze different factors. Considering the situation of wireless router capacity and network speed, we set one experiment consisting of 8 people and 16 devices. When doing experiments, all participants connect their devices to our network and log in the designated apps. After they chat with each other for a while (e.g., 30 minutes), we analyze their network traffic to correlate their devices. In order to make a decision of whether two devices are correlated or not correlated (i.e., make a decision of accepting H_1 or H_0), we require to set four parameters (i.e., α, β, θ_0, and θ_1). Here, 8 people (i.e., $|Y| = 8$) lead to $\theta_{0\_\max} = 0.352$. Then, we set $\theta_1 = 0.9$, $\theta_0 = 0.35$, $\alpha = 0.01$, and $\beta = 0.01$, and after the calculation, we have $\ln(\theta_1/\theta_0) \approx 0.944$, $\ln((1-\theta_1)/(1-\theta_0)) \approx -1.872$, $\eta_u \approx 4.595$, and $\eta_l \approx -4.595$. In addition, we count the time length of correlating devices in our experiment to evaluate the speed of our approach. We also discuss some factors such

Table 7: Correlation result of five applications.

Correlation accuracy	QQ	WeChat	WhatsApp	Facebook Messenger	Skype
8 people	91.7%	97.9%	89.6%	85.4%	83.3%
16 people	80.2%	86.5%	80.2%	77.1%	72.9%

as different users, message frequency, and user amount which can make an influence on the result.

At last, we get 30 sets of data, with each app having 6 datasets. For each app, we have 8 device pairs in one experiment, and totally, we have 240 device pairs in these experiments. Every set of the experiment lasts dozens of minutes so that we can have enough time to correlate each device pair. The first row of Table 7 shows the correlation result of each application, noting a 12.5% baseline with 8 participants. The correlation accuracy is equal to the number of correctly correlated device pairs divided by the number of all device pairs. We can find that WeChat achieves the best performance with only one device pair matching failure in 48 pairs. QQ and WhatsApp achieve about 90% correlation accuracy. We also notice that the lowest correlation accuracy of Skype is more than 80%. Moreover, in these experiments, we let participants chat with other participants freely. It means the message-receiving frequency is not under control, and the traffic of sequential messages may mix with each other. This phenomenon results in the decrease of the sync message identification accuracy which further leads to the decrease of the correlation accuracy. From the result, we can conclude that WhatsApp, Facebook Messenger, and Skype are impacted by the factors mentioned above.

(1) Time Length. In order to study the distribution of matching time, we draw a boxplot of different device pairs' matching time of different applications in Figure 9. In this diagram, we can see that QQ and WeChat require less matching time compared with the other three applications and also have a relatively average time distribution, which is caused by two factors from our point of view. One is sync message matching accuracy; high matching accuracy leads to less matching time. The other is receiving message frequency, which is related to the number of user's friends. In our experiments, participants have more friends in QQ and WeChat, so they can receive more messages within the same length of time. The average matching time of QQ is about 290 seconds and WeChat is around 340 seconds, which means our cross-device tracking approach is fast enough. In addition, according to the theoretical expected number of matching rounds, the actual matching time is longer. We think there are two reasons: one is the actual message-receiving time interval is longer, and the other is misrecognition of sync messages leading to a lower message synchronization probability.

(2) Different Users. To make a comparison, when doing experiments, we group participants randomly, and each group consists of 8 people with 16 devices. The results of different groups are shown in Figure 10, with different colors meaning different groups; in a word, the differences between the groups are not significant. Therefore, our approach is generally applicable to different users.

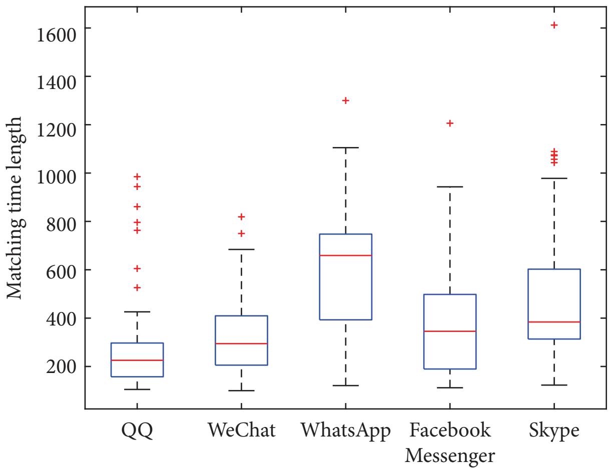


Figure 9: Matching time length of different applications.

(3) Message Frequency. In this experiment, we try to identify the impact of different message-receiving frequencies. Here, we choose WeChat and Skype as the apps to be tested. We do an 8-people experiment and let participants send messages to their partner in 5 minutes with different message frequencies (i.e., 15, 20, 25, and 30). Altogether, we have four groups of statistics from both WeChat and Skype. The correlation result is shown in Table 8. For WeChat, when the sync message frequency is relatively low (i.e., 15 and 20), we can correlate all 8 device pairs successfully, and while the sync message frequency is comparatively high (i.e., 25 and 30), there is one device pair linking failure. For Skype, the increasing message frequency decreases the correlation accuracy too. As far as we know, an increase in the message frequency leads to a rise in the probability of mismatching messages, which further results in the decrease of the correlation accuracy.

(4) User Amount. In this set of experiments, we simulate a scenario with more people to make a comparison with the 8-people experiment. In order to simulate the tracking scenario of more device pairs, we subtract each message list's initial time from the total message time and then put data from different batches together to link. To make a comparison with the 8-people experiment, we do new 8-people experiments and combine two of them to form a 16-people experiment; then, we also derive 30 datasets of the 16-people experiment and totally 480 device pairs in this experiment. In this set of experiments, 16 people (i.e., $|Y| = 16$) lead to $\theta_{0\_\max} \approx 0.606$. Therefore, we set $\theta_0 = 0.6$, $\theta_1 = 0.9$, $\alpha = 0.01$,

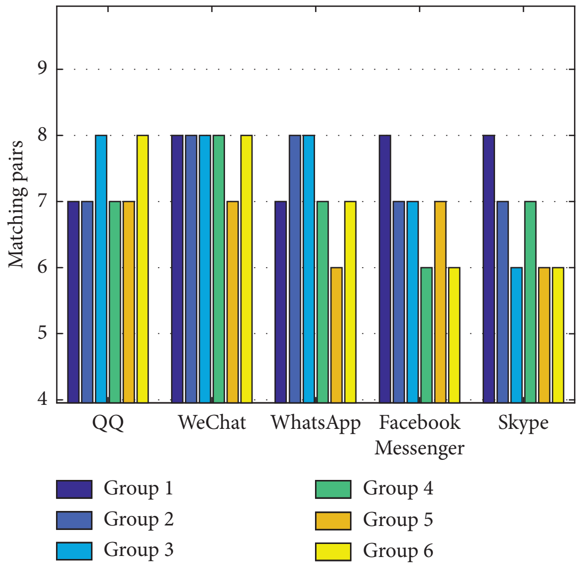


Figure 10: Matching result of different groups.

Table 8: Correlation result of different frequencies.

Correlation accuracy	15	20	25	30
WeChat	100%	100%	87.5%	87.5%
Skype	100%	87.5%	87.5%	75%

and $\beta = 0.01$, and thus, we have $\ln(\theta_1/\theta_0) \approx 0.405$, $\ln((1-\theta_1)/(1-\theta_0)) \approx -1.386$, $\eta_u \approx 4.595$, and $\eta_l \approx -4.595$ after the calculation. The correlation results are shown in Table 7, the statistics of the second row indicating that the accuracy of each app decreases about 10 percent compared with the first row. It is due to that, with the increase of the user amount, the possibility of sync message mismatching will increase, which results in the decrease of the linking accuracy.

5. Discussion and Future Work

In this section, we put forward some measures to promote our approach in the future and give some countermeasures for users to avoid devices being correlated.

5.1. Ways of Promoting Our Approach. In our experiments, we ask participants employing the same app to send messages over the same period of time, the purpose of which is to ensure the sufficient quantity of equipment so as to evaluate the performance of our approach. However, people are not likely to employ the same app to send instant messages at the same time in the real world. Therefore, we can separate different devices which do not have time overlap of network traffic and classify the devices which run different IM apps during the preprocessing period. After that, devices will be divided into different groups, which reduce the device number of one matching group, thus improving the correlation accuracy. It is important to note that our approach is universal for apps which have message synchronization function. Therefore, as long as the other IM app synchronizes messages between devices, our approach can correlate them. In particular, if the suspect employs a niche app which is out of our chosen apps and unpopular, it may cause the suspect's devices, generating the rare network traffic of the niche app in a local area network. In that case, we can correlate his different devices by identifying the network traffic of the niche app.

During our study, studying different message types is also helpful to promote our approach. As we know, the instant messaging app can send messages in various forms such as text, voice, picture, and file. If we can utilize the message type as an attribute in the period of sync message matching, it can reduce message mismatching. However, after analyzing the features of different messages, we find that there are some difficulties to employ them. One difficulty is that delivering a file or a picture message requires more than one server to work, i.e., one server sends a message notification first, and then another server sends the message content. Apart from that, different message-receiving strategies of devices also make them difficult to utilize these types of messages. For example, when the mobile phone version of QQ receives a voice message, it starts to receive the moment when the sender starts to record the voice. However, the PC version of QQ starts to receive when the sender stops the recording process. These difficulties will result in more complex identification of various messages and poorer message synchronization. In the future, we will figure out how to utilize different message types to correlate devices.

5.2. Countermeasures. In this paragraph, we give some advices to prevent the devices from being correlated. First, IM application manufacturers can modify message delivering packets to interfere sync message identification. They can add useless messages to the gate server's flow, add random packets to the message packet sequence, or add random padding to the message packets. Second, users can avoid using two devices to log in the same account, which is the simplest method from the aspect of users. However, users sometimes have to log in two devices at the same time. In that circumstance, they can choose not to keep online for a long period of time so that there will not be enough messages to correlate devices. Moreover, another way is choosing a suitable network connection mode, which means mobile phones can use 3G/4G to connect the internet to avoid their network traffic being captured.

6. Related Work

Cross-device tracking is developed from device tracking. In this section, we will introduce related research development.

According to the techniques employed in device tracking, there are two main categories, which are network traffic analysis-based tracking and web-based tracking. In the field of network traffic analysis, researchers devote to extracting attributes from network traffic to identify the device or users even though the device IP changes. In the physical layer, Polcak et al. [17] proposed a kind of attribute to identify computers by their clock skew computed from TCP timestamps. In the operating system layer, Franklin and McCoy [18] found that the 802.11 probe request time interval can reveal the active scanning algorithms employed by wireless derivers. In the network layer, the traffic bursts of mobile devices are counted to analyze the running applications [19]. In the application layer, the fields of unencrypted network traffic such as user-agent, IP address, cookies, and user IDs [20] are abstracted to identify devices. Furthermore, Gu et al. [21] found that the search history of the shopping website can also be used to track users. In addition, the DNS traffic can be used to analyze the device user's activities which can be used to track users [22].

Web tracking is based on acquiring attributes from the browsers to identify devices. It is widely used to serve the websites for statistics, tracking, and advertisement recommendation [23, 24]. Eckersley [25] first proposed this method in 2010 PETS. The author extracted the user-agent, HTTP header, screen size, fonts, and plugins from the browsers. Some researchers are interested in analyzing the tracking mechanism of commercial websites; Acar et al. [26] found that the canvas fingerprint is the most widely used web-tracking technique. Besides, Diaz et al. [27] proposed that the Battery Status API of HTML5 can be used to identify the browser. Compared to desktop browsers, the mobile device browsers, which lack plugins and some functions, therefore, achieve less information. It is not until in 2016 S&P conference that Laperdrix et al. [28] applied fingerprint identification technology to mobile terminal browser identification on a large scale. The results show that although plugin list and font information are lacking, the recognition rate still reaches 81%. With the further improvement of browser functions, Bojinov et al. [29] presented a method of achieving accelerometer data from the mobile device browser to realize identification. Then, Das et al. [30] employed the gyroscope and microphone to evaluate the effect of utilizing sensors to identify mobile devices.

Based on device tracking, cross-device tracking aims to correlate different types of devices to the same person. A recent study showed that most people operate multiple devices in daily use, and they often accomplish a task by switching devices (https://www.thinkwithgoogle.com/research-studies/the-new-multi-screen-world-study.html). The advertising companies need to collect the browsing history from different devices, which can help to recommend more comprehensive advertising. Cross-device tracking can be divided into two categories, which are deterministic tracking and probabilistic tracking [1]. Deterministic tracking requires the user to log in the same account on different devices. Then, the service provider is able to correlate different devices by the explicit identifiers such as account number and cookies. However, many apps can be used without logging in a user account, and the cookies probably cleared by users cannot be obtained each time.

Most researchers focus on probabilistic cross-device tracking. The feasibility of these methods is based on the similarity of user interests and activities when they operate different devices. For example, Kane et al. [2] found that 97.1% domains browsed on the mobile devices are also visited on the desktop. Also, 13.1% domains visited on the desktop are browsed on the mobile devices. During ICDM 2015 [3] and CIKM 2016 [4], the participants were asked to propose machine learning methods to do cross-device tracking. The datasets included many users' devices, cookies, IP addresses, and browsing history. These two competitions only focused on the algorithm design, such as using pairwise ranking [6, 7]. In the 2017 USENIX Security International Conference, Zimmeck et al. [1] indicated that online tracking is evolving from browser- and device-tracking to people-tracking. As one user tends to operate multiple devices, a person-centric view is established for cross-device tracking. In this paper, 126 users' browsing history of mobile devices and desktops was collected. The results showed that the IP address plays an important role in cross-device tracking. Choo [31] proposed a method of utilizing users' social media feeds to realize cross-device tracking. This method is based on the idea that each person has a distinctive social network, and thus, the links appearing in one's social media feeds are unique. However, this approach only works for a small number of social networks, which makes it impractical to be widely analyzed. In addition, there are some cross-device tracking approaches that do not analyze web communication, such as detecting inaudible ultrasonic sound embedded in websites [32]. Considering that using mobile phone's microphone demands user permission, Matyunin et al. [33] found that the gyroscope has a reaction to resonance frequencies in the frequency domain which means it is zero-permission tracking. Nevertheless, these approaches can only detect different devices which are physically close, and the devices are not necessarily owned by the same user. Moreover, some researchers paid attention to measure the cross-device tracking activity of commercial websites [34, 35] and found that cross-device tracking is widely used in many websites. Our work, which does not need long-time historical data, is a novel cross-device approach that is based on the network traffic analysis and easier to implement.

7. Conclusion

In this paper, we propose a novel cross-device tracking approach based on network traffic analysis. The premise of our approach is that users log in two devices with one IM account, and the two devices will receive messages simultaneously. We analyze the mechanism of devices' receiving sync messages and find that we can identify sync messages to correlate devices. In our scenario, we assume the law enforcement can sniff users' network traffic without users' active participation or long online time. Then, we extract features of five popular IM apps' received messages. In network traffic processing, we filter out useless flow and identify gate server's flow according to the server IP. After that, we employ rule matching and machine learning to identify sync messages. At last, we employ the SPRT algorithm to determine whether two devices are correlated according to the sync message-matching results. To evaluate our approach, we design contrast experiments with an 8-participant experiment and a 16-participant experiment. We find that the increasing participant amount will decrease the matching accuracy, and different apps achieve different matching accuracies, with WeChat getting the highest matching accuracy, which is 97.9% (8 participants) and 86.5% (16 participants) and Skype receiving the lowest matching accuracy, which is 83.3% (8 participants) and 72.9% (16 participants). We also study different users who make little impact on the results, while the increase of the message frequency will reduce the correlation accuracy. At the end of this paper, we discuss how to promote our approach in the future, and we give some advice to overcome this cross-device tracking problem.

Data Availability

The network traffic data used to support the findings of this study have not been made available because they contain a lot of privacy information.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Key R&D Program of China (nos. 2018YFB0803400, 2017YFB1003000, and 2018YFB2100300); the National Natural Science Foundation of China (nos. 61972088, 61632008, 61572130, 61532013, and 61702097); the Jiangsu Provincial Natural Science Foundation for Excellent Young Scholars (no. BK20190060); Jiangsu Provincial Key Laboratory of Network and Information Security (no. BM2003201); and the Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China (no. 93K-9).

References

[1] S. Zimmeck, J. S. Li, H. Kim, S. M. Bellovin, and T. Jebara, "A privacy analysis of cross-device tracking," in *Proceedings of the 26th USENIX Security Symposium*, pp. 1391–1408, Vancouver, Canada, 2017.

[2] S. K. Kane, A. K. Karlson, B. R. Meyers, P. Johns, A. Jacobs, and G. Smith, "Exploring cross-device web use on PCS and mobile devices," in *Proceedings of the International Conference on Human Computer Interaction*, pp. 722–735, Uppsala, Sweden, 2009.

[3] ICDM, "Drawbridge cross-device connections," 2015, https://www.kaggle.com/c/icdm-2015-drawbridge-cross-device-connections/overview.

[4] C. Cup, "Track 1: cross-device entity linking challenge," 2016, http://cikm2016.cs.iupui.edu/cikm-cup/.

[5] A. Wald, *Sequential Analysis*, Dover Publications, Mineola, TX, USA, 2004.

[6] J. Walthers, "Learning to rank for cross-device identification," in *Proceedings of 2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 1710–1712, Atlantic City, NJ, USA, 2015.

[7] M. C. Phan, A. Sun, and Y. Tay, "Cross-device user linking: URL, session, visiting time, and device-log embedding," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 933–936, Tokyo, Japan, 2017.

[8] R. B. Jennings, E. M. Nahum, D. P. Olshefski, D. Saha, Z.-Y. Zon-Yin Shae, and C. Waters, "A study of internet instant messaging and chat protocols," *IEEE Network*, vol. 20, no. 4, pp. 16–21, 2006.

[9] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "Appscanner: automatic fingerprinting of smartphone apps from encrypted network traffic," in *Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 439–454, Saarbrücken, Germany, 2016.

[10] N. Guo, J. Luo, Z. Ling, M. Yang, W. Wu, and X. Fu, "Your clicks reveal your secrets: a novel user-device linking method through network and visual data," *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8337–8362, 2019.

[11] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 785–794, San Francisco, CA, USA, 2016.

[12] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[13] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan, "Fast portscan detection using sequential hypothesis testing," in *Proceedings of the 2004 IEEE Symposium on Security and Privacy*, pp. 211–225, Berkeley, CA, USA, 2004.

[14] G. Gu, J. Zhang, and W. Lee, "Botsniffer: detecting botnet command and control channels in network traffic," in *Proceedings of the 2008 Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, 2008.

[15] P. Jaccard, "Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines," *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, pp. 241–272, 1901.

[16] D. Mills, *Computer Network Time Synchronization-The Network Time Protocol*, CRC Press, Boca Raton, FL, USA, 2006.

[17] L. Polcak, J. Jirasek, and P. Matousek, "Comment on "remote physical device fingerprinting"" *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 5, pp. 494–496, 2014.

[18] J. Franklin and D. McCoy, "Passive data link layer 802.11 wireless device driver fingerprinting," in *Proceedings of the 15th USENIX Security Symposium*, Vancouver, Canada, 2006.

[19] T. Stöber, M. Frank, J. B. Schmitt, and I. Martinovic, "Who do you sync you are?: smartphone fingerprinting via application behaviour," in *Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WISEC)*, pp. 7–12, Budapest, Hungary, 2013.

[20] T. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi, "Host fingerprinting and tracking on the web: privacy and security implications," in *Proceedings of the 2012 Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, 2012.

[21] X. Gu, M. Yang, C. Shi, Z. Ling, and J. Luo, "A novel attack to track users based on the behavior patterns," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 6, 2017.

[22] D. Herrmann, C. Banse, and H. Federrath, "Behavior-based tracking: exploiting characteristic patterns in DNS traffic," *Computers & Security*, vol. 39, pp. 17–33, 2013.

[23] J. R. Mayer and J. C. Mitchell, "Third-party web tracking: policy and technology," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, pp. 413–427, San Francisco, CA, USA, 2012.

[24] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, "Cookieless monster: exploring the

ecosystem of web-based device fingerprinting,” in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, pp. 541–555, Berkeley, CA, USA, 2013.

[25] P. Eckersley, “How unique is your web browser?” in *Proceedings of the 10th International Symposium on Privacy Enhancing Technologies (PETS)*, pp. 1–18, Berlin, Germany, 2010.

[26] G. Acar, C. Eubank, S. Englehardt, M. Juárez, A. Narayanan, and C. Díaz, “The web never forgets: persistent tracking mechanisms in the wild,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 674–689, Scottsdale, AZ, USA, 2014.

[27] C. Diaz, L. Olejnik, G. Acar, and C. Casteluccia, “The leaking battery: a privacy analysis of the html5 battery status api,” *Lecture Notes in Computer Science*, vol. 9481, pp. 254–263, Springer, Berlin, Germany, 2015.

[28] P. Laperdrix, W. Rudametkin, and B. Baudry, “Beauty and the beast: diverting modern web browsers to build unique browser fingerprints,” in *Proceedings of the 2016 IEEE Symposium on Security and Privacy (S&P)*, pp. 878–894, San Jose, CA, USA, 2016.

[29] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh, “Mobile device identification via sensor fingerprinting,” 2014, https://arxiv.org/abs/1408.1416.

[30] A. Das, N. Borisov, and M. Caesar, “Tracking mobile web users through motion sensors: attacks and defenses,” in *Proceedings of the 23rd Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, 2016.

[31] Y. S. Choo, *Cross-device tracking of employees with social networks*, Ph.D. dissertation, Imperial College London, London, UK, 2017.

[32] K. Waddell, “Your phone is listeningłliterally listeningłto your TV,” 2015, https://www.theatlantic.com/technology/archive/2015/11/yourphone-is-literally-listening-to-your-tv/416712/.

[33] N. Matyunin, J. Szefer, and S. Katzenbeisser, “Zero-permission acoustic cross-device tracking,” in *Proceedings of 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 25–32, Washington, DC, USA, 2018.

[34] J. Brookman, P. Rouge, A. Alva, and C. Yeung, “Cross-device tracking: measurement and disclosures,” *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 2, pp. 133–148, 2017.

[35] K. Solomos, P. Ilia, S. Ioannidis, and N. Kourtellis, “Talon: an automated framework for cross-device tracking detection,” 2019, https://arxiv.org/abs/1812.11393.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8869096, 9 pages
https://doi.org/10.1155/2020/8869096

Research Article

A Robust Image Watermarking Approach Using Cycle Variational Autoencoder

Qiang Wei,[1] Hu Wang,[2] and Gongxuan Zhang[1]

[1] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China*
[2] *Beijing Mysleepart Technology Co., Ltd., Beijing, China*

Correspondence should be addressed to Gongxuan Zhang; gongxuan@njust.edu.cn

Received 7 July 2020; Revised 26 August 2020; Accepted 7 September 2020; Published 22 September 2020

Academic Editor: Xiaolong Xu

With the rapid development of Internet and cloud storage, data security sharing and copyright protection are becoming more and more important. In this paper, we introduce a robust image watermarking algorithm for copyright protection based on variational autoencoder networks. The proposed image watermarking embedding and extracting network consists of three parts: encoder subnetwork, decoder subnetwork, and detector subnetwork. In the training process, the encoder and decoder subnetworks learn a robust image representation model and further implement the embedding of 1-bit watermark image to the cover image. Meanwhile, the detector subnetwork learns to extract the 1-bit watermark image from the embedding image. Experimental results demonstrate that the watermarked images generated by the proposed algorithm have better visual effects and are more robust against geometric and noise attacks than traditional approaches in the transform domain.

1. Introduction

In the era of big data and cloud computing, especially with the rapid development of mobile edge computing (MEC) technology, the demand for real-time services from a wide range of mobile terminals and commercial services providers (CSPs) is more and more urgent. On the one hand, many MEC-based services have been provided, such as paper citation network based link prediction and paper recommendation [1, 2], electricity load forecasting [3], and energy efficient dynamic offloading [4]. To fulfill real-time responses of MEC-based services, workflow scheduling and management are very important. In [5–8], many workflow scheduling approaches under different systems and environments (i.e., NSGA-II, edge computing environment, cyber-physical cloud systems, etc.) have been proposed. However, on the other hand, whether in the stage of data collection or application, people can access the required multimedia resources more easily than before, which will pose a serious threat to the privacy and copyright protection of those multimedia resources [9].

Privacy protection and authentication technologies can be divided into two categories. One is at the system level, which means the recommendation algorithms deployed in the service system (i.e., LSH-based recommender systems, multidimensional service recommendation, etc.) can avoid the users' request for obtaining the privacy information [10–13]. The other is at the data level, known as active authentication technology. In this kind of technology, digital image watermarking technology has become an important means of copyright protection of image resources. However, the problems of geometric attack resistance and balance between robustness and imperceptibility are still common problems in the field of digital image watermarking research.

Traditional image watermarking algorithms are often implemented in the transform domain; that is, image is firstly transformed into frequency or spatial-frequency domain (e.g., discrete cosine transform or wavelet transform). Then, appropriate coefficients in transform domain are selected for embedding watermark images. Finally, the modified transform domain coefficients, which are embedded watermarking information, are transformed back to the spatial domain to derive the watermarked digital images [14–16]. Although the watermarked images generated by this kind of approaches have good visual effect, they are not robust against geometric and noise attacks.

In recent years, some studies have introduced deep learning and adversarial learning into the field of watermarking and steganography. For instance, Volkhonskiy et al. have proposed a Steganographic Generative Adversarial Networks (SGAN) model [17], which, for the first time, incorporates the GAN and adversarial learning with information steganography technology. In this approach, an additional information embedding module was added on the basis of the original generative network to produce pseudonatural images after embedding information. Meanwhile, a steganalysis discriminant network is trained to discriminate the original natural image and the watermarked images generated by the generator. Under this framework, Shi et al. used Wasserstein GAN (WGAN) to optimize the training procedure and make the generated watermarked image more realistic with better visual quality [18]. Based on additive distortion cost function, Tang et al. firstly proposed the concept of automatic steganographic distortion learning (ASDL) model, which is called ASDL-GAN [19]. In this algorithm, the probability matrix P of image pixel modification is obtained by deep learning, and then the Syndrome-Trellis Codes (STC) method is used for information embedding. However, in this kind of GAN-based method, the discriminator is only used to distinguish whether the generated image contains hidden information or not and the quality of the generated image is not evaluated. That means its essence is to judge whether the probability distribution in the parameter space of the natural image or generated image is distinguishable. So, it cannot guarantee the visual quality of the generated image. Therefore, Mun et al. proposed a watermark network (WM-Net), which directly uses convolutional neural network (CNN) to fulfill the robust image watermarking and improves the antiattack ability of the watermarking embedding network by adding geometric attacks during the training process of the network [20]. However, the proposed CNN model does not contain any loss function to evaluate the quality of recovered watermark image either.

Therefore, in this paper, we propose a robust image watermarking embedding algorithm based on cycle variational autoencoder (Cycle-VAE) networks. One advantage of VAE model is that it can learn an abstract representation of a particular kind of images (such as face images). Furthermore, we use a convolution network similar to that in [20] to embed a 1-bit watermarking image into the cover image in the representation space. Although this strategy is similar to the WM-Net, they have two main differences. On the one hand, in the WM-Net, quaternion discrete Fourier transform (QDFT) is used before the watermark embedding, which is a fixed transform. But in the Cycle-VAE model, the network tries to learn an image transform that is more suitable for information embedding. On the other hand, in the WM-Net, the images should be partitioned into image blocks before performing the QDFT, like DCT-based watermark techniques. This will affect the ability of watermark algorithm for antigeometric attack. However, our proposed network can deal with the image entirely. In addition, because the dimension of image in the abstract representation space is usually not too high, the embedding and extraction network of watermark can also be small. Finally, to ensure the balance between the reality of the watermarked image and the reliability of the extracted watermark, we adopt a similar mechanism as Cycle Generative Adversarial Network (CycleGAN) [21]. In cycle A, an image is transformed to the representation space via encode network, after watermark embedding, and then transformed back to the image space via decode network. The loss function constrains the consistency between the input and watermarked image. Meanwhile, in cycle B, a watermark is embedded in the representation space by the embedding network, after transforming to the image space and back to the representation space again, and then extracted by the detection network. The loss function constrains the consistency between the input and recovered watermark. A demonstration of the above flow chart is shown in Figure 1.

This paper will be presented by the following parts. Section 2 gives an overview of the related works about CycleGAN and VAE approaches. Next section describes the proposed Cycle-VAE model for image watermarking, including the network structures, loss function, and implementation details. Section 4 has shown the results of robustness of our proposed Cycle-VAE model under geometric and noise attacks. In the end, the conclusion is presented in Section 5.

2. Related Works for VAE and CycleGAN

Currently, there are mainly two popular generation models: Generative Adversarial Nets (GAN) [22] and Variational Automatic Encoder (VAE) [23] and variants based on these two models. In GAN model, a generative model G and a discriminant model D are trained simultaneously. The generative model G captures the distribution of data, while the discriminant model D distinguishes the probability that the sample comes from the training data set rather than from the model G generated. However, there are some drawbacks in the GAN model. For example, it needs to find Nash equilibrium in the training process, which is much more difficult than optimizing an objective function. In addition, it uses a noise z as a prior, but the generative model G cannot control the noise z. That is, the training procedure of GAN is too free, which makes the training process and results of GAN uncontrollable with lack of robustness. In order to stabilize the training process of GAN, researchers have proposed many training techniques from the perspective of model improvement and theoretical analysis, such as Wasserstein GAN (WGAN) [24] and Least Square GAN (LS-GAN) [25].

In addition to the GAN model, Automatic Encoder Neural Network (AENN) is another unsupervised learning algorithm which can be trained by Back Propagation (BP) algorithm [23]. Its biggest characteristic is that the input and output are constrained to be consistent. In fact, a simple self-encoder is a low-dimensional representation of learning data

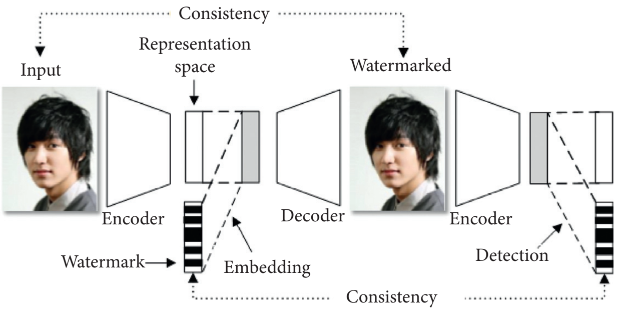


Figure 1: Framework of Cycle-VAE model for image watermarking.

sets, which is similar to Principle Component Analysis (PCA), except that PCA is linear, while self-encoder is nonlinear. However, the performance of standard automatic encoder is limited, mainly because the distribution of output vectors in the hidden layer is unknown and chaotic. Therefore, Kingma and Welling introduced the Variational Automatic Encoder (VAE) [23, 26]. It introduced a hidden variable Z in the hidden layer of standard autoencoder. Through the hidden variable Z, it can generate data automatically and combine the viewpoint of deep learning with that of statistical learning. Besides generating data, VAE can also provide an effective nonlinear data representation approach.

Furthermore, in the image watermarking task, besides an effective data representation approach, we also need to transform the image from the spatial domain to the transform domain that fits for embedding watermarks, which is similar to the image transfer between different domains. The concept of image-to-image translation was first proposed by Hertzman et al. [27]. For pair-matched dataset, several approaches have been proposed to learn a parametric translation function with the help of deep convolutional neural networks in recent years. However, for most real application scenarios, pair-matched data is scarce. To deal with this lack, CycleGAN is a famous model for unpaired image-to-image translation [21]. It is developed from the Conditional GAN (cGAN) [28] and Coupled GAN (CoGAN) [29] with the cycle-consistency loss and its ability of unpaired translation has been proved by many experiments. UNIT-like models [30, 31] are another series of unsupervised image-to-image translation models. They observe the hypothesis of latent space, combine the VAE with CoGAN, and use different codes to represent images content or style. In addition, for another kind of image translation when images are translated from simple to complex or vice versa, Dou et al. proposed an Asymmetric CycleGAN model [32, 33] for improving the CycleGAN model on image translation between domains with different complexity.

This kind of model improves the interpretability of translation model, but it also brings about higher optimization complexity. Therefore, in this paper, we propose a cycle variational autoencoder model to translate spatial domain images into a representation domain and fulfill the watermark embedding, which have low optimization complexity and are robust against noise and geometrical attacks.

3. Proposed Cycle-VAE for Image Watermarking

In this section, we demonstrate our proposed Cycle-VAE model whose goal is to learn a representation space that is suitable for image watermark embedding. To facilitate further illustration of our model, we denote the transformation from image domain to the representation domain as encoder E_I and the transformation from representation domain back to image domain as decoder D_I. The "representation space" or "representation domain" mentioned here denotes the representation feature space in the encoder or decoder network because the explicit explanation of features extracted by the networks is really difficult. In addition, for watermark embedding and detection, we denote the embedding network as E_W and the detection network as D_W. We use $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{m}$ to denote the original image, representation coefficients, and the watermark, respectively. Also we use $\widehat{\mathbf{x}}$, $\widehat{\mathbf{y}}$, and $\widehat{\mathbf{m}}$ to denote the watermarked image, the watermarked representation coefficients, and the detected watermark, respectively. That is, during each step of embedding and detection of the watermark, we have $\widehat{\mathbf{m}}, \widehat{\mathbf{y}} = E_W(\mathbf{m} \mid \mathbf{y})$, $\widehat{\mathbf{x}} = D_I(\widehat{\mathbf{y}})$, and $\widehat{\mathbf{m}} = D_W(E_I(\widehat{\mathbf{x}}))$. As illustrated in Figure 1, our model includes two cycle-consistency losses to constrain the distortion between the original and watermarked images, as $\mathbf{x}$ and $\widehat{\mathbf{x}}$, and between the original and detected watermarks, as $\mathbf{m}$ and $\widehat{\mathbf{m}}$, respectively. More detailed discussion about the model structure and implementations is in the following subsections.

3.1. Model Structure of Cycle-VAE. As shown in Figure 2, our watermarking framework consists of two cycles: an image transformation cycle with encoder and decoder networks, E_I and D_I, respectively, and a watermark embedding cycle with embedding and detection networks, E_W and D_W, respectively. In [34], some theoretical analyses and suggestions on disentangling factors of variation with cycle-consistent structures for variational autoencoders have been provided. Here, in the image transformation cycle (denoted as cycle A), we use VAE loss and identity loss to train E_I and D_I to be an image representation that is suitable for hiding watermark information. In the watermark embedding cycle (denoted as cycle B), we use image and watermark identity loss to enforce the ability of E_W and D_W for watermark hiding and detection, respectively.

In **cycle A**, our image encoding and decoding networks roughly follow the architectural guidelines set forth by [35]. We replace the pooling layers in [35] by using strided convolutions for in-network downsampling and upsampling. Our encoder network E_I comprises five residual blocks [36] with stride 2 convolution, and all nonresidual convolutional layers are followed by batch normalization [37] and ReLU activation layers. All the convolutional layers use 3 × 3 kernels. Therefore, for the encoder network E_I, the

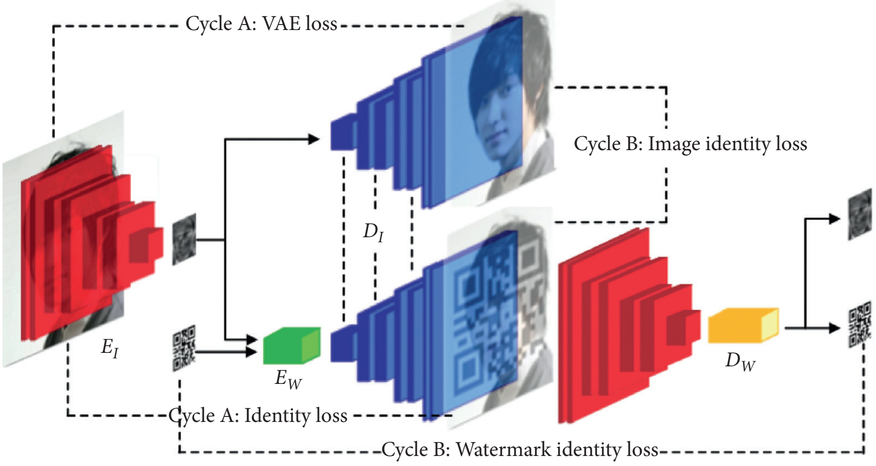

Cycle A: VAE loss
Cycle B: Image identity loss
D_I
E_I
E_W
D_W
Cycle A: Identity loss
Cycle B: Watermark identity loss

Figure 2: A detailed framework of Cycle-VAE model structure for image watermarking.

input and output are color images with shape of 3 × 128 × 128 and representation coefficients with size of 36 × 32 × 32, respectively. Furthermore, the corresponded decoder network D_I consists of 6 upsampling blocks, and each block contains an upsampling layer and a convolutional layer, followed by batch normalization, except for the final output layer, which uses scaled tanh to ensure that the output image has pixels with value between 0 and 255. For the upsampling layer, we use bilinear upsampling with the parameter of scale factor set to be 2. For the convolutional layer, the stride and kernel size are set to be 1 and 3 × 3, respectively. So, the output of decoder network is a three-channel color image with size of 128 × 128.

The loss function in **cycle A** contains a VAE loss and an identity loss of images. Considering some dataset $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ consisting of N i. i. d samples of some continuous or discrete variable $\mathbf{x}$, we assume that the data are generated by some random process, involving an unobserved continuous random variable $\mathbf{y}$. From a coding theory perspective, the unobserved variables y have an interpretation as a latent representation or code. In this paper, VAE is specified by a parametric generative model (as decoder) $p_{D_I}(\mathbf{x}\,|\,\mathbf{y})$ of the visible variables given the latent variables, a prior $p(\mathbf{y})$ over the latent variables, and an approximate inference model (as encoder) $q_{E_I}(\mathbf{y}\,|\,\mathbf{x})$ over the latent variables given the visible variables. Then, the marginal likelihood $\log p_{D_I}(\mathbf{x})$ can be rewritten as [26]

$$\begin{aligned}\log p_{D_I}(\mathbf{x}) \geq & -KL\left(p_{E_I}(\mathbf{y}\,|\,\mathbf{x}), p(\mathbf{y})\right) \\ & + E_{q_{E_I}(\mathbf{y}\,|\,\mathbf{x})}\log p_{D_I}(\mathbf{x}\,|\,\mathbf{y}),\end{aligned} \tag{1}$$

where the right-hand side is called the variational lower bound or evidence lower bound (ELBO). However, in general, this lower bound is unattainable. So, when performing maximum-likelihood training, our goal is to optimize the marginal log-likelihood.

$$\arg\max_{D_I,E_I} E\left[\log p_{D_I}(\mathbf{x})\right]. \tag{2}$$

Unfortunately, computing $\log p_{D_I}(\mathbf{x})$ requires marginalizing out $\mathbf{y}$ in $\log p_{D_I}(\mathbf{x},\mathbf{y})$, which is usually intractable. Thus, based on the inequality in equation (1) and the assumptions used in [23], with the variational Bayes algorithm, our VAE loss can be converted to the following optimization problem:

$$\begin{aligned}\min_{D_I,E_I} \ell_{\text{VAE}}(\mathbf{x},\mathbf{y}) = E\Big[&KL\left(p_{E_I}(\mathbf{y}\,|\,\mathbf{x}), p(\mathbf{y})\right) \\ & - E_{q_{E_I}(\mathbf{y}\,|\,\mathbf{x})}\log p_{D_I}(\mathbf{x}\,|\,\mathbf{y})\Big].\end{aligned} \tag{3}$$

Because of inequality (1), we still optimize a lower bound to the true maximum-likelihood objective (2). In addition to the VAE loss, which is used for training a good representation of images, we hope that the decoder network D_I can also have the ability of hiding watermarks. So, the identity loss between decoded image and decoded watermarked image is used, which is the squared Frobenius norm of the difference between these two images:

$$\min_{D_I,E_I} \ell_{\text{Ident}-I1}(\mathbf{x},\mathbf{m}) = \left\|\mathbf{x} - D_I\left(E_W\left(\mathbf{m}\,|\,E_I(\mathbf{x})\right)\right)\right\|^2. \tag{4}$$

Therefore, during the training process in cycle A, the encoder and decoder networks, E_I and D_I, are generated by solving the problem

$$\left\{\widehat{E_I}, \widehat{D_I}\right\} = \arg\min_{D_I,E_I}\left\{\ell_{\text{VAE}}(\mathbf{x},\mathbf{y}) + \lambda_1\ell_{\text{Ident}-I1}(\mathbf{x},\mathbf{m})\right\}. \tag{5}$$

In **cycle B**, for the embedding network E_W, we simply use 3 blocks, and each contains a 3 × 3 convolutional layer with padding and stride of 1 and a 1 × 1 convolutional layer. The input of embedding network includes an image representation coefficients vector with size of 36 × 32 × 32 and a 1-bit watermark image with size of 32 × 32. The 1-bit watermark image means the value of pixels in watermark can only be 0 or 1. So, we concatenate the watermark to the image coefficients as an additional channel. Then, the watermark and coefficients are sent to the embedding network with an output with size of 36 × 32 × 32 as the watermarked coefficients. For the detection network D_W, we also use a 3-block convolutional neural network but with each block containing a 1×1 convolutional layer, a 3×3 transpose convolutional layer, and a batch normalization. Finally, to keep the output pixel at 0 or 1, we add a sigmoid activation at the last layer of the detection network. The output of detection network is a one-channel binary image with size of 32 × 32.

The loss function in **cycle B** contains two identity losses: one is for watermarked image and the other is for detected watermark. The identity loss for images is used to train the embedding network E_W for hiding the watermark to a specified image, which is the squared Frobenius norm of the difference between the images with and without a watermark:

$$\min_{E_W} \ell_{\text{Ident}-I2}(\mathbf{y},\mathbf{m})\left\|D_I(\mathbf{y}) - D_I\left(E_W(\mathbf{m} \mid \mathbf{y})\right)\right\|^2. \tag{6}$$

The identity loss for watermark is used to train the detection network D_W for detecting the watermark from embedded image representation coefficients, which is the squared Frobenius norm of the difference between the original watermark and the detected watermark:

$$\min_{D_W} \ell_{\text{Ident}-W}(\widehat{\mathbf{x}},\mathbf{m})\left\|\mathbf{m} - D_W\left(E_I(\widehat{\mathbf{x}})\right)\right\|^2. \tag{7}$$

Thus, in the training process of cycle B, the embedding and detection networks, E_W and D_W, are generated by minimizing the following objective function:

$$\left\{\widehat{E_W}, \widehat{D_W}\right\} = \arg \min_{E_W, D_W} \left\{\ell_{\text{Ident}-I2}(\mathbf{y},\mathbf{m}) + \lambda_2 \ell_{\text{Ident}-W}(\widehat{\mathbf{x}},\mathbf{m})\right\}. \tag{8}$$

3.2. Implementation Details. Since we use an image representation model, as E_I and D_I, to fulfill our watermark imbedding task, the images should belong to one category rather than any kinds of natural images. So, we study embedding 1-bit QR-code watermark into a specified kind of images, that is, face images. For the image training data, we use 200, 000 24-bit images with size of 128 × 128 from CelebA dataset. For the QR-code watermark data, we also randomly generated 200, 000 1-bit binary images. However, it should be noted that the image and QR-code are not in one-to-one correspondence; they are randomly selected and matched.

For the parameters setting, we set $\lambda_1 = 0.001$ and $\lambda_2 = 0.2$ in equations (5) and (8), respectively, in our training process. We use adaptive moment estimation (Adam) solver [38] with a batch size of 64. All networks were trained from scratch with a learning rate of 0.0001. We keep the same learning rate for the first 100 epochs and linearly decay the rate to zero over the next 100 epochs.

4. Experimental Results and Discussions

To show the effectiveness of our method, we give some comparison results on face image watermarking in this section. We randomly selected 10 face images from CelebA dataset, which were not in the training set. Five of these testing images are shown in the top row in Figure 3. Our watermark images are randomly generated binary QR-code images with size of 32 × 32, which were also not used in the training process. The network was trained using a GPU, NVIDIA GTX 1080Ti, under the PyTorch 0.4.1 environment for two days followed by the instructions as in Section 3.2. The performance of our proposed watermark algorithm is measured from two aspects: visual imperceptibility and robustness against noise and geometric affine transform attack. We compared our algorithm to the state-of-the-art block-based watermark algorithm in the quaternion discrete Fourier transform (QDFT) [39].

For a good watermarking algorithm, the embedded watermarking information should not be visible. So, we use peak signal-to-noise ratio (PSNR) and Structural Similarity (SSIM) [40] to measure the invisibility of the watermarked image, which are defined as

$$\text{PSNR}(\mathbf{x},\widehat{\mathbf{x}}) = 10\log_{10}\frac{255^2 \times 3 \times M \times N}{\|\mathbf{x} - \widehat{\mathbf{x}}\|^2}, \tag{9}$$

$$\text{SSIM}(\mathbf{x},\widehat{\mathbf{x}}) = \frac{\left(2\mu_x\mu_{\widehat{\mathbf{x}}}+c_1\right)\left(2\sigma_{x\widehat{\mathbf{x}}}+c_2\right)}{\left(\mu_x^2 + \mu_{\widehat{\mathbf{x}}}^2+c_1\right)\left(\sigma_x^2 + \sigma_{\widehat{\mathbf{x}}}^2+c_2\right)}, \tag{10}$$

respectively. In equations (9) and (10), $\mathbf{x}$, $\widehat{\mathbf{x}}$, $\mu_{\mathbf{x}}$, and $\sigma_{\mathbf{x}}$ denote the original image, the watermarked image, the mean, and the standard deviation of the image, respectively. From equations (9) and (10), we can find that these two indices, that is, PSNR and SSIM, can reflect, respectively, the pixel level and structural difference of two images, which means the higher PSNR and SSIM values, the smaller the difference between two images and the better the visual invisibility of watermarks. Ordinarily, when the PSNR (resp., SSIM) value is greater than 35 dB (resp., 0.95), we cannot distinguish the difference between two images by our naked eyes directly. Figure 3 shows five original images (top row) and their watermarked equivalents (bottom row), from which we can find that, in the watermarked images, the embedded watermarking information is invisible; that is, our proposed algorithm has strong imperceptibility. For quantitative comparison, the PSNR and SSIM values of ten watermarked test images embedded by our proposed model and the QDFT algorithm are shown in Table 1. Note that the PSNR and SSIM values of these ten test images in Table 1 are derived by averaging watermarked images with embedding five different watermarks as shown in Figure 4. Besides the index comparison about the visual imperceptibility, the computational efficiency is another important index for practical applications. Since the QDFT is a traditional transform based algorithm, its computational complexity is $\mathrm{O}(n^2)$, where n is the number of pixels of an image. However, our proposed watermark algorithm is a deep

Figure 3: Visual impact comparison before and after watermark embedding. (a) The top row shows the original test images from CelebA dataset and (b) the bottom row shows the watermarked test images.

Table 1: Comparison of PSNR and SSIM of watermarked images derived by different methods.

Image ID		200863	201822	200364	200528	200292	200273	201739	201160	201100	200290
QDFT	PSNR	32.09	33.52	37.10	31.59	33.95	32.09	36.01	33.73	32.74	34.33
	SSIM	0.948	0.932	0.978	0.956	0.955	0.925	0.959	0.935	0.938	0.965
Proposed	PSNR	32.91	33.99	37.91	32.54	34.44	32.97	36.85	34.14	33.23	34.79
	SSIM	0.952	0.942	0.979	0.976	0.964	0.936	0.974	0.951	0.959	0.979

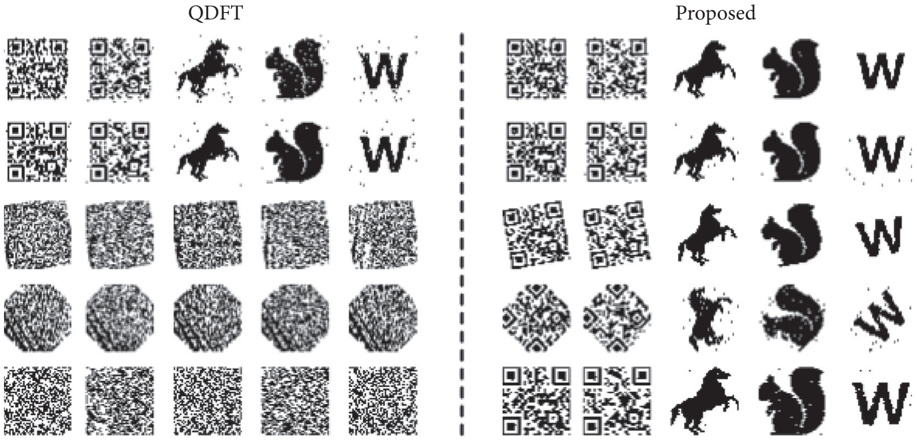


Figure 4: Comparison of the robustness of QDFT and our proposed algorithm against the noise and geometric attacks. From top to bottom are attacks including Gaussian noise, salt and pepper noise, rotation 10°, rotation 45°, and zooming 20%. In each side of the image, from left to right are images with ID 200273, 200290, 201100, 201160, and 201739.

neural network based model whose computational complexity is hard to calculate explicitly. But, for numerical evaluation, our model can process images with size of 128 × 128 pixels at 40 FPS (frames per second) under our experimental environment (NVIDIA GTX 1080Ti, PyTorch 0.4.1). Although the computational time of the proposed model will increase linearly according to the size of input images, with the help of CUDA and AI chip technology, the proposed model is still possible for practical applications.

To show the robustness of our proposed watermark approach, we also conduct the noise and geometric attacks experiment. For noise attacks, we add two kinds of noise, that is, Gaussian noise and pepper noise, to the watermarked images. For geometric attacks, we apply two types of affine transformations, that is, rotation and resize, to the watermarked images. After these attacks, we use the detection network D_W to extract the 1-bit watermark image from the attacked images. Figure 5 shows an example of noise and

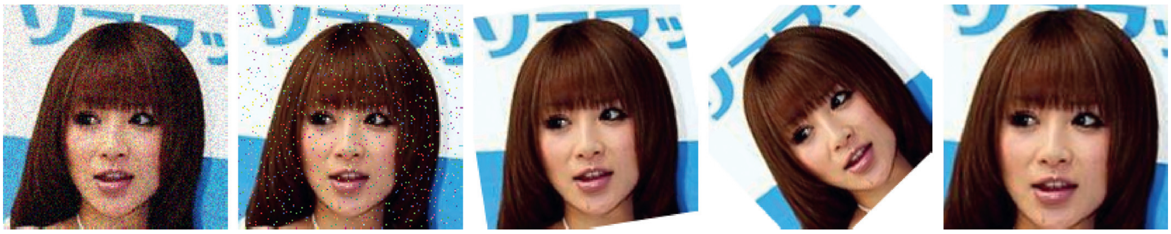


Figure 5: Demonstration of noise and geometric attack used in the experiments. From left to right are Gaussian noise 0.05, salt and pepper noise 0.02, rotation 10°, rotation 45°, and zooming 20% attacks.

geometric attack to the test image. The parameters of Gaussian noise and pepper noise are set to be 0.05 and 0.02 under the scale with image pixel values normalized between 0 and 1. The extracted watermark images derived by QDFT and our proposed algorithm are shown in Figure 4.

From Figure 4, we can find that both QDFT and our proposed algorithm are quite robust against the Gaussian noise and salt & pepper noise attacks. But it seems that our proposed algorithm can extract more clean watermarks compared to the QDFT algorithm. This might be because the autoencoder network itself has a certain ability of image denoising. Then, for the geometric attacks, including rotation and zooming, our proposed algorithm can still extract the correct watermark images, while QDFT algorithm cannot derive satisfactory results. This is because we use the entire image as the input to train our autoencoder and embedding network, but QDFT algorithm is a block-based watermark algorithm. Thus, our proposed algorithm is also quite robust against the geometric attacks.

As shown in Table 1, we can find that, compared to the QDFT method, the proposed approach can achieve an average 0.5 ~ 1.0 dB improvement in PSNR in the aspect of visual quality of watermarked images. Meanwhile, as shown in Figure 4, the proposed approach is more robust than the QDFT method in the aspect of geometric attacks for watermarked images. Therefore, the proposed approach is more robust to attacks and better for watermark information hiding, which means that it has great potential value for practical applications.

5. Conclusion

We propose a new framework for robust image watermarking embedding using cycle variational autoencoder networks. Since the VAE model can learn an abstract representation of a specific kind of images, we use face images to validate our proposed algorithm in this paper. In addition, we train a convolution network to embed a 1-bit watermarking image into the face image in the representation space. Unlike block-based algorithm, that is, QDFT, and DCT-based techniques, our algorithm processes the input image entirely. Therefore, as validated in the experimental section, the proposed algorithm can preserve a better visual quality and is more robust against the noise and geometric attacks compared to those block-based algorithms. However, since we process the input image as a whole, the size of our network will be too big to be practically used for large images directly. So, developing lightweight autoencoder network for large images is an important issue that warrants further study. Moreover, in many real applications, we need to embed watermark information to many different kinds of images, not just face images. That means, compared to the traditional transform-based watermark algorithm, the versatility of our proposed deep learning based model needs to be tested and discussed. To extend our watermark embedding approach to natural images is another issue that merits further study.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61773206).

References

[1] H. Liu, H. Kou, C. Yan, and L. Qi, "Link prediction in paper citation network to construct paper correlated graph," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019.

[2] H. Liu, H. Kou, C. Yan, and L. Qi, "Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph," *Complexity*, vol. 2020, pp. 1–15, Article ID 2085638, 2020.

[3] L. Qi, W. Dou, W. Wang, G. Li, H. Yu, and S. Wan, "Dynamic mobile crowdsourcing selection for electricity load forecasting," *IEEE Access*, vol. 6, pp. 46926–46937, 2018.

[4] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "Energy efficient dynamic offloading in mobile edge computing for internet of things," *IEEE Transactions on Cloud Computing*, p. 1, 2019.

[5] X. Xu, S. Fu, W. Li, F. Dai, H. Gao, and V. Chang, "Multi-objective data placement for workflow management in cloud infrastructure using NSGA-II," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–11, 2020.

[6] X. Xu, H. Cao, Q. Geng, X. Liu, F. Dai, and C. Wang, "Dynamic resource provisioning for workflow scheduling under uncertainty in edge computing environment," *Concurrency and Computation-Practice & Experience*, 2020.

[7] X. Xu, X. Zhang, M. Khan, W. Dou, S. Xue, and S. Yu, "A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems," *Future Generation Computer Systems*, vol. 105, pp. 789–799, 2020.

[8] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2020.

[9] Q. Wei, H. Shao, and G. Zhang, "Flexible, secure, and reliable data sharing service based on collaboration in multicloud environment," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–16, Article ID 5634561, 2018.

[10] X. Chi, C. Yan, H. Wang, W. Rafique, and L. Qi, "Amplified locality-sensitive hashing-based recommender systems with privacy protection," *Concurrency and Computation: Practice and Experience*, 2020.

[11] W. Zhong, X. Yin, X. Zhang et al., "Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment," *Computer Communications*, vol. 157, pp. 116–123, 2020.

[12] C. Zhou, Li Ali, A. Hou, Z. Zhang, Z. Zhang, and F. Wang, "Modeling methodology for early warning of chronic heart failure based on real medical big data," *Expert Systems with Applications*, vol. 151, Article ID 113361, 2020.

[13] X. Xu, Q. Liu, X. Zhang, J. Zhang, L. Qi, and W. Dou, "A blockchain-powered crowdsourcing method with privacy preservation in mobile environment," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1407–1419, 2019.

[14] S. D. Lin and C.-F. Chen, "A robust DCT-based watermarking for copyright protection," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 3, pp. 415–421, 2000.

[15] N. Kashyap and G. Sinha, "Image watermarking using 3-level discrete wavelet transform (DWT)," *International Journal of Modern Education and Computer Science*, vol. 4, pp. 1–7, 2012.

[16] X. Hu, S. Peng, and W. Hwang, "EMD revisited: A new understanding of the envelope and resolving the mode-mixing problem in am-fm signals," *IEEE Transactions on Signal Processing*, vol. 60, no. 3, pp. 1075–1086, 2012.

[17] D. Volkhonskiy, I. Nazarov, B. Borisenko, and E. Burnaev, "Steganographic generative adversarial networks," in *Proceedings of the NIPS 2017 Workshop on Adversarial Training*, pp. 201–208, Long Beach, CA, USA, November 2017.

[18] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, "Secure steganography based on generative adversarial networks," in *Proceedings of the Advances in Multimedia Information Processing - PCM 2017*, pp. 534–544, Harbin, China, September 2017.

[19] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1547–1551, 2017.

[20] S.-M. Mun, S.-H. Nam, H. Jang, D. Kim, and H.-K. Lee, "Finding robust domain from attacks: A learning framework for blind watermarking," *Neurocomputing*, vol. 337, pp. 191–202, 2019.

[21] J. Zhu, T. Park, P. Isola, and A. A. Efros, ""Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, Venice, Italy, October 2017.

[22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 2672–2680, Montreal, Canada, December 2014.

[23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," http://arxiv.org/abs/1312.6114.

[24] A. Martin, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 214–223, Sydney, Australia, August 2017.

[25] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2813–2821, Venice, Italy, October 2017.

[26] L. Mescheder, S. Nowozin, and A. Geiger, "Adversarial variational bayes: unifying variational autoencoders and generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 2391–2400, Sydney, Australia, August 2017.

[27] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 327–340, Los Angeles, CA, USA, August 2001.

[28] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *Computer Science*, pp. 2672–2680, 2014.

[29] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proceedings of the 29th Advances in Neural Information Processing Systems*, pp. 379–390, Barcelona, Spain, December 2016.

[30] M.-Yu Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proceedings of the 30th Advances in Neural Information Processing Systems*, pp. 700–708, Long Beach, CA. USA, December 2017.

[31] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 1–18, Munich, Germany, September 2018.

[32] H. Dou, C. Chen, X. Hu, and S. P. Peng, "Asymmetric cyclegan for unpaired NIR-to-RGB face image translation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1757–1761, Brighton, United Kingdom, May 2019.

[33] H. Dou, C. Chen, X. Hu, L. Jia, and S. Peng, "Asymmetric cyclegan for image-to-image translations with uneven complexities," *Neurocomputing*, vol. 415, pp. 114–122, 2020.

[34] A. H. Jha, S. Anand, M. Singh, and V. S. R. Veeravasarapu, "Disentangling factors of variation with cycle-consistent variational auto-encoders," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 829–845, Munich, Germany, September 2018.

[35] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–16, San Juan, Puerto Rico, May 2016.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[37] Sergey Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456, Lille, France, July 2015.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference*

on Learning Representations (ICLR), San Diego, CA, USA, May 2015.

[39] X.-Y. Wang, C.-P. Wang, H.-Y. Yang, and P.-P. Niu, "A robust blind color image watermarking in quaternion fourier transform domain," *Journal of Systems and Software*, vol. 86, no. 2, pp. 255–277, 2013.

[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8872586, 13 pages
https://doi.org/10.1155/2020/8872586

Review Article

Artificial Intelligence for Securing IoT Services in Edge Computing: A Survey

Zhanyang Xu, Wentao Liu, Jingwang Huang, Chenyi Yang, Jiawei Lu, and Haozhe Tan

School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China

Correspondence should be addressed to Wentao Liu; liuwentao@nuist.edu.cn

Received 30 July 2020; Revised 17 August 2020; Accepted 29 August 2020; Published 14 September 2020

Academic Editor: Dou Wanchun

With the explosive growth of data generated by the Internet of Things (IoT) devices, the traditional cloud computing model by transferring all data to the cloud for processing has gradually failed to meet the real-time requirement of IoT services due to high network latency. Edge computing (EC) as a new computing paradigm shifts the data processing from the cloud to the edge nodes (ENs), greatly improving the Quality of Service (QoS) for those IoT applications with low-latency requirements. However, compared to other endpoint devices such as smartphones or computers, distributed ENs are more vulnerable to attacks for restricted computing resources and storage. In the context that security and privacy preservation have become urgent issues for EC, great progress in artificial intelligence (AI) opens many possible windows to address the security challenges. The powerful learning ability of AI enables the system to identify malicious attacks more accurately and efficiently. Meanwhile, to a certain extent, transferring model parameters instead of raw data avoids privacy leakage. In this paper, a comprehensive survey of the contribution of AI to the IoT security in EC is presented. First, the research status and some basic definitions are introduced. Next, the IoT service framework with EC is discussed. The survey of privacy preservation and blockchain for edge-enabled IoT services with AI is then presented. In the end, the open issues and challenges on the application of AI in IoT services based on EC are discussed.

1. Introduction

With the widespread deployment of sensors in the real world, increasing physical entities are connected to the Internet of Things (IoT) through sensors to achieve information sharing. Currently, IoT technology has been widely applied in various fields such as smart city, smart home, wearable medical, and environmental perception, [1–3]. In conventional IoT services, those sensors and devices interconnected with IoT need to upload the data to the cloud servers to handle computing tasks. After the tasks are completed, the processed data will be returned to the IoT devices. Although the cloud reduces the computing burden of sensors and devices, huge transmission overhead of the data cannot be ignored. In 2018, the total amount of devices connected to IoT around the world reached 11.2 billion, and it is predicted to grow to 20 billion in 2020 [4], which brings rapid data growth. However, the current growth of network bandwidth is far behind the speed of data growth, and the complex network environment greatly hinders the reduction of latency. Network bandwidth has become the major bottleneck that should be solved for the traditional IoT services.

To solve the abovementioned bottleneck, a new computing paradigm called edge computing (EC) has been proposed recently and gets widespread attention. EC refers to the technology that deploys computing tasks to the edge of the network [5, 6]. Compared with cloud computing, EC has many advantages, including protecting end-users' privacy, reducing the latency while data transmission, decreasing the burden of network bandwidth, and lessening the energy consumption of data center. Under EC, the raw data generated by IoT devices are no longer required to be uploaded to the centralized cloud platform but can be computing, stored, and transmitted at edge nodes (ENs), reducing the latency time owing to voiding redundant data transmission. Those IoT applications and mobile computing that have strict requirements on response time will be better supported by EC.

However, EC is not a panacea. On the one hand, the potential of IoT devices under the EC has been greatly expanded in many fields (computation offloading, precise positioning, real-time processing, etc.), giving the credit for low-latency data processing near end-users. On the other hand, EC introduces more security issues and widens the attack surfaces [7] of the system from 3 aspects:

(1) Distributed layout: the ENs are distributed at various locations on the edge of the network [8], making it difficult to unify all equipment for centralized management. The adversary can attack those ENs that have security flaws and use the nodes hijacked as a springboard to make an incursion to the entire system.

(2) Limited computing source: unlike cloud computing, the computational functionality of ENs is limited for the reason of the physical structure, which means that heavyweight security mechanisms are not suitable for ENs and large-scale centralized attacks such as the distributed denial of service (DDoS) attack will cause great damage to ENs.

(3) Heterogeneous environment: a wide range of technologies are applied in EC, including wireless sensor networks, mobile data collection, grid computing, and mobile data collection. Under this heterogeneous environment, it is difficult to design a unified security mechanism and achieve consistency of security policies between different security domains.

In order to make up for the safety hazards caused by the characteristics of edge computing, many security methods and algorithms come forth [9, 10]. Most of the current security mechanisms are based on the algorithms and models that follow a single pattern for intrusion detection, privacy preservation, or access control. With the continuous upgrade of attack techniques and methods, traditional defense mechanisms are often quickly eliminated. However, what is exciting is that the emergence and rise of artificial intelligence (AI) provide new solutions to security and privacy issues:

(1) Intrusion detection: common intrusive attacks are denial of service (DoS) attack and distributed denial of service (DDoS) attack. DoS makes frequent requests to the server, which increases the burden on the server and affects the server's response to normal requests, and DDoS refers to controlling the multiple compromised ENs to attack the server. The intrusion detection system (IDS) identifies attacks from the hijacked ENs by monitoring anomalous traffic on the network and cut off access from them. Machine learning (ML) extracts malicious access patterns through the training of the previous data sets, which can help IDS to quickly and accurately identify intrusions, greatly improving the detection efficiency compared with traditional recognition methods [11, 12].

(2) Privacy preservation: the IoT devices exist in every aspect of our lives, which contains much privacy-sensitive information as well [13]. Most existing privacy preservation methods encrypt the transmitted data to ensure data security, such as anonymization, cryptographic methods, and data obfuscation. Nonetheless, the above methods generally require high computational overhead, making it difficult to deploy on resource-constrained ENs. Compared to common encryption methods, distributed machine learning (DML) makes the ENs only need to pass the parameters to other ENs for cooperative learning after each training, instead of directly passing the original data, reducing the risk of data leakage and network burden during transmission [14].

(3) Access control: when multiple IoT devices work together in the same environment, access control becomes a key issue. Each authenticated node can only access the nodes and data within their authorities and cannot perform other operations beyond their access authorities [15, 16]. ENs need to be classified into different categories according to permissions, which coincides with the classification algorithm under ML [17]. The algorithm classifies the ENs connecting to the network to low-privileged IoT devices and high-privileged IoT devices. Access to those high-privileged devices will be strictly controlled to prevent potential attacks.

As the investigations of AI continue to advance, AI has gradually been applied to many fields of edge security [18, 19]. However, there are still many challenges in the realization of related theories on ENs. For instance, large amounts of clear data are important to the training efficiency of ML, but the premise of sufficient data is that the system has received mass attacks and can accurately identify these malicious behaviors [20]. Meanwhile, the attacks against the training set also need to be vigilant, which will reduce the performance of the model by tampering the parameters [21]. The lightweight AI algorithm is also needed because of the restricted computing resource and storage at ENs, but it will bring a drop in accuracy.

Although lots of investigations on the combination of AI and EC have been carried out, there is still little discussion and inquiry of AI in the security of IoT based on EC. Therefore, a comprehensive review which focuses on state-of-the-art technology and achievements about the above-mentioned field is presented.

The remaining parts of this paper are organized as follows. Section 2 introduces the basic definitions of IoT and EC. In Section 3, the IoT service framework with EC is discussed, followed by the survey of privacy preservation in EC enabled IoT with AI in Section 4. Section 5 presents the AI for blockchain in EC enabled IoT. Finally, Section 6 talks about the open issues and challenges of the application of AI in IoT security based on EC.

2. Basic Concepts and Definitions

2.1. IoT Service. Literally, IoT is to construct a global network of things where everything is connected to the Internet, thus realizing the interconnection of all objects using Internet technologies. With IoT technology, devices are able to transmit information to each other and several devices work together to complete a task without the intervention of humans. IoT can be applied in various industries by embedding sensors into objects such as medical equipment, home equipment, transports, implementing the integration of human society, and physical world.

IoT architecture is comprised of perceptual layer, network layer, and application layer [22], and each layer has its own specific function. The perceptual layer is employed to perceive the environment and obtain data by virtue of sensor technology, RFID, wireless communication technology, etc., acting as the indispensable foundation of IoT. The network layer is responsible for data transmission from the perceptual layer to the application layer. Besides, cloud platform serves as a vital component of this layer to store and analyse substantial perceived data. The application layer is the top layer of IoT architecture. This layer provides specific services for users based on processed and analysed data. Through the three layers, IoT devices can understand users' needs and accordingly give them the services they want, improving their living quality.

Next, we will illustrate three typical IoT services and their respective specific application scenarios which are introduced in Table 1 as follows:

(1) Remote monitoring and control: IoT allows users to control the devices connected to the Internet and monitor a scenario remotely, which brings generous convenience to our life. Users are enabled to monitor the condition of their babies anywhere with the help of the sensors installed at home that collect data on the baby's health status at any time. Furthermore, cameras can transfer baby's video to users timely. When it comes to logistics, customers can easily know about the condition of products in transit. Information about the quality statue and current location of goods they purchased online can be queried regardless of time.

(2) Smart home: smart home [27] has developed several years so that it is not a novel concept for us. However, what deserves our attention is that with the usage of IoT, smart home products contain a huge potential to become more intelligent and versatile, able to serve users better. Suppose that as soon as you enter your room from the outside, the air conditioner is turned on and adjusted to a comfortable temperature automatically for you. Many other products such as sweeping robots will free you from housework, and even lights can be switched on/off by themselves without any manual operation. Thus, it can be seen that smart home is one of the most direct manifestations of how IoT services make our lives easier and more comfortable.

(3) Natural disaster prediction: IoT plays an important role in the prediction of disasters such as earthquakes, floods, drought, and tsunami [28]. Sensors deployed outside are appointed to gather data from the ambient environment, and the processed data may reveal crucial information about the coming natural calamity, thus saving up enough time for us to remove people away from the disaster area and avoid property loss as much as possible.

So far, on the topic of benefits IoT brings about, we have only referred to the tip of the iceberg. Undeniably, IoT has served as a powerful engine driving revolutions in many traditional offline industries. Though IoT is still in its initial state, it has a wide application range which is just limited by humans' imagination and it is bound to influence almost every aspect of our life in the near future.

2.2. Edge Computing. EC is a new computing mode that processes and stores data at the edge of the network in close proximity to mobile devices and users [29]. In Table 2, we describe the definition of EC from two different angles.

With the advent of the IoT era, the scale of mobile devices is expanding incredibly and the high volume of data is produced by terminal devices every day [35–37]. It is unwise to transmit all data to the cloud center considering the excessive burden of bandwidth and massive energy consumption in the cloud. Besides, traditional cloud computing cannot process such a huge amount of data efficiently, which extends latency time and reduces response speed [38, 39]. At the same time, certain emerging technologies such as AR and VR [29] have higher requirements for low latency and fast response time. The contradiction between our growing need for higher computing efficiency as well as better privacy security and the limitations of cloud computing calls for a decentralized computing mode that can complement the cloud computing and push the future development of the IoT industry. Naturally, EC's advantages begin to be valued by humans under this circumstance.

Three outstanding advantages of EC are introduced as follows:

(1) Low latency: instead of transmitting all data to cloud center, data computations are completed at the edge of the network closer to mobile devices, thus increasing the response speed and declining the latency [40].

(2) Privacy and security: thanks to EC, data are allowed to be stored locally or in ENs and privacy information does not have to be transmitted to cloud center so that the threat of privacy leakage has been effectively reduced [40].

(3) Decrease energy consumption in cloud center: in EC, part of computing tasks is offloaded to several ENs, which not only relieves the burden of bandwidth [29] but also helps reduce the energy consumption in the cloud center.

Table 1: Three typical IoT services and their specific application scenarios.

IoT services	Application scenarios
Application scenarios	(1) A greenhouse system based on IoT can monitor and control environmental parameters to facilitate plant growth and production [23]. (2) Cold chain logistics [24] can depend on IoT to maintain suitable storage and transportation temperature, ensuring the quality of goods.
Smart home	(1) Users can easily control devices inside the home via smart home systems to avoid unnecessary energy waste [25]. (2) With the usage of smart home and various devices connected to the Internet, users can enjoy the convenience of controlling the house at any time [25].
Disaster prevention	Disaster management systems based on IoT can be deployed in buildings of seismic areas to monitor the conditions of buildings of seismic areas, providing earthquake early warning [26].

Table 2: Two angles to define edge computing.

Definition of edge computing	Advantages	Related work
A distributed computing mode that offloads computation tasks to different edge nodes	Better privacy protection	A dispersed edge cloud infrastructure called Nebula [30] is presented.
	Relieve bandwidth burden	Cloudlet, an edge computing platform is introduced in [24, 31].
	Save energy in the cloud	A lightweight differential privacy-preserving mechanism used in edge computing is proposed in [32].
A new paradigm where computational resources are placed closer to data sources	Low latency	LAVEA [33] is a system designed to provide low-latency video analytics at places in close proximity to users.
	Fast response speed	The impact of both latencies in MEC architecture with regard to latency-sensitive services is researched in [34].

Today's IoT services are mostly cloud-based and centralized so that all data processing and analyses have to be completed in cloud [41]. With the prosper of IoT, more IoT devices demanding low latency and high response spring up [42]. However, cloud computing has encountered its bottleneck, unable to provide support for the sharp development of IoT continuously. Only making best of advantages of EC can IoT services be blessed with a bright outlook.

3. IoT Service Framework with EC

IoT service framework with EC can be divided into four major layers: device layer, network layer, edge layer, and cloud layer. Figure 1 shows the basic diagram of the IoT service framework with EC.

3.1. Device Layer. Various objects or electronic devices such as mobile phones, computers, cars, and even humans (in IoMT [43]) are equipped with different kinds of sensing devices such as RFID, intelligent sensors, and QR code. With them, 'things' in the IoT have the ability to provide context-based information about themselves or their surroundings in real-time, thus generating a large amount of real-time data. These data vary greatly due to different processing requirements, but most of them are fast, instantaneous, and frequent.

3.2. Network Layer. This layer can be seen as a channel among cloud, edge, and end. On the one hand, the layer is the transition between the device layer at the bottom and the edge layer at the upper end. It is the nervous system of the IoT service framework, connecting the sensing devices all over the IoT and undertaking the task of transmission. The data obtained from sensing devices are transmitted through different communication technologies [44] such as cellular networks composed of base stations, WiFi, ZigBee, Bluetooth, etc., which follow various IoT protocols or data transmission protocols, such as Hypertext Transfer Protocol and Message Queuing Telemetry Transport.

In order to adapt to the new computing model of EC and meet the requirements of establishing computing path and dynamically realizing computing services and data migration, named data networking (NDN [45]), a data network that names and addresses data and services, is applied to the context of edge computing. Besides, software-defined networking (SDN [46]), a programmable network that separates the control plane from the data plane and can perform simple network management, is paid attention to. As a result, through the combination of the two, data migration and transmission can be well-realized, and service organizations can be carried out quickly, so as to meet the requests of service discovery and rapid configuration in the network layer under the background of EC. On the other hand, the layer also links up the edge layer and cloud layer composed of cloud-data centers. It takes on the task of transmitting the data organized or concluded by the edge layer and transferring orders or feedback from the cloud layer to the edge layer.

Moreover, the safety of the network layer cannot be neglected. Physical isolation design and logical security design are two main approaches to securing the layer. Specifically, Air Gap that makes use of physical isolation technology and high-strength protocol analysis function to isolate the inside and outside network, routing attack protection design, and denial of service protection design is usually used.

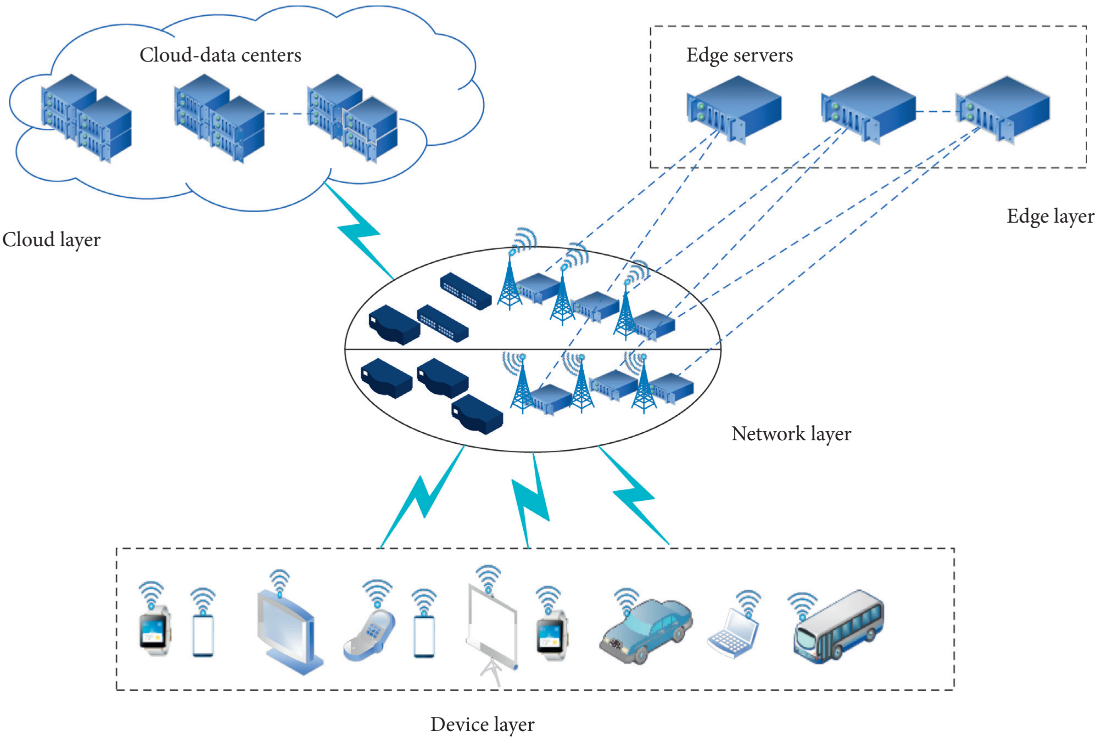


Figure 1: IoT service framework with edge computing.

3.3. Edge Layer. Compared to the conventional IoT service framework, this layer is the main characteristic of the IoT service framework with EC, which solves the problems of insufficient bandwidth and high delivery delay, to a certain extent. To processing a great number of data from IoT devices efficiently and accurately, partial computing resources are shifted from cloud to edge which is much closer to data sources.

As core compositions in edge layers, edge servers are principal undertakers of data processing, data management, and data storage. The results are transmitted to corresponding devices or uploaded to the cloud layer for further analysis or storage through the network layer. The deployment of edge servers which needs to satisfy the requirements of users under resource constraints has a significant impact on computing efficiency and computing resources utilization. Edge servers are usually deployed in cellular base stations' vicinity. Besides, they are often deployed in a single entity rather than multitenant [47]. In 2018, Zhao et al. proposed an innovative three-phase deployment way [48] that takes traffic diversity and wireless diversity of IoT into consideration in large-scale IoT, which greatly promotes the reduction of ENs.

To ensure the smooth and efficient operation of computing tasks, some core technologies such as edge operating systems, isolation techniques, and data processing platforms boost the development of the edge layer.

3.4. Cloud Layer. The layer is the brain of the IoT service framework with EC. It is usually composed of large cloud-data centers with extraordinary computing power. In the IoT service framework with EC, the cloud layer tends to be applied to further processing data from the edge layer, storing or updating significant information and carrying out advanced deployment.

Nonetheless, in some special situations, the importance of cloud-edge collaboration is highlighted. Cloud-edge collaboration includes resource collaboration, management collaboration, safety collaboration, and so forth, which think of cloud and edge as all in one to reinforce each other and schedule dynamically. Specifically, when computing resources in the edge layer are insufficient, the cloud layer can offer computing support with virtual machines and containers. When a certain edge layer appears malicious traffic, the relevant cloud layer which is equipped with better security policy has the ability to discover and block it so as to prevent it from continuing spreading. The establishment of cloud-edge collaboration has aroused wide concern. A few cloud-edge collaboration platforms such as KubeEdge, Edge Tunnel, and AWS Wavelength are pushing ahead with the prosperity of cloud-edge collaboration.

The application of IoT service with EC is booming and hot. Table 3 shows some typical examples of IoT service with EC.

4. Privacy Preservation for Edge-Enabled IoT Services with AI

The privacy protection methods in ML can be generally divided into two kinds, namely, training schemes and inference schemes in [54]. The privacy-preserving training schemes target to use encryption methods to ensure the security of sensitive privacy information during the

Table 3: Typical examples of IoT service with edge computing.

Reference	Application field	Author	Specific design
[49]	Smart cities	Sapienza et al.	Make use of mobile edge computing to monitor critical events (e.g., terrorist attack or disasters)
[50]	Smart farms	Caria et al.	Propose a smart farm animal welfare monitoring system based on edge computing (e.g., collecting and processing data from animals and surroundings)
[51]	Connected and autonomous vehicles	Liu et al.	Create an edge-based attack detection (e.g., detecting speech, video data, and driving behavior)
[52]	Smart home	Cao et al.	Implement a home operating system named EdgeOSH which includes various modules (e.g., data management, communication, and self-management)
[53]	Public safety	Zhang et al.	Present an AMBER alert assistant (A3) based on extended firework (e.g., following an illegal vehicle)

transmission. The privacy-preserving inference schemes focus on protecting the privacy data in the inference phase. Usually, in preserving inference schemes, a well-trained model receives the unclassified data sent by the EN for inference [54]. The common encryption methods include anonymization, cryptographic method, data obfuscation, and so on. However, the above methods for encryption always require different levels of computing overheads and communication overheads. It hinders the implementation of encryption methods on resource-limited ENs. As Figure 2 shows, the following parts of this section will talk about existing basic encryption approaches firstly and then furtherly discuss proper privacy-preserving methods for edge-enabled IoT service.

4.1. Existing Basic Encryption Methods

4.1.1. Anonymization. Anonymization techniques are applied to anonymize participants' identities in a group of people, by removing some obvious characteristics such as a user's name, sex, and ID number. Since the loss information is related to the user's specific identity, the most valuable data we need will not be ruined during the transmission. Many privacy preservation models using anonymization technology have been proposed, such as k-anonymity, l-diversity [55], and t-closeness [56] models. In the k-anonymity model, the participants' attributes are generally classified into three categories: explicit identifiers, the quasi-identifier attribute set, and sensitive attributes. Before the data are released, the explicit identifiers will be removed and the data in the quasi-identifier attribute set will be generalized to ensure that there are at least k records with the same quasi-identifier. However, the k-anonymity technique is flawed. The attacker can reidentify victims by linking or matching the data to other background data or by looking at unique attributes found in the released data [57]. Later, some researchers proposed I-diversity and t-closeness models based on the k-anonymity technique to defend against the above attacks. The I-diversity model requires that the diversity of sensitive attributes should not be less than I in each quasi-identifier class, thus reducing the matching probability between sensitive attributes and their owners. The t-closeness module requires that the distance between the distribution of sensitive attributes in each equivalence class and the general distribution of sensitive attributes do not exceed the upper limit t.

4.1.2. Cryptographic Method. Cryptographic methods encrypt the context of the data before uploading them to the cloud servers. However, cryptographic methods incur high compute overhead (millions of times higher than multiplicative projection) and require reliable and effective key management [58]. Homomorphic encryption (HE) can entrust third parties, such as various applications of cloud computing, to process the data without revealing the information. HE technology is secure in that they generate a key pair based on some mathematical problems which are difficult to be solved by the computer. The key pair includes a public key and a private key. The public key and some operation measures will be published to third parties. Then, the third parties carry out all the operations on the encrypted data and send back the results, which can only be decrypted by the private key; thus, the information is confidential throughout the whole process. The common homomorphic encryption algorithms include the RSA algorithm and the ECC algorithm. The later one has a lower computing overhead.

4.1.3. Data Obfuscation. Obfuscation methods perturb the data samples used for training a global module. The methods include additive perturbation and multiplicative perturbation. Additive perturbation is always related to differential privacy (DP), which is used to aggregate information without revealing any special entry [59]. Under the mechanism of DP, the adversary cannot tell the difference between the output of neighboring datasets, thus protecting the safety of different records of neighboring datasets. DP obfuscates the data by adding noises through some mechanisms such as Laplacian [60], exponential [61], and median mechanisms [62]. Laplacian mechanism realizes the DP protection by adding random noise with Laplacian distribution to the exact query outputs. Different from the Laplacian mechanism, the exponential mechanism selects the optimal output according to the probability after each query. The randomize multiplicative data perturbation technique is a type of multiplicative perturbation. The random projection scheme tries to create a new data representation with fewer dimensions through randomize multiplicative matrices [63]. Generally, data obfuscation has been widely applied in data mining to protect the users' privacy while obtaining high-quality data.

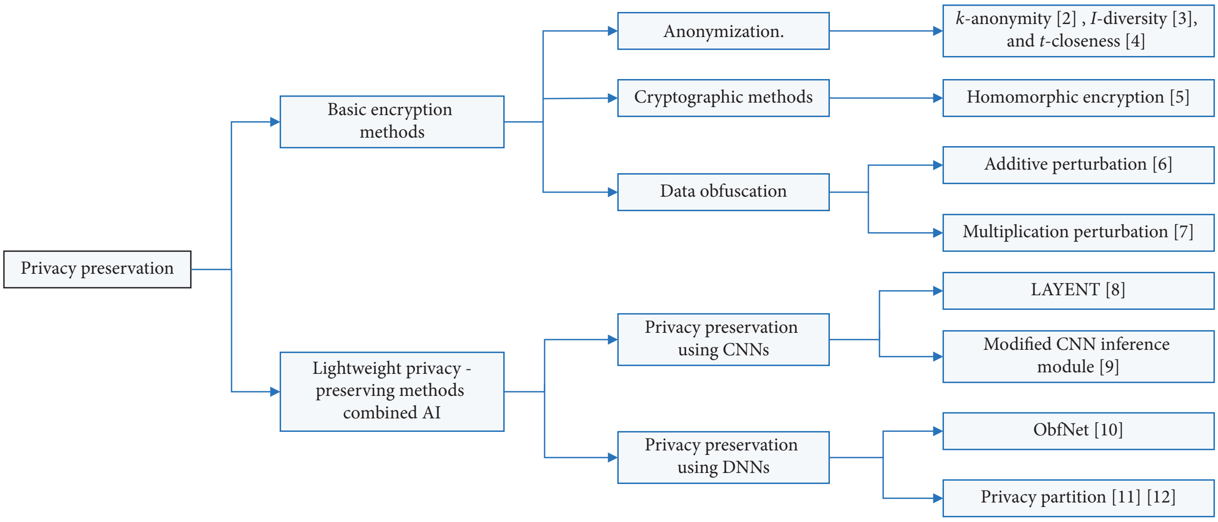

Privacy preservation
Basic encryption methods
Lightweight privacy - preserving methods combined AI
Anonymization.
Cryptographic methods
Data obfuscation
Privacy preservation using CNNs
Privacy preservation using DNNs
k-anonymity [2] , l-diversity [3], and t-closeness [4]
Homomorphic encryption [5]
Additive perturbation [6]
Multiplication perturbation [7]
LAYENT [8]
Modified CNN inference module [9]
ObfNet [10]
Privacy partition [11] [12]

Figure 2: Structure of this section.

4.2. Lightweight AI Privacy-Preserving Methods in ENs. The booming development of IoT encourages a new computing paradigm "edge computing," which leverages the computing and storage capability of device nodes between the cloud center and terminal devices. Compared with traditional cloud computing, EC pushes the process of data close to the data sources, reducing the data required to be sent to the data center originally. It provides real-time services with low latency and reduces communication bandwidth usage. The attention has concentrated on the privacy problems not only in the training of the module but also in tasks offloading schemes [64]. Conventional encryption methods such as anonymization, cryptographic method, and data obfuscation have been requested in computing power and they are originally implemented in the cloud center. Thus, it is hard for conventional encryption methods to work effectively on the resource-constrained ENs. Recently, some research studies have focused on creating lightweight privacy preservation methods combined with AI technologies such as convolutional neural network (CNN) and deep neural network (DNN) models and other ML technologies to optimize the traditional encryption methods.

4.2.1. Privacy Preservation Using CNNs. The following context will introduce two schemes: LAYENT and the modified CNN inference module. The former scheme optimizes the basic framework to make the module privacy-aware and the later uses the trained module for privacy preservation.

(1) LAYNET. LAYENT is a new privacy-preserving algorithm in machine learning. Compared with most related algorithms based on cloud computing's processing power, LAYRNT not only has a very high accuracy up to 91% but also well protects the privacy incurring a low computing overhead [61]. Before the data are transmitted to the potential unsafe third party, the data will be perturbed. To achieve this function, the algorithm LAYENT improves the original CNN framework, by adding a new layer—the randomization layer between convolution layers and full connected layers. Moreover, the randomization layer employs a new unary encoding protocol to enhance the flexibility of randomization when encoding the context.

(2) Modified CNN Inference Module. An energy theft detection scheme is proposed to detect the unusual behavior of the smart meters in the smart grid. The scheme combines the modified convolutional neural network (CNN) module in the framework. The data generated by the smart meter are used to train the CNN module, and then the trained module can detect the abnormal data by making reasonable inferences after training. The scheme combined the modified CNN module has excellent behavior in the experiment in that the accuracy of the inference has reached up to 92.67% [62].

4.2.2. Privacy Preservation Using DNNs. Deep neural network is a framework in deep learning, and it has been widely applied in many areas such as the understanding of natural language, speech recognition, and image recognition. The training of DNN modules needs to consume large computing power. The following context is two lightweight schemes:

(1) ObfNet. An obfuscation neural network (ObfNet) approach is proposed to obfuscate the inference data before being transmitted to the backend [63]. ObfNet is an approach that realizes lightweight and unobtrusive data obfuscation for remote inference. The lightweight and unobtrusive characters refer that the ENs only need to implement a small neural network and do not need to indicate whether the data are obfuscated.

There are two issues in the implementation of the edge-enable IoT. One of them is the separation of information sources and computer power, and the other is the privacy preservation of inference models. Remote inference can overcome the above issues. In remote inference, the collected data will be sent to the backend and then the inference results will be returned.

ObfNet is a light neural network suitable to be deployed in ENs. The training process is designed as follows. The backend connects the untrained ObfNet with the trained in-service inference model (named InfNet) in the center, forming a concentrated DNN module. In the DNN module, the output of ObfNet is the input of InfNet. The backend uses the part random data which are used for the training of InfNet to train ObfNet. Meanwhile, only the ObfNet's weights are sent to the backend until convergence. Repeating the procedure, the backend can generate a group of ObfNets. Due to the random data sources and random original weights of ObfNets, all the ObfNets are different from each other. Finally, EN chooses an ObfNet randomly and dynamically.

(2) Privacy Partition. A practical method named privacy partition for privacy preservation in ML is presented in [65, 66]. Privacy partition is a privacy-preservation framework for deep neural networks, and the basic structure of the framework is made up of a bipartite topology network and an interactive adversarial network [65].

A bipartite deep network topology is made up of two partitions: a trusted local computing context and the untrusted remote computing context, forming a neural network. The output of the last transformation in a trusted local computing context will be processed by a learning module. After that, the processed information will be the input of the first transformation layer in remote computing context [65]. Under the architecture of the edge network, privacy partition provides an optional choice to some centralized deep learning frameworks. Users can limit access to the sensitive data stream for privacy preservation.

The interactive adversarial network provides a practical solution when the ENs need to use remote services and computing. It can attenuate the capacity of the adversary who has access to deep network intermediate state to learn privacy-sensitive input.

5. Blockchain for Edge-Enabled IoT Services with AI

Blockchain is a distributed computing and storage paradigm with a variety of existing technologies. The distributed consensus algorithm is used to generate and update data, transmits data between nodes by a peer-to-peer network and keeps the stored data immutable by a distributed ledger. It also uses an automated script code or smart contracts to implement upper-layer application logic [67]. In short, blockchain provides a new approach to preserve and transmit data safely against attack or bug and gives a decentered environment.

Part 1 includes the method of most urgent security problems in IoT services by blockchain. Part 2 discusses the sharing of data resources which is from one mechanical device to another mechanical device and provides many communication facilities. Part 3 includes the improvement of efficiency in the environment based on the IoT networks. Besides, the hierarchical taxonomy of the section is shown in Figure 3 and the research studies we discuss are listed in Table 4.

5.1. Blockchain for IoT Services' Security.

In order to build an IoT network that can be in use, massive terminal devices will be set and any device in IoT network can get the data from the whole IoT network. Due to the number of devices, the weakness of a single device cannot be avoided. If the device is hacked into, massive data in IoT services will be leaked out which may result in disastrous consequences [80]. Therefore, improving the security of IoT services becomes unavoidable.

The connection of these IoT devices is not safe due to the quantity of the devices. As a result, it is easy for ill-disposed people to steal the data which are transmitted between devices. Although there are some ways to solve the unsafety such as CAPTCHAs, it still has the limitation in the protection of data. So, blockchain technology and AI technology are introduced to solve them. There are two aspects which will be introduced below: (1) access control and authentication management and (2) confidentiality and reliability of data.

5.1.1. Access Control and Authentication Management.

Access control is to provide a set of methods to identify, organize, and host all functions in the system, organizing and identifying all data, and then provide a simple and unique interface [81]. Authentication is to identify the access by verification tools such as passwords and decide whether to give the interface of the system.

In the traditional IoT service which is without AI or blockchain, the way of identity authentication is to authenticate the combination of the user name and password for each device. This way will cost a lot of energy and have difficulty in extension, so it can be used in IP cameras. Single sign-on protocols can simplify identity authentication, which is to provide a reliable third-party organization to give the user access to multiple devices by authenticate identity for a single device. Although it can accelerate the authentication, it will result in a horrible consequence to the whole IoT system if the account of users is destroyed or one device is broken down.

To solve these problems, a new design has been proposed in the article [68]. In this design, the users only need to authenticate identity to the blockchain (such as Ethereum) once then use the smart contract token to access the system. Smart contact will broadcast the token and the Ethereum address when authenticating identity, and the IoT service will receive the package which includes the user's public key, IP, and token to authenticate the package. Besides, fingerprint information collection, storage, and verification can be completed by blockchain to solve the falsify problem in the access authentication technology at present [69].

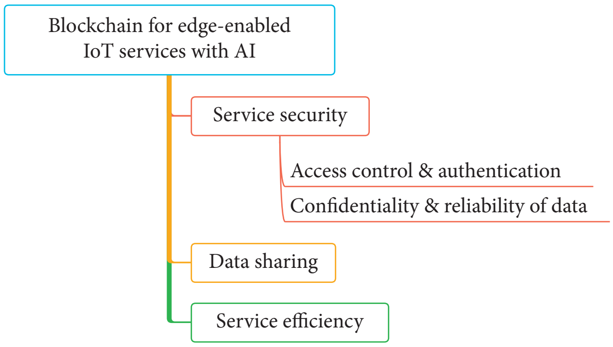

Blockchain for edge-enabled IoT services with AI
Service security
Access control & authentication
Confidentiality & reliability of data
Data sharing
Service efficiency

Figure 3: Hierarchical taxonomy of blockchain for edge-enabled IoT services with AI.

5.1.2. Confidentiality and Reliability of Data. The application of the IoT continues to grow in various fields such as healthcare, finance, and agriculture. In the field of health care, with the application of IoT, different kinds of physical sensors are be in use to record the body data, which can help the doctors to improve the methods of medical treatment for patients. These personal data need to be protected safely.

Rui et al. [70] propose a method which realizes distributed storage and tampers resistance of data in data blockchain and improves the utility Byzantine fault-tolerant (PBFT) mechanism consensus algorithm to store IoT data safely. Xu et al. [71] propose a blockchain-powered crowdsourcing method. They design a mobile crowdsourcing architecture based on blockchain to keep players' data private and complete. They generate service policies by density-based spatial clustering of applications with noise and improved dynamic programming. Besides, they judge the polices by using simple additive weighting and multiple criteria decision making.

5.2. Blockchain for Edge-Enabled IoT Data Sharing. Data are the basis of the IoT network, more data can be collected, and the research results and the improvement of the application are more accurate. At present, IoT data are collected by lots of different types of ways in many fields such as agriculture, industry, healthcare, and automatic drive. This shows that the sensors collecting different kinds of data are heterogeneous, and the database is owned by different companies, organizations, or governments. The isolation of data costs a large cost of energy and time because of collecting repetitive data. Therefore, sharing data from the IoT services in the database can assign the resource properly and reduce the avoidable cost.

However, massive data, heterogeneous devices, lack of trust, security problem, and some other problems become barriers to safe data sharing. In order to build a platform which can share data safely, blockchain becomes a good choice. We can build a distributed platform with trust way without central support by blockchain technology.

Zheng et al. [72] propose an architecture called MicrothingsChain. In this architecture, they proposed an EC network based on blockchain and every point's data are untamable and traceable. By designing Proof-of-Edge Computing Node which is based on Proof-of-Authority, data can be shared fairly. Besides, Truong et al. [73] propose an architect called Sash which transmits more data to the back end of blockchain to avoid malicious action by its own resilience. They also use smart contact to put Policy Decision Point into blockchain and analyse requests by access control which can both benefit the owners and the costumers to share data. In the field of Industry Internet of Things (IIoT), Liu et al. [74] propose an architecture which can collect and share data by blockchain and deep reinforcement learning. It divides points in private blockchain networks into computing and sharing and uses DRL to collect distributed data in IIoT.

Moreover, knowledge just as data in IoT networks can be shared safely and equally. Lin et al. [75] propose a market based on edge-enabled IoT with AI by blockchain. Consortium blockchain and smart contract from blockchain are used to keep knowledge such as data trading fairly, efficiently, and safely. They design a new consensus mechanism called Proof-of-Trading which can reduce the cost of computing resources.

5.3. Blockchain for Edge-Enabled IoT Services' Efficiency. Application of blockchain for IoT can effectively ensure the safety of IoT services' data just as mentioned in part 1 and 2 before, but with the expansion of IoT services, the demand for computing sources will easily exceed the resources that the Internet can provide which impact the efficiency of IoT service. If this kind of situation happens, it may result in data overflow, service delay, and so on. However, it is impractical for now to solve the fundamental problem by only updating the computing ability of IoT devices. We introduce some research studies from different aspects below which help improve the efficiency of IoT services.

Khanji et al. [76] discuss the balance between cache capacity and computing ability to improve the efficiency of the whole system. They desired a mechanism by Geometric Programming which combines each data point of IoT networks to exchange data which can disperse a single device's cache to others. Fu et al. [77] introduce a method to solve the problem by cooperative computing which virtualizes the servers of data points into computation-intensive virtual machines and design a three-level cache to assign the computing properly. Chen et al. [78] design an algorithm based on game theory to solve the multihop computing offloading problems with normal and mining tasks in blockchain IoT services.

When discussing the offloading problem in edge IoT, Xu et al. [79] design an algorithm called BeCome which is monitoring EC devices' resource by blockchain ledgers and allocating computing resources by nondominated sorting genetic algorithm III (NSGA-III).

6. Open Issues and Challenges

Though the application of AI is expected to enhance the security of IoT services in EC, many serious problems should be wiped out before AI can finally be used to secure IoT.

Table 4: Current research studies in blockchain for edge-enabled IoT services with AI.

Reference	Problem addressed	Technique used
[68]	Access control, authentication management	Blockchain
[69]	Authentication management	Blockchain, HTTPS protocol, and HMAC technology
[70]	Confidentiality and reliability of IoT data	ECC asymmetric encryption, DH key exchange, RAF consensus protocol, blockchain
[71]	Integrality and confidentiality of IoT data	Crowdsourcing, blockchain, DBSCAN
[72]	IoT data sharing	Blockchain, edge computing
[73]	IoT data sharing	Smart contact
[74]	Collection and share of IIoT data	Blockchain, deep reinforcement learning
[75]	Knowledge sharing	P2P networks, smart contact, consortium blockchain
[76]	The balance between cache capacity and computing ability	Blockchain
[77]	IoT cache offloading and computing	Cooperative computing, blockchain
[78]	Multihop computing offloading	Game theory, blockchain
[79]	Edge computing offloading	Blockchain ledger, NSGA-III

6.1. ML-Based Security Schemes.

ML is an advisable choice to secure IoT services because of its ability to augment the analytical capabilities of IoT devices and there actually exist security schemes based on ML. However, most of these schemes have fatal defects that make it impractical to adopt them into IoT systems at present.

6.1.1. High Computation and Communication Cost.

Many ML-based security schemes have an obvious deficiency that a flood of training data is required by machines in order to deduce a feasible model to tackle practical issues and the feature-extraction [82] process is very complicated as well. Worse still, its computation and communication cost [82] is very high. So, it is an urgency for us to devise a new ML-based security scheme with low computation and communication costs.

6.1.2. Backup Security Solutions.

Deep learning (DL) and reinforcement learning (RL) are two different types of ML and they have shortcomings, respectively. DL may fail to detect attacks precisely due to overfitting or insufficient training data. Hence, a suitable training dataset is a key for DL to reducing error rates. Then, let us talk about RL. Existing RL-based schemes are feasible merely on the premise that the intelligent agent knows the accurate state and is capable of evaluating the feedback of each action timely [82]. However, in fact, RL usually learns from scratch so that security schemes based on RL often lack the capability to handle attacks at the very beginning of the learning process, which increases the risk of IoT being attacked. So, to further secure IoT services, reliable backup security solutions should be designed in case of failures of ML-based schemes.

6.2. Adopt ML in Blockchain Technology.

IoT is maturing rapidly, and IoT services are gradually infiltrating into every aspect of our life. However, IoT is doomed to encounter cyber-attacks and undergo a security threat in its developing process. Moreover, trust problems hindering the information exchange among different IoT devices also act as obstacles to IoT's future advancement. Fortunately, blockchain technology can be used in IoT to facilitate security and resolve trust problems thanks to its nature of decentralization [83], ultimately optimizing IoT services.

Meanwhile, new security problems such as double spending and majority attack [28] come with the application of blockchain. Therefore, the help of ML technologies is instrumental in preventing underlying attacks to blockchains, but there is still much work to be done before we can successfully integrate ML and blockchain to enhance the security of IoT.

In view of the fact that data stored in the blockchain can be accessed by all blockchain nodes, privacy problems are worthy of our great attention. Private blockchains [27] and encryption have been utilized to solve privacy problems, but paradoxically, this will inevitably lead to limited and even insufficient training data for ML, making it difficult to acquire a satisfactory model for privacy protection [28]. When used in real scenarios, chances are that the performance of these models may fail to live up to our expectations and finally let us down.

7. Conclusions

As a new computational paradigm which provides the various solutions to the challenges of traditional cloud faces, EC will greatly promote the development of the IoT field and enrich the diversity of the IoT application ecosystem. Reliable privacy protection and security mechanisms are indispensable for high-quality IoT services, putting strict requirements on the privacy and security of EC. In this paper, the survey of the combination of AI and EC in IoT security is presented. Firstly, the basic concepts and definitions are introduced. Then, the IoT service framework with EC is summarized. Afterward, conventional and AI-driven privacy preservation of edge-based IoT are compared and the latter are elaborated. The collaboration of blockchain and AI on IoT security is also discussed. Finally, the paper talks about the open challenges and issues on AI for securing IoT services in EC.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research is supported by the Financial and Science Technology Plan Project of Xinjiang Production and Construction Corps under grant no. 2020DB005. Also, the research was supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund.

References

[1] F. X. Ming, R. A. A. Habeeb, F. H. B. Md Nasaruddin, and A. B. Gani, "Real-time carbon dioxide monitoring based on iot & cloud technologies," in *Proceedings of the 2019 8th International Conference on Software and Computer Applications*, pp. 517–521, Cairo, Egypt, April 2019.

[2] S. Moin, A. Karim, Z. Safdar, K. Safdar, E. Ahmed, and M. Imran, "Securing IoTs in distributed blockchain: analysis, requirements and open issues," *Future Generation Computer Systems*, vol. 100, pp. 325–343, 2019.

[3] F. Chu, S. Yuan, and Z. Peng, "Using machine learning techniques to identify botnet traffic," *Encyclopedia of Structural Health Monitoring*, pp. 967–974, Wiley, Hoboken, NJ, USA, 2006.

[4] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge computing security: state of the art and challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1608–1631, 2019.

[5] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[6] X. Xia, F. Chen, Q. He, J. Grundy, M. Abdelrazek, and H. Jin, "Cost-effective app data distribution in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 31–44, 2020.

[7] P. K. Manadhata and J. M. Wing, "A formal model for a system's attack surface," in *Moving Target Defense*, pp. 1–28, Springer, Berlin, Germany, 2011.

[8] P. Lai, Q. He, M. Abdelrazek et al., "Optimal edge user allocation in edge computing with variable sized vector bin packing," in *Proceedings of the International Con-Ference on Service-Oriented Computing*, pp. 230–245, Springer, Hangzhou, China, November 2018.

[9] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[10] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[11] M. A. Amanullah, R. A. A. Habeeb, F. H. Nasaruddin et al., "Deep learning and big data technologies for IoT security," *Computer Communications*, vol. 151, pp. 495–517, 2020.

[12] X. Chi, C. Yan, H. Wang, W. Rafique, and L. Qi, "Amplified locality-sensitive hashing-based recommender systems with privacy protection," *Concurrency and Computation: Practice and Experience*, 2020.

[13] W. Zhong, X. Yin, X. Zhang et al., "Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment," *Computer Communications*, vol. 157, pp. 116–123, 2020.

[14] G. Song and W. Chai, "Collaborative learning for deep neural net-works," in *Proceedings of the Advances in Neural Information Processing Sys-Tems*, pp. 1832–1841, Montreal, Canada, December 2018.

[15] H. A. Khattak, M. A. Shah, S. Khan, I. Ali, and M. Imran, "Perception layer security in internet of things," *Future Generation Computer Systems*, vol. 100, pp. 144–164, 2019.

[16] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the internet of things," in *proceedings of the IEEE World Congress on Services*, pp. 21–28, IEEE, New York, NY, USA, June 2015.

[17] L. Li, T.-T. Goh, and D. Jin, "How textual quality of online reviews affect classification performance: a case of deep learning sentiment analysis," *Neural Computing and Applications*, vol. 32, no. 9, pp. 4387–4415, 2020.

[18] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[19] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, and L. Qi, "Trust-oriented IoT service placement for smart cities in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4084–4091, 2019.

[20] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of Internet of Things (IoT): a survey," *Journal of Network and Computer Applications*, vol. 161, Article ID 102630, 2020.

[21] F. Liang, W. G. Hatcher, W. Liao, W. Gao, and W. Yu, "Machine learning for security and the internet of things: the good, the bad, and the ugly," *IEEE Access*, vol. 7, pp. 158126–158147, 2019.

[22] H. Haddadpajouh, R. Khayami, A. Dehghantanha, K. R. Choo, and R. M. Parizi, "AI4SAFE-IoT: an AI-powered secure archi-tecture for edge layer of internet of things," *Neural Comput-Ing and Applications*, 2020.

[23] P. Vimal and K. Shivaprakasha, "IoT based greenhouse environ-ment monitoring and controlling system using arduino plat-form," in *Proceedings of the International Conference on Intelligent Com-Puting, Instrumentation and Control Technologies (ICICICT)*, pp. 1514–1519, IEEE, Kannur, India, July 2017.

[24] A. Mohsin and S. S. Yellampalli, "IoT based cold chain logistics monitoring," in *Proceedings of the IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, pp. 1971–1974, IEEE, Chennai, India, September 2017.

[25] T. Adiono, B. A. Manangkalangi, R. Muttaqin, S. Harimurti, and W. Adijarto, "Intelligent and secured software application for iot based smart home," in *Proceedings of the IEEE 6th Global Conference on Consumer Electronics (GCCE)*, pp. 1-2, IEEE, Nagoya, Japan, October 2017.

[26] F. Franchi, A. Marotta, C. Rinaldi, F. Graziosi, and L. D'Errico, "IoT-based disaster management system on 5G uRLLC network," in *Proceedings of the International Conference on Information and Commu-Nication Technologies for Disaster Management (ICT-DM)*, pp. 1–4, IEEE, Paris, France, December 2019.

[27] T. Malche and P. Maheshwary, "Internet of things (IoT) for build-ing smart home system," in *Proceedings of the*

International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), IEEE, Coimbatore, India, pp. 65–70, February 2017.

[28] S. Chaudhary, R. Johari, R. Bhatia, K. Gupta, and A. Bhatnagar, "CRAIoT: concept, review and application(s) of IoT," in *Proceedings of the 4th International Conference on Internet of Things: Smart In-Novation and Usages (IoT-SIU)*, pp. 1–4, IEEE, Ghaziabad, India, April 2019.

[29] W. Shi, X. Zhang, Y. Wang, and Q. Zhang, "Edge computing: state-of-the-art and future directions," *Journal of Computer Research and Development*, vol. 56, no. 1, pp. 69–89, 2019.

[30] A. Jonathan, M. Ryden, K. Oh, A. Chandra, and J. Weissman, "Nebula: distributed edge cloud for data intensive computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 11, pp. 3229–3242, 2017.

[31] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[32] X. Zhang, Q. Chen, X. Peng, and X. Jiang, "Diffierential privacy-based indoor localization privacy protection in edge com-puting," in *Proceedings of the IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scal-Able Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pp. 491–496, IEEE, Leicester, UK, August 2019.

[33] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, and Q. Li, "Lavea: latency-aware video analytics on edge computing platform," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pp. 1–13, San Jose, CA, USA, October 2017.

[34] K. Intharawijitr, K. Iida, H. Koga, and K. Yamaoka, "Practical en-hancement and evaluation of a low-latency network model us-ing mobile edge computing,"vol. 1, pp. 567–574, in *Proceedings of the IEEE 41st Annual Com-puter Software and Applications Conference (COMPSAC)*, vol. 1, pp. 567–574, IEEE, Turin, Italy, July 2017.

[35] T. Cai, J. Li, A. S. Mian, R. Li, T. Sellis, and J. X. Yu, "Target-aware holistic influence maximization in spatial social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 4347, p. 1, 2020.

[36] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, "Community-diversified influence maximization in social networks," *Information Systems*, vol. 92, Article ID 101522, 2020.

[37] H. Liu, H. Kou, C. Yan, and L. Qi, "Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph," *Complexity*, vol. 2020, Article ID 2085638, 15 pages, 2020.

[38] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Transactions on Cloud Computing*, p. 1, 2019.

[39] L. Wang, X. Zhang, R. Wang, C. Yan, H. Kou, and L. Qi, "Diversified service recommendation with high accuracy and efficiency," *Knowledge-Based Systems*, vol. 204, Article ID 106196, 2020.

[40] Z. Ziming, L. Fang, C. Zhiping, and X. Nong, "Edge computing: platforms, applications and challenges," *Journal of Computer Research and Development*, vol. 55, no. 2, p. 327, 2018.

[41] C. Avasalcai, C. Tsigkanos, and S. Dustdar, "Decentralized resource auctioning for latency-sensitive edge computing," in *Proceedings of the IEEE International Conference on Edge Computing (EDGE)*, pp. 72–76, IEEE, Milan, Italy, July 2019.

[42] Q. He, G. Cui, X. Zhang et al., "A game-theoretical approach for user allocation in edge computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 3, pp. 515–529, 2019.

[43] A. Gatouillat, Y. Badr, B. Massot, and E. Sejdic, "Internet of medical things: a review of recent contributions dealing with cyber-physical systems in medicine," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3810–3822, 2018.

[44] M. Capra, R. Peloso, G. Masera, M. R. Roch, and M. Martina, "Edge computing: a survey on the hardware requirements in the internet of things world," *Future Internet*, vol. 11, no. 4, p. 100, 2019.

[45] L. Zhang, A. Afanasyev, J. Burke et al., "Named data networking," *ACM Sigcomm Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[46] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge comput-ing benefit from software-defined networking: a survey, use cases & future directions," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, p. 1, 2017.

[47] S. A. Noghabi, L. Cox, S. Agarwal, and G. Ananthanarayanan, "The emerging landscape of edge computing," *GetMobile: Mobile Computing and Communications*, vol. 23, no. 4, pp. 11–20, 2020.

[48] Z. Zhao, G. Min, W. Gao, Y. Wu, H. Duan, and Q. Ni, "Deploying edge computing nodes for large-scale iot: a diversity aware approach," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3606–3614, 2018.

[49] M. Sapienza, E. Guardo, M. Cavallo, G. La Torre, G. Leombruno, and O. Tomarchio, "Solving critical events through mo-bile edge computing: an approach for smart cities," in *Proceedings of the IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 1–5, IEEE, St. Louis, MO, USA, May 2016.

[50] M. Caria, J. Schudrowitz, A. Jukan, and N. Kemper, "Smart farm computing systems for animal welfare monitoring," in *Proceedings of the 40th International Convention on Information and Com-Munication Technology, Electronics and Microelectronics (MIPRO)*, pp. 152–157, IEEE, Opatija, Croatia, May 2017.

[51] L. Liu, X. Zhang, M. Qiao, and W. Shi, "Safeshareride: edge-based attack detection in ridesharing services," in *Proceedings of the IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 17–29, IEEE, Seattle, WA, USA, October 2018.

[52] J. Cao, L. Xu, R. Abdallah, and W. Shi, "EdgeOS_H: a home op-erating system for internet of everything," in *Proceedings of the IEEE 37th International Conference on Distributed Computing Sys-Tems (ICDCS)*, pp. 1756–1764, IEEE, Atlanta, GA, USA, June 2017.

[53] Q. Zhang, Q. Zhang, W. Shi, and H. Zhong, "Distributed collaborative execution on the edges and its application to AMBER alerts," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3580–3593, 2018.

[54] M. Zheng, D. Xu, L. Jiang, C. Gu, R. Tan, and P. Cheng, "Challenges of privacy-preserving machine learning in IoT," in *Proceedings of the First International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, pp. 1–7, New York, NY, USA, November 2019.

[55] B. Zhou and J. Pei, "The *k*-anonymity and *l*-diversity approaches for privacy preservation in social networks against neighborhood attacks," *Knowledge and Information Systems*, vol. 28, no. 1, pp. 47–77, 2011.

[56] N. Li, T. Li, and S. Venkatasubramanian, "*t*-closeness: privacy be-yond *k*-anonymity and *l*-diversity," in *Proceedings of the*

IEEE 23rd International Conference on Data Engineering, pp. 106–115, IEEE, Istanbul, Turkey, April 2007.

[57] L. Sweeney, "*k*-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[58] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41–47, Washington, DC USA, November 2002.

[59] R. Kalaivani and S. Chidambaram, "Additive Gaussian noise based data perturbation in multi-level trust privacy preserving data mining," *International Journal of Data Mining & Knowledge Management Process*, vol. 4, no. 3, pp. 21–29, 2014.

[60] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based mul-tiplicative data perturbation for privacy preserving distributed data mining," *IEEE Transactions on Knowledge and Data En-Gineering*, vol. 18, no. 1, pp. 92–106, 2005.

[61] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "Local diffierential privacy for deep learn-ing," *IEEE Internet of Things Journal*, vol. 7, pp. 5827–5842, 2020.

[62] Y. Donghuan, M. Wen, X. Liang, Z. Fu, K. Zhang, and B. Yang, "Energy theft detection with energy privacy preservation in the smart grid," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7659–7669, 2019.

[63] D. Xu, M. Zheng, L. Jiang, C. Gu, R. Tan, and P. Cheng, "Lightweight and unobtrusive privacy preservation for remote inference via edge data obfuscation," 2019, https://arxiv.org/abs/1912.09859.

[64] X. Xu, X. Liu, X. Yin, S. Wang, Q. Qi, and L. Qi, "Privacy-aware offloading for training tasks of generative adversarial network in edge computing," *Information Sciences*, vol. 532, pp. 1–15, 2020.

[65] J. Chi, E. Owusu, X. Yin et al., "Privacy partition: a privacy-preserving framework for deep neural networks in edge networks," in *Proceedings of the IEEE/ACM Symposium on Edge Computing (SEC)*, IEEE, Seattle, WA, USA, pp. 378–380, October, 2018.

[66] J. Chi, E. Owusu, X. Yin et al., "Privacy partitioning: protecting user data during the deep learning inference phase," 2018, https://arxiv.org/abs/1812.02863.

[67] H. Gamage, H. Weerasinghe, and N. Dias, "A survey on blockchain technology concepts, applications, and issues," *SN Computer Science*, vol. 1, pp. 1–15, 2020.

[68] A. Z. Ourad, B. Belgacem, and K. Salah, "Using blockchain for iot access control and authentication management," in *Proceedings of the Interna-tional Conference on Internet of Things*, pp. 150–164, Springer, Santa Barbara, CA, USA, October 2018.

[69] Y. Cheng, M. Lei, S. Chen, Z. Fang, and S. Yang, "IoT security ac-cess authentication method based on blockchain," in *Proceedings of the Interna-tional Conference on Advanced Hybrid Information Process-Ing*, pp. 229–238, Springer, Nanjing, China, September 2019.

[70] H. Rui, L. Huan, H. Yang, and Z. YunHao, "Research on secure transmission and storage of energy IoT Informa-tion based on blockchain," *Peer-to-Peer Networking and Applications*, vol. 13, no. 4, pp. 1225–1235, 2019.

[71] X. Xu, Q. Liu, X. Zhang, J. Zhang, L. Qi, and W. Dou, "A blockchain-powered crowdsourcing method with privacy preservation in mobile environment," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1407–1419, 2019.

[72] J. Zheng, X. Dong, T. Zhang, J. Chen, W. Tong, and X. Yang, "Mi-crothingschain: edge computing and decentralized IoT archi-tecture based on blockchain for cross-domain data shareing," in *Proceedings of the International Conference on Networking and Net-Work Applications (NaNA)*, pp. 350–355, IEEE, Xi'an, China, October 2018.

[73] H. T. T. Truong, M. Almeida, G. Karame, and C. Soriente, "To-wards secure and decentralized sharing of IoT data," in *Proceedings of the IEEE International Conference on Blockchain (Blockchain)*, pp. 176–183, IEEE, Seoul, Korea, May 2019.

[74] C. H. Liu, Q. Lin, and S. Wen, "Blockchain-enabled data collection and sharing for industrial iot with deep reinforcement learn-ing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3516–3526, 2018.

[75] X. Lin, J. Li, J. Wu, H. Liang, and W. Yang, "Making knowledge tradable in edge-ai enabled iot: a consortium blockchain-based efficient and incentive approach," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6367–6378, 2019.

[76] S. Khanji, F. Iqbal, Z. Maamar, and H. Hacid, "Boosting iot efficiency and security through blockchain: blockchain-based car insurance process-a case study," in *Proceedings of the 4th International Conference on System Reliability and Safety (ICSRS)*, pp. 86–93, IEEE, Rome, Italy, November 2019.

[77] S. Fu, L. Zhao, X. Ling, and H. Zhang, "Maximizing the system energy efficiency in the blockchain based internet of things," in *Proceedings of the ICC 2019-IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Shanghai, China, May 2019.

[78] W. Chen, Z. Zhang, Z. Hong et al., "Cooperative and distributed computation offloading for blockchain-empowered industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8433–8446, 2019.

[79] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "Become: blockchain-enabled computation offloading for iot in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2019.

[80] M. Singh, A. Singh, and S. Kim, "Blockchain: a game changer for securing IoT data," in *Proceedings of the IEEE 4th World Forum on Internet of Things (WF-IoT)*, pp. 51–55, IEEE, Singapore, February 2018.

[81] S. H. Hashemi, F. Faghri, P. Rausch, and R. H. Campbell, "World of empowered IoT users," in *Proceedings of the IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 13–24, IEEE, Berlin, Germany, April 2016.

[82] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning: how do IoT devices use AI to enhance security?" *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41–49, 2018.

[83] H. Desai, M. Kantarcioglu, and L. Kagal, "A hybrid blockchain architecture for privacy-enabled and accountable auctions," in *Proceedings of the IEEE International Conference on Blockchain (Blockchain)*, pp. 34–43, IEEE, Atlanta, GA, USA, July 2019.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8831341, 10 pages
https://doi.org/10.1155/2020/8831341

Research Article

Collaborative Intelligence: Accelerating Deep Neural Network Inference via Device-Edge Synergy

Nanliang Shan, Zecong Ye, and Xiaolong Cui

College of Information Engineering, Engineering University of PAP, Xi'an 710086, China

Correspondence should be addressed to Xiaolong Cui; 18182437082@163.com

Received 31 May 2020; Revised 11 July 2020; Accepted 31 July 2020; Published 7 September 2020

Academic Editor: Xiaolong Xu

With the development of mobile edge computing (MEC), more and more intelligent services and applications based on deep neural networks are deployed on mobile devices to meet the diverse and personalized needs of users. Unfortunately, deploying and inferencing deep learning models on resource-constrained devices are challenging. The traditional cloud-based method usually runs the deep learning model on the cloud server. Since a large amount of input data needs to be transmitted to the server through WAN, it will cause a large service latency. This is unacceptable for most current latency-sensitive and computation-intensive applications. In this paper, we propose Cogent, an execution framework that accelerates deep neural network inference through device-edge synergy. In the Cogent framework, it is divided into two operation stages, including the automatic pruning and partition stage and the containerized deployment stage. Cogent uses reinforcement learning (RL) to automatically predict pruning and partition strategies based on feedback from the hardware configuration and system conditions so that the pruned and partitioned model can better adapt to the system environment and user hardware configuration. Then through containerized deployment to the device and the edge server to accelerate model inference, experiments show that the learning-based hardware-aware automatic pruning and partition scheme can significantly reduce the service latency, and it accelerates the overall model inference process while maintaining accuracy. Using this method can accelerate up to 8.89× without loss of accuracy of more than 7%.

1. Introduction

As the backbone technology to support modern intelligent services and applications, deep neural network (DNN) has become more and more popular due to their superior performance in computer vision [1], speech recognition [2], natural language processing [3], and big data analysis [4]. With the development of mobile edge computing (MEC), more and more intelligent services and applications based on DNN are deployed on mobile terminal devices to meet the diverse and personalized needs of users. Unfortunately, today's mobile devices cannot support these DNN-based intelligent services and applications well because these intelligent models usually require a lot of computing resources.

To solve a large number of service resource requirements of the DNN model, the traditional method relies on powerful cloud servers to provide rich computing power. In this case, the training and inference data generated on the mobile device are transmitted to the remote cloud server, and the inference result is returned after the execution is completed. However, many intelligent applications require frequent information interaction, which makes large amounts of data to be transmitted between mobile devices and remote cloud server through a long wide area network. This will bring intolerable communication latency, making cloud-based computing impractical. To solve this problem, we explored the emerging computing architecture of device-edge synergy. By adding an edge service node composed of an edge server and an edge microbase station near the terminal device, the cloud computing capability is sunk from the core network to the edge network closer to the user. This new computing architecture provides sufficient computing power support for DNN-based computing-intensive and latency-sensitive applications at the edge network. So how to effectively deploy the intelligent model based on DNN to the edge and make full use of the rich computing resources of the edge server to perform model inference (i.e., edge intelligence) [5] to minimize the service latency will be the main consideration in this paper.

In response to the above problems, the predecessors have made many efforts. These include collaborative computing between terminal devices and cloud servers [6–8], model compression and parameter pruning [9–12], or customized mobile implementation [13–15]. Despite all these efforts made by the predecessors, on the premise of ensuring the accuracy of the model required by the user, the service latency is minimized, and the user's hardware configuration and system status can be sensed to implement automatic model pruning and partition. The current edge intelligence architecture still has major defects.

On this issue, this paper proposes a device-edge synergy framework Cogent, which uses reinforcement learning (RL) to achieve automatic pruning and partition of models. And through the container technology, the divided model blocks are packaged and deployed on the edge server and the terminal device, and the rich computing resources of the edge server are used to accelerate the model inference collaboratively. The Cogent framework is a latency-sensitive collaborative intelligent design. It is mainly divided into two operation stages, namely, automatic pruning and partition stage and containerized deployment stage. In the automated pruning and partition phase, Cogent uses RL to observe hardware accelerators and system status (including network bandwidth and edge server load) and provides model pruning and partition strategies. We observe that the accuracy of the compression model is very sensitive to the sparsity of each layer and requires a fine-grained action space. Therefore, instead of searching in discrete space, we propose a continuous compression ratio control strategy with DDPG [16] agent, which learns through trial and error and penalizes loss of accuracy, while encouraging model acceleration and reduction. Specifically, our DDPG agent handles the network model in an integrated and layered approach. For the overall network model, the agent receives network structure information of the entire model, system network bandwidth B, hardware accelerator information A, and edge server load information E, and then it outputs the model partition point. For each layer, the agent receives the state information S_t and hardware accelerator information A, and then it outputs the precise pruning ratio of each layer. Our Cogent framework automates this process through learning-based strategies rather than relying on rule-based strategies and experienced engineers. In the containerized deployment phase, we use Docker and Kubernetes to dynamically package model blocks and assign containers to one or more available devices to complete a DNN task, which greatly increases the flexibility and reliability of Cogent. Cogent makes full use of device-edge synergy to achieve collaborative intelligence, which can minimize inference latency while meeting user accuracy requirements.

To summarize, we present the contribution of this paper as follows:

(i) We propose Cogent, an execution framework that accelerates deep neural network inference through device-edge synergy. We use Cogent automated pruning and partition to jointly optimize DNN model inference to minimize service latency while ensuring user accuracy requirements.

(ii) We propose an automated DNN model pruning and partition algorithm, which uses reinforcement learning to determine pruning and partition strategies automatically. At the same time, we receive the feedback of the hardware accelerator and system state in the design cycle, so that the pruned and partitioned models can better adapt to different hardware architectures and system conditions, greatly reducing service latency.

(iii) We use Docker and Kubernetes to dynamically package model blocks and assign containers to the edge server and terminal devices to complete a DNN task cooperatively. It not only makes full use of the rich computing resources of the edge server to accelerate the inference process but also greatly increases the flexibility and reliability of Cogent.

The rest of this paper is organized as follows. First, we review the related work in Section 2. The proposed overall framework of Cogent is introduced in Section 3. The results of the performance evaluation are shown in Section 4 to demonstrate the effectiveness of Cogent. Finally, the paper is concluded in Section 5.

2. Related Work

The rapid development of DNN makes it quickly become one of the most important components of artificial intelligence technology today. DNN consists of a series of network layers, and each layer of network consists of a group of neurons. DNN is widely used in the field of computer vision and natural language processing, including image classification, target detection, video recognition, and text processing. At present, edge intelligence technology has attracted the attention of researchers. To implement artificial intelligence at the edge, edge intelligence technology deploys DNN models on mobile devices that are closer to users to enable more flexible and safe interaction between users and smart models. However, due to the resource limitations of terminal devices, it has become very challenging to directly deploy and infer computation-intensive DNN models on edge devices. On this issue, existing efforts are devoted to optimizing DNN calculations on edge devices. There are three main areas worthy of attention here.

2.1. Optimize DNN Model. DNN model optimization is used to include model structure optimization and hardware acceleration. In terms of model structure optimization, some researchers tried to develop new DNN structures to achieve the desired accuracy under moderate calculations, such as DNN models that were much smaller than normal network models without sacrificing excessive accuracy [13]. Also, in order to reduce the amount of data transmission during DNN inference, the DNN model was compressed by model pruning [17–19]. Others were focused on reducing the redundancy in the original model by the model compression techniques [20–22] to obtain an effective model. Recent advances in this optimization had turned to network architecture search (NAS) [23–25]. In terms of hardware acceleration, mobile devices could be embedded with deep learning inference chips to improve latency and energy efficiency with the help of architectural acceleration technology [26, 27]. Other works were aimed at optimizing the use of existing resources [28–30] and improving service performance [31–34].

2.2. Device-Edge Synergy. The most involved in the mobile device and edge server collaboration were model partition and related technologies. DNN model partition technology referred to partitioning a specific DNN model into some continuous parts and deploying these parts on multiple participating devices. The goal of model partition technology was similar to computation offloading and aimed to maximize the use of external computing resources to accelerate mobile edge computing. For example, some frameworks used DNN partition to optimize computation offloading between mobile devices and the cloud, while other frameworks aimed to distribute computing workload among mobile devices [35–37]. The most critical technique of model partition lied in the choice of partition point. In the work [38], the DNN partitioning problem was transformed into the shortest path problem, and the approximate solution was used to solve the problem. At the same time, they also used PNG encoding to reduce the amount of intermediate data transmission. To study the influence of network conditions and server load during the partition process, Kang [7] studied the hierarchical partition of the DNN model, and the variation of latency with server load changed under three typical wireless communication conditions. Besides, an improved DNN structure had been proposed for device-edge synergy [8, 39], where early exit network branches were added to the original network. Their evaluation proved the effectiveness of the improved DNN structure in low-latency inference and accuracy assurance.

2.3. Automatic Machine Learning (AutoML). Besides, many research efforts aimed to improve the performance of DNNs through an automated search of network structures: NAS [40] aimed to search for transferable network blocks whose performance exceeds many manually designed architectures. Progressive NAS [24] adopted sequential model-based optimization methods to accelerate architecture search by 5×. Pham et al. [25] introduced efficient NAS used parameter sharing to accelerate the speed of architecture exploration by 1000×. Cai et al. [41] introduced path-level network transformation to efficiently search the tree structure space. Driven by these AutoML frameworks, He et al. [42] leveraged reinforcement learning to automatically prune the convolution channel.

Compared with the current work, the Cogent framework designed in this paper makes a good combination of DNN model optimization, device-edge synergy, and AutoML. Cogent leverages reinforcement learning to automatically predict the pruning ratio of each layer and the partition point of the model. At the same time, it takes into account the hardware architecture and system state. Finally, through containerized deployment, the flexibility and reliability of the Cogent framework are greatly improved. Cogent can speed up model inference as much as possible while ensuring user accuracy requirements and at the same time has better adaptability to different hardware devices and system status. This provides a good choice for latency-sensitive service requests on mobile devices.

3. Overall Framework Design

3.1. Framework Overview. As shown in Figure 1, we proposed the design of the Cogent framework, which includes two operational stages, namely, automated pruning and partition stage and containerized deployment stage. First of all, the Cogent framework uses reinforcement learning to automatically search the huge pruning design space in the loop. Its RL agent integrates hardware accelerators and system status (including network bandwidth and edge server load) into the detection loop so that it can obtain direct feedback from the hardware and system status. Then, the agent proposes an optimal model partitioning and pruning strategy under the given amount of computing resources and network bandwidth. The automatic model pruning and partition algorithm on Cogent will perform model pruning and partition according to these strategies. Finally, the divided model blocks are packaged and delivered. The Cogent framework automates the pruning process and model partition process by using learning-based methods that take hardware- and system-state-specific metrics as direct rewards to meet the requirements of service request accuracy while minimizing service latency. We use the actor-critic model with the deep deterministic policy gradient (DDPG) agent to give actions: the pruning ratio of each layer and the partition point of the model. We collect hardware counters as constraints and use latency as a reward to search for optimal pruning and partition strategies. We have two hardware environments, including terminal device accelerators and edge server accelerators. The following describes the details of each element of reinforcement learning.

3.2. State Space. Our agent deals with neural networks in a combination of whole and layer. For the overall model, the agent needs to determine the most appropriate partition point. For each layer, the agent needs to determine the proportion of pruning for each layer. In this paper, we introduced an 11-dimensional feature vector as our state value S_t:

$$\left\{B, E, A_{m,c}, L_t\left(t, n, c, \text{FLOPs}, \text{ reduced}, \text{rest}, a_{t-1}, time\right)\right\}, \tag{1}$$

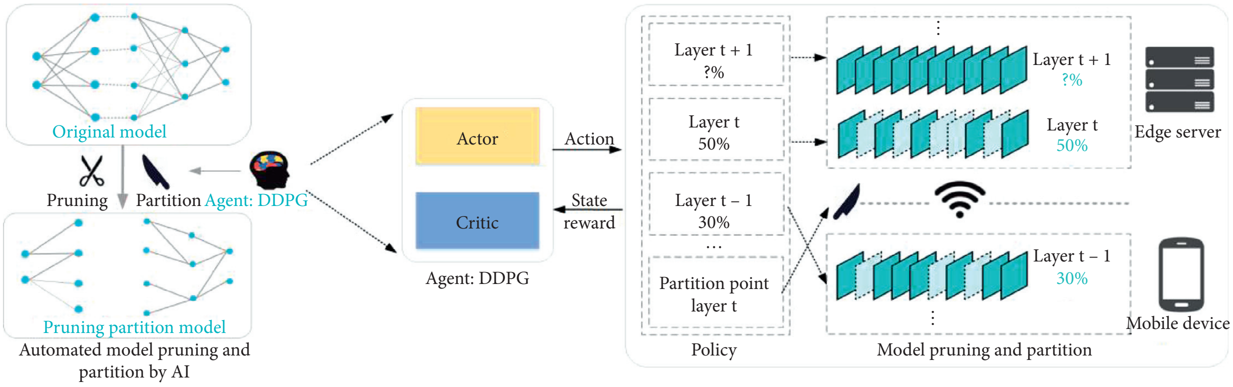


Figure 1: Overview of Cogent framework for model pruning and partition.

where B is the network bandwidth, E is the edge server load, and $A_{m,c}$ is the hardware accelerator configuration, which usually refers to the CPU speed of the mobile terminal device and the CPU speed of the edge server. t is the layer index, n is the dimension of the core, c is the input of this layer, FLOPs is the FLOPs calculation of the layer, reduced is the total number of calculations reduced in previous layers, rest is the number of calculations remaining in the subsequent layer, a_{t-1} is the pruning ratio selected by the upper layer, and *time* is the inference time spent on this layer. Before being passed to the agent, they are scaled within [0, 1]. These features are essential for the agent to distinguish one network layer from another.

3.3. Action Space. For the partition point p, we use the discrete space as the action space, because the choice of p is fixed and limited. For the pruning ratio of each layer, most of the existing works use discrete space as a coarse-grained action space. For high-accuracy model architecture search, coarse-grained action space may not be a problem. However, we observe that model compression is very sensitive to the sparse ratio, which leads to a surge in the number of discrete actions, so we need a more fine-grained action space. Otherwise, such a large action space will be difficult to effectively explore [16]. At the same time, discretization will also make the action selection jumpy and may miss the optimal sparse ratio. Therefore, we suggest using the continuous action space $a \in (0, 1]$, which can achieve more fine-grained and more accurate model pruning.

3.4. DDPG Agent. As shown in Figure 1, we first select the partition point p before the agent starts to determine the pruning rate for each layer. The choice of the partition point is mainly affected by the network architecture. Of course, the agent will also adjust the decision based on the hardware accelerator and system status. After determining the partition point, there is no need to partition immediately, but first determine the pruning ratio of each layer. The agent receives an embedding state S_t of layer L_t from the environment and then outputs a sparse rate as action a_t. The agent then moves to the next layer L_{t+1} and receives state S_{t+1}. After the final layer of the decision is made, the accuracy of the model is evaluated on the validation set and returned to the agent. On the premise of ensuring the accuracy requirements of users, according to the decision set, the specified compression algorithm (e.g., channel pruning) is used to compress the model. To improve the speed of exploration, we only evaluate the accuracy without fine-tuning, which is a good method to approximate the precision of fine-tuning. At this time, we will get the optimal pruning ratio of each layer for the hardware characteristics and system status when the current partition point is p.

For agent decision a_t, we use DDPG to continuously control the compression ratio. For the noise distribution during exploration, we use a truncated normal distribution. The noise σ is initialized to 0.5 and decays exponentially after each episode:

$$\mu'(s_t) \sim TN\left(\mu\left(s_t \,\middle|\, \theta_t^{\mu}\right), \sigma^2, 0, 1\right). \tag{2}$$

The design of the DDPG agent follows Block-QNN [43], applying a variant of the Bellman equation [44]. Each state input of each episode is (S_t, a_t, R, S_{t+1}), where R is the reward after pruning the network. In the update process, to reduce the gradient loss, the gradient estimate needs to subtract the baseline reward b, which is equivalent to the exponential moving average of the previous reward [40]:

$$\begin{aligned} \text{Loss} &= \frac{1}{N}\sum_i \left(y_i - Q\left(s_i, a_i \,\middle|\, \theta^Q\right)\right)^2, \\ y_i &= r_i - b + \gamma Q\left(s_{i+1}, \mu\left(s_{i+1}\right) \,\middle|\, \theta^Q\right). \end{aligned} \tag{3}$$

The discount factor γ is set to 1 to avoid overprioritizing short-term rewards.

3.5. Reward Function. By adjusting the reward function, we can accurately find the limit of compression and minimize service latency. Using the reward function to limit the action space (sparse rate of each layer), we can accurately obtain the target accuracy. Take fine-grained pruning to reduce inference latency as an example: we allow arbitrary operations in the first few layers. When we find that we are close to the target accuracy, we begin to limit action a, pruning all the following layers with the most conservative strategy. Our reward function is as follows:

$$R = \lambda \times \Delta\text{latency}, \quad (4)$$

where Δlatency is the latency difference before and after pruning and λ is a scaling factor, which is set to 0.1 in our experiment.

3.6. Automated Model Pruning and Partition. Our goal is to find the optimal partition point of the model first and then find the redundancy of each layer according to the hardware environment that the preparatition part will configure in the future. The optimal partition point of a DNN model depends on the topology of the DNN, which is reflected in the change of the amount of calculation and data of each layer. Besides, even for the same DNN structure, dynamic factors such as wireless network bandwidth and edge server load will affect the choice of optimal partition point. For example, the instability of the wireless network bandwidth will directly affect the transmission latency between the mobile device and the edge server, and the load change of the edge server will directly affect the queuing latency or calculation latency of the application request at the edge server. We train an RL agent to predict the partition point and pruning ratio and then perform pruning and partition. We quickly assess the accuracy after pruning before fine-tuning, as an effective representation of the final accuracy. Then, we update the agent by rewarding faster, smaller, and more accurate models. We introduced the AutoML process of the Cogent framework in Algorithm 1.

Cogent first analyzes the constituent layers of the target DNN model and extracts the type and configuration (L_t) of each layer and then uses the RL agent to predict the latency T_m and T_c of the current layer on mobile devices and the edge server, respectively. At the same time, the current network bandwidth B and edge server load E should be considered. Line 10 of Algorithm 1 uses the agent to predict the output parameter amount D_t of the execution layer L_t on the mobile device. Line 13 of Algorithm 1 calculates the transmission latency T_t under the current wireless network bandwidth. Line 15 of Algorithm 1 evaluates the inference latency and inference accuracy of each candidate partition point and selects the partition point with the lowest inference latency under the premise of meeting the user's accuracy requirements. After determining the partition point, there is no need to partition immediately, because this will affect the subsequent pruning process. The pruning ratio needs to be comprehensively decided according to the hardware environment and system status of the model block to be deployed in the future. Also, the agent must ensure the accuracy required by the user when making decisions, which is a prerequisite for Cogent to perform pruning and partition. Finally, the Cogent framework will automatically perform model pruning and partition according to the agent's decision.

3.7. Containerized Deployment. We containerized each model block after partitioning and deployed the model by launching pods on the edge server and mobile devices through Kubernetes. Please note that the general model is configured to work together on one mobile device and one edge server. In this case, only two pods will be used for deployment. If multiple mobile devices request an application at the same time, multiple pods can be deployed. If the system status changes, Cogent will periodically recalculate the optimal partition point. Once the model execution graph changes, we will adjust the pod configuration and reschedule them. Besides, application service requests may fail in mobile edge networks. To quickly restore services without affecting the normal operation of the pods, Kubernetes assigned a static virtual IP to each pod. Each pod communicates with its upstream and downstream pods via virtual IP. The association between the virtual IP and the pod is based on the position of the pod in the model execution graph. If the service request fails, we can easily launch a new pod from the edge server and associate the new pod with the virtual IP. In this way, the application services can be kept in the normal operation of the mobile device without being affected. By containerizing each model block, we use Docker and Kubernetes to simplify model update and deployment and efficiently handle runtime resource management and scheduling of containers.

4. Evaluation

4.1. Experimental Setup. We use Xilinx Zynq-7020 FPGA [45] as our terminal device and Xilinx VU9P [46] as our edge server and prove the feasibility and efficiency of Cogent through design experiments. Table 1 shows our experimental configuration on both platforms and the resources available to them. Configure the inbound and outbound network bandwidth of each terminal device to connect to the edge server through the traffic control (TC) infrastructure. Besides, all physical servers run Ubuntu 18.04 and deploy the Kubernetes (Release-1.7) cluster. VGG19 [47] is a state-of-the-art image classification DNN, which serves as the target network for device-edge synergy inference in this paper. Our dataset is CIFAR-10 [48], a widely used image classification dataset with 10 classes of objects. Our network of actors μ has two hidden layers, each with 300 units. The final output layer is a Sigmoid layer that binds the action within (0, 1). Our critic network Q also has two hidden layers, each with 300 units. We set the learning rate to 0.01, the batch size to 64, and the replay memory capacity to 2000. Our agent first explores 100 episodes with constant noise $\sigma = 0.5$ and then explores 300 episodes with exponential attenuation noise σ, with an attenuation coefficient of 0.99. We set the cloud-based computation offloading method [49] as the baseline and added the status quo method HierTrain [50] as a comparison. We compared the performance of the Cogent architecture with the baseline and status quo in terms of inference latency (Section 4.2). We also assessed the robustness of Cogent to the variation in wireless network bandwidth (Section 4.3) and server load (Section 4.4), demonstrating the importance of a dynamic runtime architecture for collaborative inference speedup. Finally, we verified the performance of the Cogent framework in terms of hardware awareness (Section 4.5).

4.2. Latency Improvement. In this section, we examine the latency improvement that can be brought about by using the Cogent collaborative intelligence framework proposed in this paper. Figure 2 shows per-layer execution latency and output data size after each layer's execution (input for next layer) of the baseline approach, the status quo approach, and Cogent

```
Input:
$N$: number of layers in the DNN
$\{L_t \mid t = 1,\ldots, N\}$: layers in the DNN
$Agent\ (L_t)$: reinforcement learning agent predicting the output parameters and latency of executing $L_t$
$B$: current wireless network uplink bandwidth
$E$: current edge server load
$H$: hardware accelerator's feedback
procedure THE FIRST STEP
    for each $t$ in $1 \ldots N$ do
        $D_t \longleftarrow \mathrm{Agent}_{\mathrm{moblie}}\,(L_t)$
        $T_m \xleftarrow{H_m} \mathrm{Agent}_{\mathrm{moblie}}\,(L_t)$
        $T_c \xleftarrow{H_c, E} \mathrm{Agent}_{\mathrm{cloud}}\,(L_t)$
        $\mathrm{T_t}\ T_t \longleftarrow D_t/B$
    end for
return Partition point $\longrightarrow$ $\boldsymbol{Action_p}$
    $j = \mathrm{argmin}_{j=1\ldots N}\left(\sum_{k=1}^{j} Tm_k + \sum_{k=j+1}^{N} Tc_k + Tt_j\right)$

procedure THE SECOND STEP
  for each $t$ in $1 \ldots N$ do
      if OptTarget $\geq$ $_{\min}$(Accuracy) then
        return Pruning ratio $\longrightarrow$ $Action_t\ P_t \xleftarrow{H_{m,c}} \mathrm{Agent}\,(L_t)$
      end if
  end for
return NULL
```

ALGORITHM 1: Automated model pruning and partition algorithm.

executing VGG model on the mobile device, respectively. The histogram in Figure 2 shows the per-layer execution latency, which shows that a fully cloud-based baseline approach has significantly more execution latency than the other two model partition methods due to network conditions and edge server load. The status quo method is mainly used to prune to the network layer at the back end of the model, so that the network layer at the front of the execution model will also face a large execution latency. Cogent architecture minimizes the latency of the network layer's execution on mobile devices by automatically pruning each layer of the VGG model. The broken line in Figure 2 shows the size of output data after each layer's execution. It can be seen that the network layer parameter output through the Cogent framework has been significantly reduced, which can fully reduce the transmission latency from the terminal device to the edge server. Combining the results of the per-layer execution latency and size of output data after each layer's execution, Cogent predicts that the best latency optimization can be obtained by partitioning the VGG model in the pool3 layer.

We show a comparison of the inference results of running the VGG model through three different methods in Table 2. From the second column of the table, we can see that Cogent can almost guarantee the inference accuracy of the VGG model. The third and fourth columns represent the proportion of the pruning parameters and the number of parameters remaining in the model, respectively. The fifth and sixth columns represent the proportion of the pruning calculation and the number of

TABLE 1: The configurations of device and edge accelerators.

	Hardware	Batch	PE array	AXI port	Block RAM
Device	Zynq-7020	1	8 × 8	4 × 64b	140 × 36 Kb
Edge server	VU9P	16	16 × 16	4 × 256b	2160 × 36 Kb

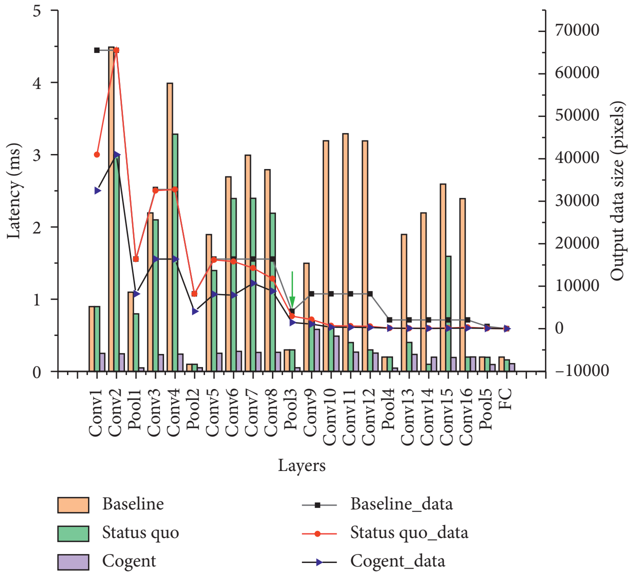


FIGURE 2: Comparisons of per-layer execution latency and output data size of different strategies.

Table 2: Comparisons of pruned parameter ratio and latency speedup of different strategies.

Prune process	Accuracy (%)	Pruned (%)	Parameter (M)	Pruned (%)	Mult-Adds (M)	Time-32
VGGNET (baseline)	94.64	—	20.3	—	398.14	69.0641 ms
VGGNET (status)	93.34	67.5	6.51	37.2	250.03	24.8501 ms (2.78×)
VGGNET (cogent)	93.82	88.5	2.31	50.8	195.87	7.7703 ms (8.89×)

residual multiplier calculations, respectively. The last column shows the average inference time when the input test image set is $32 * 32$. From the table, the inference latency acceleration of the Cogent framework reached a maximum of 8.89×.

Since the loss of information in the feature mapping pruning may affect the accuracy of the model, we study the trade-off between the percentage of pruning parameters and the accuracy of the model. Figure 3 shows the trade-off between the accuracy loss of Cogent and the percentage of pruning parameters. This curve represents the accuracy loss threshold of the model implemented on the CIFAR-10 dataset, which corresponds to the percentage of model pruning parameters by the Cogent architecture. We observed that, for VGG19 networks, the percentage of pruning parameters less than 90% can guarantee the accuracy loss less than 5%.

4.3. Impact of Network Bandwidth Variation. In this section, we evaluate the resilience of Cogent to the variation in wireless network bandwidth between terminal devices and the edge server. In Figure 4, the purple line shows the wireless bandwidth we configure through the traffic control (TC) infrastructure. The green and orange curves show the end-to-end latency of the status quo method and Cogent performing VGG19 on the mobile device platform, respectively. As you can see, the status quo approach is easily affected by network bandwidth variation, so application latency increases significantly during the low bandwidth phase. In contrast, Cogent can effectively adapt to variation in network bandwidth and provide consistent low latency. The main reason is that Cogent can dynamically adjust the partition point and pruning ratio based on the available bandwidth to change the amount of data transmission, thereby minimizing the impact of network bandwidth variation.

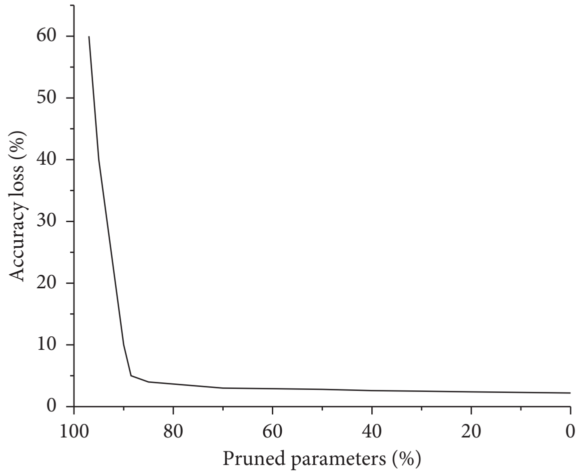


Figure 3: Accuracy loss versus the percentage of the pruned parameter with Cogent.

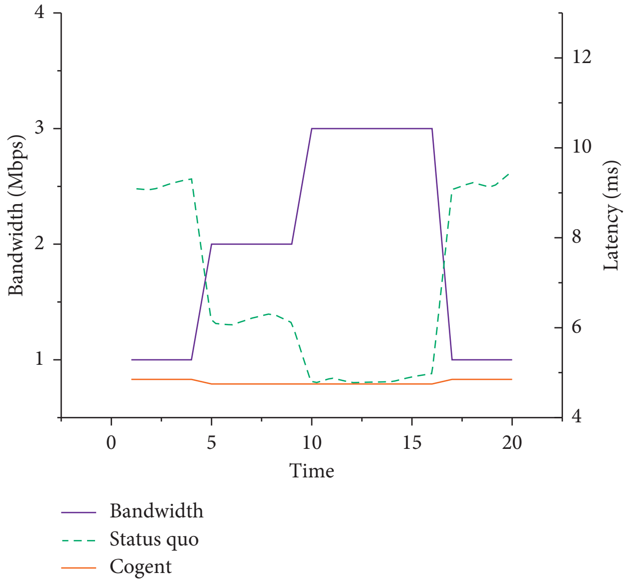


Figure 4: The effect of bandwidth variation on end-to-end latency.

4.4. Impact of Server Load Variation. In this section, we evaluate how Cogent makes dynamic decisions to the variation in the edge server load, so we assume that the network bandwidth is sufficient. Servers typically have high and low traffic queries, and a high server load will increase in the service time for DNN queries. Cogent makes the best decision to meet the current load state by periodically sending query information to the edge server to get the server's occupancy status. Figure 5 shows the end-to-end inference latency of VGG19, which is implemented by the status quo method and Cogent as the edge server load increases. The status quo method does not dynamically adapt to different server loads and therefore suffers significant performance degradation as the server load increases. On the other hand, by considering the server load, Cogent dynamically selects the partition point and pruning ratio to adapt well to the variation. In Figure 5, two vertical dashed lines indicate that Cogent has changed its computing strategy: from completely edge server execution at low load to partitioning the DNN between the mobile device and edge server at medium load and eventually completely local execution on the mobile device when the load is above 80%. Cogent keeps the end-to-end latency of performing image classification below 10 ms regardless of the server load. By considering the server load and its impact on server performance, Cogent always provides the best latency regardless of the variation in server load.

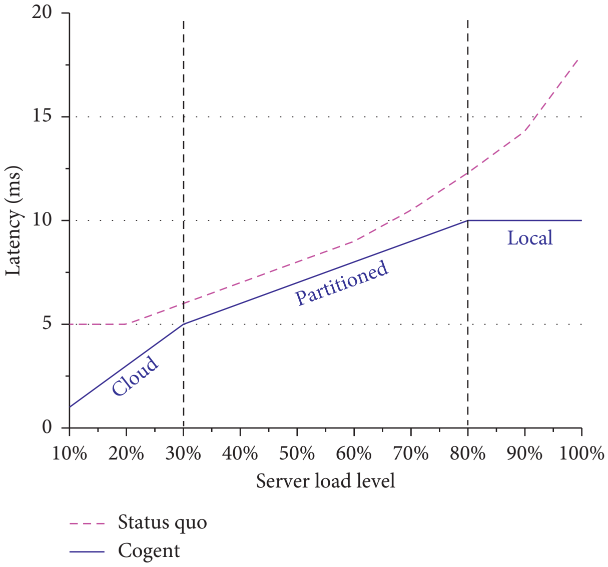


Figure 5: Cogent adjusts its partitioned execution as the result of varying edge server load.

Table 3: Inference latency of VGGNET on four hardware architectures under different strategies.

Inference latency on	HW1 (ms)	HW2 (ms)	HW3 (ms)	HW4 (ms)
VGGNET (baseline)	69.0641	69.1523	64.2927	16.2941
VGGNET (status)	24.8501	25.1915	19.9543	19.9781
VGGNET (cogent)	7.7703	12.1214	11.4689	3.7743

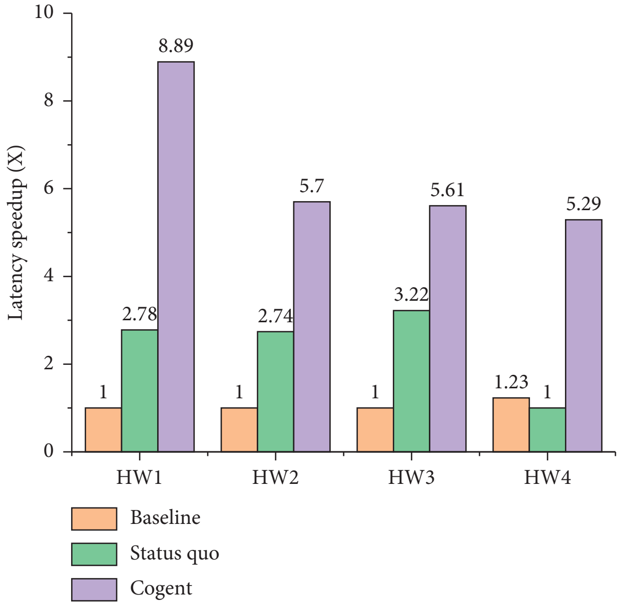


Figure 6: Comparison of latency speedup on four hardware architectures under different strategies.

4.5. Impact of Hardware Architecture. In this section, we evaluate the adaptability of Cogent to different hardware accelerators. Because the behavior of different hardware is very different, the performance of the model on the hardware is not always accurately reflected by the proxy signal. Therefore, receiving performance feedback directly from the hardware architecture is important to adapt to the operating environment of the model. The experiment set up four different hardware architectures: HW1: device accelerator1, HW2: device accelerator2, HW3: device accelerator3, and HW4: edge server accelerator. HW1 and HW4 are already described in Section 4.1, HW2 is a Raspberry Pi 3 with a quad-core 1.2 GHz ARM processor and 1 GB RAM, and HW3 is a mobile device with 1 NVIDIA Quadro K620 GPU. As can be seen from Table 3, a solution usually can only achieve optimal performance on hardware architecture. The Cogent framework we proposed can use reinforcement learning to automatically predict pruning and partition strategies based on given hardware feedback so that it can adapt to different hardware architectures. From the comparison results in Figure 6, it can be seen that Cogent running the same intelligent model can obtain better inference acceleration on different hardware architectures, reaching a maximum of 8.89×. Besides, the reason why the baseline method performs better than the status quo method on HW4 is that the baseline method has better adaptability to cloud-based hardware architectures.

5. Conclusion

In this paper, we propose a device-edge synergy intelligent acceleration framework Cogent based on reinforcement learning. The framework receives hardware configuration and system status feedback in a learning-based manner, uses

DDPG agents to automatically predict model pruning and partition strategies, and uses container technology to flexibly deploy partitioned model blocks. The Cogent framework includes two operational phases: automated pruning and partition phase and containerized deployment phase. Through the automatic pruning and partition stage of Cogent, the amount of computation and data transmission of intelligent model inference can be greatly reduced. Through the containerized deployment stage of Cogent, the flexibility and reliability of the system can be greatly improved. Our simulation results show that the Cogent acceleration framework has a significantly latency improvement compared to the completely cloud-based method and the representative partition synergy method when meeting user accuracy requirements. Besides, the Cogent framework also has better adaptability to network bandwidth, server load, and different hardware architectures. In future work, we hope that Cogent has user memory for the data cache and resource requirements of service requests. In the stage of pruning and partition of the model, Cogent can adjust the model according to the user's request habits to make the service more suitable for the user's personalization. User's personalization is a characteristic of the development of artificial intelligence services. The future service framework will only be recognized by users if it develops in a direction that better suits the needs of users.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] C. Szegedy, W. Liu, and Y. Jia, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Boston, MA, USA, June 2015.

[2] A. v. d. Oord, S. Dieleman, and H. Zen, "Wavenet: A generative model for raw audio," 2016, https://arxiv.org/abs/1609.03499.

[3] X. Chi, C. Yan, H. Wang, W. Rafique, and L. Qi, "Amplified LSH-based recommender systems with privacy protection," *Concurrency and Computation: Practice and Experience*, vol. 1, 2020.

[4] A. Almomani, M. Alauthman, F. Albalas, O. Dorgham, and A. Obeidat, "An online intrusion detection system to cloud computing based on NeuCube algorithms," in *Cognitive Analytics*, pp. 1042–1059, IGI Global, PA,USA, 2020.

[5] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[6] R. Hadidi, J. Cao, M. Woodward, M. Ryoo, and H. Kim, "Distributed perception by collaborative robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, p. 1, 2018.

[7] Y. Kang, J. Hauswald, C. Gao et al., "Neurosurgeon: collaborative intelligence between the cloud and mobile edge," in *ACM SIGARCH Computer Architecture News*, vol. 45, no. 1, pp. 615–629, 2017.

[8] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *Proceedings of the IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 328–339, Atlanta, GA, USA, June 2017.

[9] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6869–6898, 2017.

[10] F. Tung and G. Mori, "Deep neural network compression by in-parallel pruning-quantization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 3, pp. 568–579, 2020.

[11] T. Paine, H. Jin, J. Yang et al., "GPU asynchronous stochastic gradient descent to speed up neural network training," 2013, https://arxiv.org/abs/1312.6186.

[12] J. Yu, A. Lukefahr, D. Palframan, G. Dasika, R. Das, and S. Mahlke, "Scalpel," *ACM SIGARCH Computer Architecture News*, vol. 45, no. 2, pp. 548–560, 2017.

[13] A. G. Howard, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[14] H. Halawa, H. A. Abdelhafez, A. Boktor, and M. Ripeanu, "NVIDIA jetson platform characterization," in *Proceedings of the European Conference on Parallel Processing*, pp. 92–105, Santiago de Compostela, Spain, August 2017.

[15] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, Salt Lake UT, USA, June 2018.

[16] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Transactions on Network Science and Engineering*, vol. 42, no. 1 2020.

[17] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," 2015, https://arxiv.org/abs/1510.00149.

[18] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," 2015, https://arxiv.org/abs/1511.06530.

[19] N. D. Lane, S. Bhattacharya, A. Mathur, C. Forlivesi, and F. Kawsar, "DXTK: Enabling resource-efficient deep learning on mobile and embedded devices with the DeepX toolkit," in *Proceedings of the 8th EAI International Conference on Mobile Computing, Applications and Services*, pp. 98–107, Cambridge, UK, November 2016.

[20] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," 2016, https://arxiv.org/abs/1611.06440.

[21] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5058–5066, Venice, Italy, October 2017.

[22] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE*

International Conference on Computer Vision (ICCV), pp. 1389–1397, Venice, Italy, October 2017.

[23] H. Cai, T. Chen, W. Zhang, Y. Yu, and J. Wang, “Efficient architecture search by network transformation,” in *Proceedings of the National Conference on Artificial Intelligence*, pp. 2787–2794, New Orleans, LA, USA, Febraury 2018.

[24] C. Liu, “Progressive neural architecture search,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 19–34, Munich, Germany, September 2018.

[25] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, “Efficient neural architecture search via parameter sharing,” 2018, https://arxiv.org/abs/1802.03268.

[26] K. Ovtcharov, O. Ruwase, J.-Y. Kim, J. Fowers, K. Strauss, and E. S. Chung, “Accelerating deep convolutional neural networks using specialized hardware,” *Microsoft Research Whitepaper*, vol. 2, no. 11, pp. 1–4, 2015.

[27] S. Han, X. Liu, H. Mao et al., “Eie,” *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 243–254, 2016.

[28] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, “Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2019.

[29] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, “Joint optimization of offloading utility and privacy for edge computing enabled IoT,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2622–2629, 2020.

[30] W. Zhong, X. Yin, X. Zhang et al., “Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment,” *Computer Communications*, vol. 157, pp. 116–123, 2020.

[31] H. Liu, H. Kou, C. Yan, and L. Qi, “Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph,” *Complexity*, vol. 2020, pp. 1–15, Article ID 2085638, 2020.

[32] H. Liu, H. Kou, C. Yan, and L. Qi, “Link prediction in paper citation network to construct paper correlated graph,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019.

[33] L. Qi, Q. He, F. Chen et al., “Finding all you need: Web APIs recommendation in web of things through keywords search,” *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1063–1072, 2019.

[34] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. Sherman Shen, “Energy efficient dynamic offloading in mobile edge computing for internet of things,” *IEEE Transactions on Cloud Computing*, vol. 13, p. 1, 2019.

[35] J. Zhou, Y. Wang, K. Ota, and M. Dong, “AAIoT: Accelerating artificial intelligence in IoT systems,” *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 825–828, 2019.

[36] R. Hadidi, J. Cao, M. Woodward, M. S. Ryoo, and H. Kim, “Musical chair: Efficient real-time recognition using collaborative Iot devices,” 2018, https://arxiv.org/abs/1802.02138.

[37] X. Xu, S. Fu, W. Li, F. Dai, H. Gao, and V. Chang, “Multi-objective data placement for workflow management in cloud infrastructure using NSGA-II,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 9, pp. 1–11, 2019.

[38] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, “Community-diversified influence maximization in social networks,” *Information Systems*, vol. 92, pp. 1–12, Article ID 101522, 2020.

[39] E. Li, Z. Zhou, and X. Chen, “Edge intelligence,” in *Proceedings of the 2018 Workshop on Mobile Edge Communications*, pp. 31–36, Budapest, Hungary, August 2018.

[40] B. Zoph and Q. V. Le, “Neural architecture search with reinforcement learning,” 2016, https://arxiv.org/abs/1611.01578.

[41] H. Cai, J. Yang, W. Zhang, S. Han, and Y. Yu, “Path-level network transformation for efficient architecture search,” 2018, https://arxiv.org/abs/1806.02639.

[42] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, “AMC: Automl for model compression and acceleration on mobile devices,” in *Proceedings of the European Conference on Computer Vision*, pp. 784–800, Munich, Germany, September 2018.

[43] Z. Zhong, J. Yan, and C.-L. Liu, “Practical network blocks design with *Q*-learning,” 2017, https://arxiv.org/abs/1708.05552.

[44] B. J. A. Kröse, “Learning from delayed rewards,” *Robotics and Autonomous Systems*, vol. 15, no. 4, pp. 233–235, 1995.

[45] W. Farhat, H. Faiedh, C. Souani, and K. Besbes, “Real-time hardware/software co-design of a traffic sign recognition system using Zynq FPGA,” in *Proceedings of the 11th International Design & Test Symposium (IDT)*, Hammamet,Tunisia, December 2016.

[46] X. V. Ultra, “UltraScale architecture and product data sheet: Overview,” vol. 2018, p. 28, 2018.

[47] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, “Blockchain empowered arbitrable data auditing scheme for network storage as a service,” *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 289–300, 2020.

[48] Y. Abouelnaga, O. S. Ali, H. Rady, and M. Moustafa, “Cifar-10: knn-based ensemble of classifiers,” in *Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 1192–1195, Las Vegas, NV, USA, December 2016.

[49] H. Wu, Q. Wang, and K. Wolter, “Methods of cloud-path selection for offloading in mobile cloud computing systems,” in *Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science*, pp. 443–448, Taipei, Taiwan, December 2012.

[50] D. Liu, X. Chen, Z. Zhou, and Q. Ling, “HierTrain: Fast hierarchical edge AI learning with hybrid parallelism in mobile-edge-cloud computing,” *IEEE Open Journal of the Communications Society*, vol. 1, pp. 634–645, 2020.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8849181, 15 pages
https://doi.org/10.1155/2020/8849181

Review Article

A Review of Techniques and Methods for IoT Applications in Collaborative Cloud-Fog Environment

Jielin Jiang,[1,2] Zheng Li,[1,2] Yuan Tian,[3] and Najla Al-Nabhan[4]

[1] *School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China*
[2] *Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing, China*
[3] *Nanjing Institute of Technology, Nanjing, China*
[4] *Department of Computer Science, King Saud University, Riyadh, Saudi Arabia*

Correspondence should be addressed to Zheng Li; lizheng@nuist.edu.cn

Received 5 August 2020; Revised 17 August 2020; Accepted 28 August 2020; Published 7 September 2020

Academic Editor: Dou Wanchun

Cloud computing is widely used for its powerful and accessible computing and storage capacity. However, with the development trend of Internet of Things (IoTs), the distance between cloud and terminal devices can no longer meet the new requirements of low latency and real-time interaction of IoTs. Fog has been proposed as a complement to the cloud which moves servers to the edge of the network, making it possible to process service requests of terminal devices locally. Despite the fact that fog computing solves many obstacles for the development of IoT, there are still many problems to be solved for its immature technology. In this paper, the concepts and characteristics of cloud and fog computing are introduced, followed by the comparison and collaboration between them. We summarize main challenges IoT faces in new application requirements (e.g., low latency, network bandwidth constraints, resource constraints of devices, stability of service, and security) and analyze fog-based solutions. The remaining challenges and research directions of fog after integrating into IoT system are discussed. In addition, the key role that fog computing based on 5G may play in the field of intelligent driving and tactile robots is prospected.

1. Introduction

Over the years, with the rapid development of distributed computing, parallel computing, grid computing, network storage, and virtual machine technique, computing resources have become more abundant, cheaper, and more accessible than ever before. The development of the Information Technology (IT) industry and the influx of electronic devices into the market have increased the demand for computing and storage resources. In this context, a new computing mode called cloud computing was proposed. In this mode, resources (such as networks, computing, storage, and applications) are provided to users to access on demand at any time. Service providers are divided into infrastructure providers that manage cloud platforms and lease resources based on pricing models and service providers that rent resources from infrastructure providers to provide services to users. Because of the maturity of cloud computing technology and its advantages such as low cost, easy access to information, rapid deployment, data backup, and automatic software integration [1, 2], cloud has been widely used.

However, in the trend of Internet of everything, the application demand of low latency and high interactivity makes the remote connection between cloud and user devices become the key factor restricting the development. At the same time, the number and types of IoT devices (such as smart headsets, mobile computers, smart home appliances, on-board networking systems, smart traffic control lights, and more connected utilities) are rapidly increasing [3]. Large-scale data transmission poses great challenges to the performance of user devices and the existing network bandwidth. In addition, the security and privacy of personal and enterprise data are questioned, because data are stored centrally in cloud servers far away from users and they cannot determine whether their data is stolen by malicious actors with interest.

Fog computing is a new computing paradigm proposed to solve these challenges. Different from centralized servers in cloud, servers in fog are moved to the edge of the network, known as fog nodes. Some delay-sensitive tasks can be processed on these nodes [4], while some computation-intensive or delay-tolerant tasks are still processed in the cloud. Therefore, the user task request will not be sent directly to the remote cloud. Instead, it is received and processed by the neighboring fog nodes [5], which requires lower network bandwidth and user equipment performance than the former. In addition, fog computing also has some other advantages like stable service, high security, and privacy.

However, the practical technology of fog computing is not mature, and some problems that fog computing faces after being integrated into IoT system still need to be solved, such as heterogeneity, mobility, data and equipment management, QoS management, security, and privacy.

The article can be described as follows. Based on the introduction of the concepts and characteristics of cloud and fog computing in Section 2, the comparison and collaboration between them are presented in Section 3, while the methods and techniques of task offloading among cloud and fog are emphatically reviewed and analyzed. Challenges that IoT face (e.g., low latency, network bandwidth constraints, resource constraints of devices, stability of service, and security) are discussed in Section 4, which are linked to surveys about fog's contribution to addressing these challenges and some specific solutions that have been proposed. Particularly, methods and technologies to reduce delays and protect security and privacy are scrutinized. In Section 5, the remaining challenges and research directions of fog after integrating into IoT system are discussed. In Section 6, the key role of 5G-based fog computing in the field of intelligent driving and tactile robots is prospected.

2. Basic Definition and Characteristic

2.1. Cloud Computing

2.1.1. Definition of Cloud Computing. Cloud computing was proposed to support ubiquitous, convenient, and on-demand network access to configurable computing resources like storage space, servers, networks, applications, and services in a shared pool. These resources can be provisioned and released rapidly with minimal service provider interaction or management effort [6]. Figure 1 shows a structure of cloud computing.

(1) Up-front investment in cloud computing is not needed because it is a pay-as-you-go pricing model that allows service providers to benefit from the cost of renting resources in the cloud without investing in infrastructure.

(2) The leasing and management of resources are flexible. Resources in the cloud can be quickly released according to the requirements of users. When service demand is low, service providers can actively release idle resources in the cloud, reducing the pressure of center load and energy consumption, and reduce costs.

(3) Data resources received by cloud data center can be collected and analyzed by infrastructure providers. Service providers access the data analysis to discover the potential business trends and determine the growth demand for services, which is the basis for them to expand their service direction and scale.

(4) Services in the cloud are easily accessible via the Internet by devices such as mobile phones, computers, and PDAs.

2.1.2. Characteristics of Cloud Computing. Cloud computing is widely used for the following characteristics [8].

2.2. Fog Computing

2.2.1. Definition of Fog Computing. Fog computing has been proposed as a layered model to support convenient access to a shared continuum of extensible computing resources. This model, consisting of physical or virtual fog nodes which are context-aware, facilitates the deployment of delay-aware and distributed applications and services. Fog nodes are located between intelligent terminal equipment and cloud services, which support common communication system and data management. The organizational form of fog nodes in the cluster is based on the specific working mode. Association and separation are supported by horizontal and vertical distribution, respectively, and can also be supported by the delay distance from the terminal devices to the fog nodes. Fog computing provides network connections to centralized services and local computing resources for terminal devices with minimizing request response time [9]. Figure 2 shows a fog-cloud system with three-layer architecture: cloud layer, fog layer, and IoT/end-users layer. The fog layer can be composed of one or more fog domains which are controlled by the same or different service providers. Each fog domain consists of the nodes including gateways, edge routers, switches, PCs, set-top boxes, and smart phones. The IoT/end-users layer is formed by two domains which include end-user devices and IoT devices, respectively [10].

2.2.2. Characteristics of Fog Computing. As a supplement to cloud computing, fog computing has several distinct characteristics from cloud computing. Cloud computing is based on social public cloud and IT operator services, while fog computing is based on small clouds such as enterprise, private, and personal clouds. Cloud computing typically consists of clusters of computing devices with high performance, while fog computing consists of more decentralized computers, each with its own function [9]. In addition, the following characteristics of fog computing are also essential.

(1) *Location Awareness and Low Latency.* Fog nodes are located between terminal devices and the cloud,

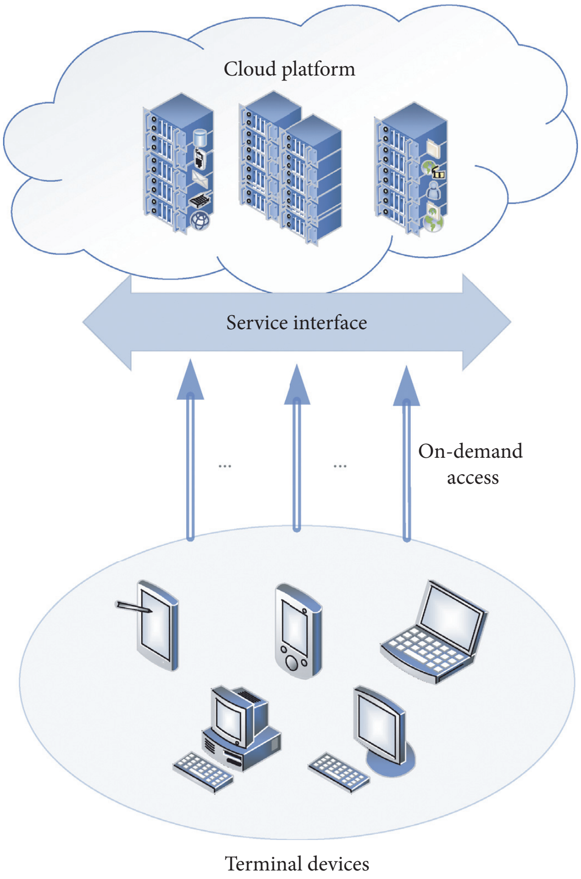


Figure 1: The structure of cloud computing [7].

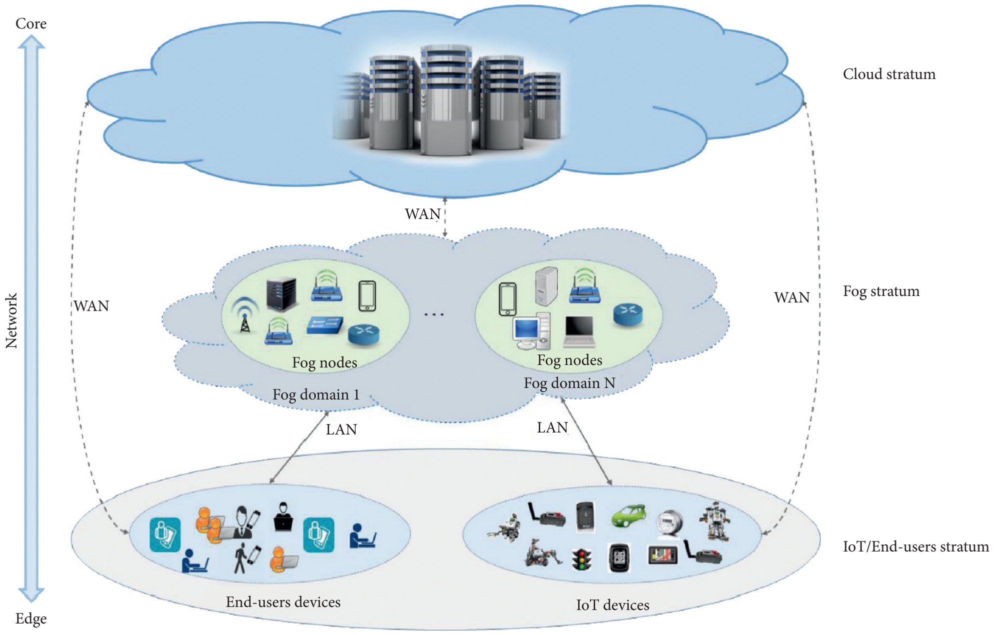


Figure 2: The structure of fog-cloud system [10].

tightly coupled to network and terminal devices, providing computing resources for them. Since the logical location of the fog node in the system and the cost of communication delays with other nodes are available, when the service requirements and data generated by the terminal device are sent to the network, the nearest available fog node will receive and process these requests and data [11]. Because fog nodes usually coexist with devices, the latency between them is much lower than the latency in cloud system [9].

(2) *Geographical Distribution*. Fog nodes provide some form of communication and data management service between the centralized cloud center and the edge of network where the terminal devices reside [3]. For example, high-quality streaming media services are provided to mobile vehicles in fog system by locating fog nodes on tracks and highways, which requires the extensive deployment of applications and target services which can identify location in fog. In order to deploy the capability to fog, the operation of geographically concentrated or dispersed fog nodes is adopted. Such geographical distribution makes fog system achieve good results in the service based on geographical location [12, 13].

(3) *Agility*. Fog nodes are distributed at the edge of the network, directly interacting with user terminals. The amount of data, network environment, and resource conditions that fog faces change constantly. Fog computing is adaptive in nature, supporting data load changes, flexible computing, resource pool, and network conditions changes at the cluster level and listing some of the adaptive features that are supported [9, 14].

(4) *Heterogeneity*. A large amount of heterogeneous data with different formats and storage forms is generated in terminal devices at the edge of the network [9]. The ability to collect, aggregate, and process these heterogeneous data is critical as the fog node acts as a base station to provide processing and storage services.

(5) *Interoperability and Federation*. Fog computing extends the powerful computing resources of the cloud to the network edge. Although the service requests of user devices can be quickly responded, the computing and storage capacity of each node is far less powerful than that of a centralized cloud center. Services requiring dense computation provided by fog to user devices may require joint support from multiple fog nodes. Therefore, interoperability of fog components and cross-domain cooperation among nodes are supported in the fog [10].

(6) *Real-Time Interactions*. The interaction between user devices and fog nodes is in real time with no long wait and transmission delays. The data sent by the user devices is received and processed by the adjacent fog node immediately. When the processing is completed, the processing results will be timely returned back to the user devices, which allows fog to support time-sensitive applications [9].

3. Cloud and Fog Computing

3.1. Comparison of Cloud and Fog Computing. As an extension of cloud computing, fog computing has many similarities and differences with cloud computing. The comparison between cloud and fog computing in this paper is made in the following aspects, which can intuitively reflect similarities and differences between them [7].

(1) *Reaction Time and Latency*. In fog system, time-sensitive data is sent to the fog node closest to data source for processing and analysis, rather than being sent directly to the distant cloud center, significantly reducing the service response time. Computation-intensive or delay-tolerant tasks can be processed in the dense area of fog nodes, which may take a few seconds. But the time to interact with the cloud can be a few minutes, minutes, or even hours [15, 16]. Thus, faster responses and more flexible choices are provided by fog computing than by cloud computing.

(2) *Node Location Distribution*. Far away from user devices, the cloud center provides users with resource-intensive computing and storage services in the form of central servers, while the fog is close to the edge of the network in the form of scattered fog nodes. Each fog node can be either individual computing devices or servers with strong capabilities [15].

(3) *Service Scope and Location Perception*. The service scope of cloud computing covers the whole world, but the cloud center is far away from the user terminal and cannot accurately perceive the location of the service. Unlike cloud computing, there are numbers of fog nodes in fog system with the capability of service location awareness [17–19]. Fog domains formed by multiple fog nodes serve user devices in local areas like city blocks. The service scope is usually determined by the density and computing power of fog nodes [15].

(4) *Vulnerability and Security*. In the cloud system, the user data is stored in the cloud computing center, far away from users and more likely to be attacked centrally. Although Xu et al. [20] proposed a fault-tolerant resource allocation method for data-intensive workflow to solve the recovery problem of failed tasks caused by the failure of a certain computing link, the possibility of systematic collapse caused by faster workflow aggregation still exists. Fog nodes are geographically dispersed and close to users, allowing users to protect the security and privacy of their data [1, 21].

3.2. Collaboration between Cloud and Fog Computing. Cloud center is far away from user terminals, and there are some problems in the application of the emerging IoTs (e.g., delay-sensitive applications). When fog extends the cloud to terminal devices on the edge of the network, these new service demands are met. So how can data sent by IoT devices be processed and analyzed in the collaboration of cloud and fog? This problem is discussed in the next two subsections.

3.2.1. Flow of Task Processing. Figure 3 shows an example of a cloud-fog-IoT interaction model. What the fog and the cloud need to do, respectively, in the flow of task processing are introduced as follows.

(1) *The Things Fog Nodes Need to Do.* At the network edge, fog nodes receive real-time data from the terminal devices which are usually heterogeneous and dynamic. Then the analysis and real-time control of the data are carried out by running the applications supported by the IoT to achieve response within milliseconds [22]. Temporary data storage is also provided by fog nodes which usually lasts about 1 to 2 hours. After the data has been processed and analyzed, data summaries are sent to the cloud center regularly [9].

(2) *The Things Cloud Needs to Do.* Data digests sent by fog nodes are collected and integrated in the cloud platform. Then, an overall analysis and evaluation of these data are made to obtain service growth trends which are helpful for service providers to determine the direction of business development. Finally, new application rules based on the results of business evaluation are formulated by platform which are used to achieve the goal of adjusting service balance [9].

3.2.2. Task Offloading among Fog and Cloud. Because of the limited resources, the tasks of user devices that cannot be processed locally are sent to the fog nodes. Assuming a situation where the data and service demands generated by the edge devices are only processed in the fog server, when the situation of dense service demands occurs, the computing delay will exceed the allowable range of the requester for the limited resource of fog nodes. In general, two strategies are adopted: (1) Offload tasks to cloud and process them in cloud with rich resources and powerful computing capacity. After the task completed, the required results will be returned. (2) Offload tasks to adjacent fog nodes and complete the user's service demands through distributed collaboration among the selected fog nodes. Figure 4 shows a simple task offloading model that includes both fog-to-fog and fog-to-cloud task offloading. Many scholars have investigated the two offloading strategies. They have been committed to building efficient architectures or developing smart strategies to decide whether to offload tasks to the cloud or fog nodes. In the latter case, the optimal offloading destination should be found in surrounding nodes.

Sun et al. [23] proposed a generic IoT-fog-cloud architecture. The problems of task offloading, time efficiency calculation, and allocation were turned into the problem of time and energy cost minimization. To solve this problem, they proposed an ETCORA algorithm that significantly reduces energy consumption and request completion time.

Yang et al. [14] and Du et al. [24] proposed joint offloading methods based on the existing fog-cloud framework and took various factors affecting the offloading (such as link bandwidth, latency, and computing capacity of the offloading targets) as parameters to generate mixed-integer programming. The difference is that Yang [14] proposed a greedy algorithm that maximizes operator benefits on the premise of ensuring user performance and security requirements, while Du [24] proposed a low-complexity suboptimal algorithm. The latter obtains offloading decision through randomization and semidefinite relaxation and obtained resource allocation policy using Lagrangian dual decomposition and fractional programming theory, which optimizes the time delay and energy consumption effectively.

Chen et al. [25] proposed an energy-saving offloading method. In this method, energy-saving offloading problem is formulated as a random optimization problem and random optimization technique is used to determine it.

Xu et al. [26] proposed a computation offloading method using blockchain technology. The balanced allocation strategy is generated by nondominant sorting genetic algorithm, and then the optimal offloading strategy is determined by simple additive weighting and multicriterion decision. In this way, overloaded tasks in a node can be offloaded to the most suitable adjacent node for safe and prompt processing.

Chen et al. [27] also proposed an offloading strategy. They designed an efficient online algorithm that took into account the unnecessary consumption of idle servers, combined with the computation offloading and sleep decisions of node servers, to maximize server quality and reduce energy consumption.

Nassar and Yilmaz [28] formulated the resource allocation problem as a Markov decision process and finally solved it by learning the optimal decision strategy with some reinforcement learning methods, namely, SARSA, Expected SARSA, Q-learning, and Monte Carlo. Their work can make the fog node decide the processing location of service request according to its own resources, realizing the low latency transaction offloading and processing with high performance.

Chen et al. [29] developed an energy-efficient computing offload scheme that comprehensively considers the energy-consumption components at a fog node, including the energy consumption of transmission, local computing, and waiting state. They proposed an accelerated gradient algorithm to find optimal offloading point with a high speed.

Deng et al. [30] investigated a load allocation problem that can be used to solve the optimal workload allocation between fog and cloud. They then decomposed the problem into three subproblems by using approximation method. By solving these subproblems separately, the optimal collaboration scheme with low latency and energy consumption is given. Ye et al. [31] proposed an extensible fog computing paradigm and developed a distribution strategy through genetic algorithm. This strategy enables the roadside nodes to offload computing tasks with the minimum cost within the allowable delay.

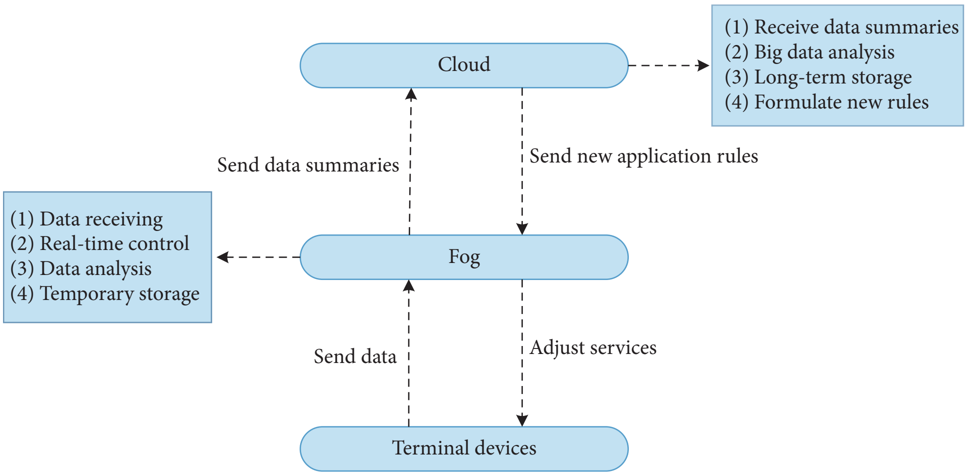


Figure 3: The interaction model of cloud-fog-IoT [7].

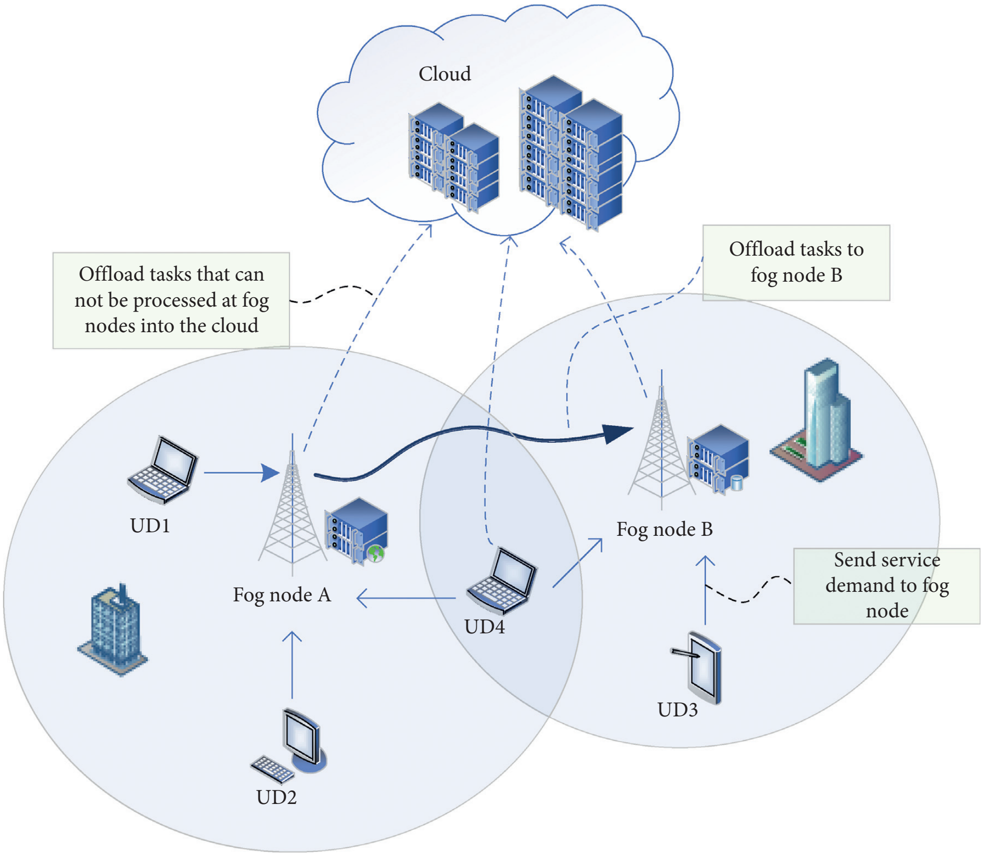

Cloud
Offload tasks that can not be processed at fog nodes into the cloud
Offload tasks to fog node B
UD1
Fog node A
Fog node B
UD4
Send service demand to fog node
UD3
UD2

Figure 4: The task offloading scheme among fog and cloud.

Table 1 is a summary of reviewed techniques or methods about task offloading among fog and cloud which contains each of the solutions and its unique advantages.

4. Fog Computing Helps on the New Challenges of IoT

Moving computing, storage, and analysis tasks to the cloud center with powerful resource has been a major solution to meet service needs over the past decades. However, the emergence of many delay-sensitive applications in the IoT has created many new application requirements, such as low latency and real-time interaction which the cloud cannot meet. The fog computing, expanding the cloud to network edge, is a better solution proposed to solve these problems. The challenges of the cloud-based IoTs and fog-based solutions with specific techniques are discussed in this section [7].

Table 1: Work summary of task offloading among fog and cloud.

Reference	Solution	Advantages
[23]	A generic IoT-fog-cloud architecture	Reduces energy consumption and request completion time
[14, 24]	A joint offloading method based on the existing fog-cloud framework	Takes various factors affecting the offloading as parameters to generate mixed-integer programming
[25]	An energy-saving offloading method	Uses random optimization technique to solve energy-saving offloading problem
[26]	A computation offloading method using blockchain	Offloads overloaded tasks to the suitable adjacent node with the optimal offloading strategy determined by simple additive weighting and multicriterion decision
[27]	An efficient online algorithm considering the unnecessary consumption of idle servers	Combines the computation offloading and sleep decisions of node servers to maximize server quality and reduces energy consumption
[28]	An optimal decision strategy with some reinforcement learning methods	Makes the fog node decide the processing location itself and realizes low-latency offloading and high-quality processing
[29]	An energy-efficient computing offload scheme	Proposes an accelerated gradient algorithm to find optimal offloading point with a high speed
[30]	An optimal collaboration scheme with low latency and energy consumption	Decomposes the load allocation problem into three subproblems by using approximation method
[31]	An extensible fog computing paradigm	Develops a distribution strategy based on genetic algorithm to enable the roadside nodes to offload computing tasks with the minimum cost

4.1. Low Latency Requirements. Nowadays, many life applications and industrial systems require end-to-end communication with low latency, such as smart home applications and virtual reality applications, especially those requiring ultralow latency and affecting personal safety like driverless car [32]. In order to minimize the latency in data transmission, fog nodes are deployed on the edge of network, allowing data to be processed, analyzed, and stored near end-users.

Due to the decentralized distribution of IoT devices, computing and service load distribution between fog nodes affect the computing and communication delay of data flows, respectively. To solve this problem, Fan and Ansari [33] proposed a workload-balancing scheme that associates the appropriate fog nodes with user devices, minimizing the data flow latency for data communication and task processing.

Xu et al. [34] introduced 5G into Internet of connected vehicles (IoCV) to improve the transmission rate of roadside equipment. They innovatively designed an adaptive computation offloading method for the prospective 5G-driven IoCV in which multiobjective evolutionary algorithm is used to generate the available solutions. The optimal solution obtained by utility evaluation effectively optimizes the task response time and resource utilization efficiency.

Li et al. [35] proposed a delay estimation framework for IoT based on fog, which can accurately predict the end-to-end delay in cloud-fog-things continuum. Mohammed et al. [36] proposed a data placement strategy for fog architecture. They solved the problem of data layout with generalized assignment and developed two solutions. These solutions reduce latency by about 86% and 60%, respectively, compared with solutions based on cloud.

Naranjo et al. [37] proposed a smart city network architecture based on fog. In order to manage the application under the premise of satisfying QoS, the communication between devices in the architecture is divided into three categories. With this approach, the nodes in the architecture can run in high efficiency and low latency.

Craciunescu et al. [38] and Cao et al. [39] proposed algorithms for medical systems in order to detect individual falls in time. Gu et al. [40] minimized communication time by optimizing resource utilization in the healthcare system.

Dragi et al. [41] proposed a new nature-inspired smart fog architecture. This architecture is a distributed intelligent system modeled using new techniques in the fields of graph theory, multicriteria decision-making, and machine learning. It can provide adaptive resource management and low decision latency by simulating the function of the human brain.

Yousefpour et al. [42] and Elbamby et al. [43] proposed task offloading strategies to reduce service latency. In paper [42], fog-to-fog communication was employed to share workload, while a clustering method was designed in [43], which groups user devices and their service nodes with common task interests and uses a matching game, where computing delay is minimized under delay and reliability constraints.

Diro et al. [44] proposed an aggregated software defined network (SDN) and fog/IoT architecture, which allocates different flow spaces for heterogeneous IoT applications according to flow categories to meet the priority-based QoS requirements. This architecture reduces the impact of packet blocking on QoS delivery through more fine-grained control.

Rahbari et al. [45] proposed a greedy scheduling algorithm based on knapsack, which allocates resource to nodes in fog considering various network parameters. Through simulation experiments, they proved that the proposed algorithm has better performance in the optimization of time delay and energy consumption.

Shi et al. [46] set up a proof-of-concept platform. They tested the face recognition application and reduced the response time from 900 milliseconds to 169 milliseconds by offloading the computing tasks from cloud to the edge. Fog also supports time-sensitive control functions in local physical systems [3].

Table 2 is a summary of techniques or methods about reducing latency based on fog for IoT applications, containing reviewed solutions and their unique advantages.

4.2. Network Bandwidth Constraints. As the number of devices connected to the network is increasing rapidly, the speed of data generation is increasing exponentially [47]. For example, a connected car can generate tens of megabytes of data per second including vehicle status (e.g., wear of vehicle components), vehicle mobility (e.g., driving speeds and routes), vehicle surroundings (e.g., weather conditions and road conditions), and videos recorded by automobile data recorder. A driverless car can generate much more data [48]. The American Smart Grid generates 1,000 gigabytes of data per year. Google trades 1 gigabyte a month and the Library of Congress generates about 2.4 gigabytes of data each month [49]. If all data is transmitted to cloud, ultrahigh-quality network bandwidth is required, which poses a heavy burden on the existing network bandwidth and even leads to congestion, obviously not advisable.

Fog nodes receive and process the data near user devices, filtering out irrelevant or inappropriate data to prevent them from traveling across the whole network [1]. Data generated by user devices is allocated to the nearest fog node for processing instead of being transmitted to cloud center, because much critical analysis does not require powerful computing and storage capabilities of the cloud. ABI Research estimated that 90% of the data generated by endpoints will be stored and processed locally, not in the cloud [47]. The way fog processes data significantly reduces the amount of data sent to the cloud, easing the burden of network bandwidth.

4.3. Resource Constraints of Devices. In IoT system, user devices with limited resources (e.g., computing, network, and storage resources) cannot interact with the cloud directly, for which sending data to the cloud is impractical [50]. It is also unrealistic to update resources for these devices at high cost.

In this case, the functions of cloud cannot be performed well, while fog nodes can handle resource-intensive tasks for these devices without requirement of high performance [51]. Fog nodes are core components in the fog computing architecture, which are either physical components (such as servers, routers, gateways, and switches) or virtual components (such as virtual machines and virtual switches). Fog nodes tightly couple with access networks or intelligent devices and provide computing resource for these devices [9]. Therefore, the complexity and resource requirements of terminal devices are reduced.

4.4. Stability of Service. When a stable connection between user device and cloud is not guaranteed, continuous service cannot be obtained from cloud. For example, when a car enters an area not covered by a stable network, the cloud service is intermittently disconnected. Some necessary services are unavailable to the on-board devices and other user devices [3].

But, unlike cloud, fog nodes are distributed geographically. Edge networks created by fog computing are located at different points to extend the isolated infrastructure in cloud. A local system formed by fog nodes which can operate autonomously, with continuous coverage of the service scope, helps to process service demands more quickly and steadily [1]. Location-based mobility requirements and diversified service are supported by the administrators of fog nodes [18, 40, 52]. Due to the decentralized distribution of IoT devices, computing and service load distribution between fog nodes affect the computing delay and stability of service, respectively. To solve this problem, Fan Qiang and Ansari Nirwan [33] proposed a workload-balancing scheme that associates the appropriate fog nodes with user devices, which minimizes the data flow latency of communication and significantly improves the stability of service.

Yousefpour et al. [53] proposed a dynamic service-provisioning framework based on QoS perception for dynamically deploying application services on fog nodes. Then a possible formula and two efficient greedy algorithms were given to address the service provision, which can provide stable and continuous service with low latency.

4.5. Security and Privacy. With the purpose of requesting a service, large amounts of data are sent to the network, including personal privacy and corporate data. For example, work logs generated by smart home appliances can be mined to reveal the work and rest rules of users, and important private information (e.g., password and possessions) can be eavesdropped from chat logs. Therefore, both the static data and the transmission process in IoT need to be protected, which requires the monitoring and automatic response of malicious attacks in the whole process [13].

In cloud computing, corporate and private data, and even confidential data, are stored centrally in cloud servers far away from users. The security and privacy of personal and enterprise data are questioned by users because they cannot determine whether their data is stolen by malicious actors with interest and whether their data will be lost in the expansion of the cloud center [1].

In fog computing, sensitive data is processed locally rather than being sent to the cloud. Local administrators can inspect and monitor the devices that collect, process, analyze, and store data [13, 51]. In the fog system, each fog node can act as a proxy for user devices which cannot adequately protect data due to resource constraints, helping update and manage the security credentials of user devices, to compensate for their security vacancies.

Abbas et al. [54] proposed an innovative fog security service to transfer confidential and sensitive data generated by IoT devices to fog nodes for processing and provide end-to-end security between them by using two mature encryption schemes, identity-based signature and identity-based encryption. Local information also can be used to

Table 2: Work summary of reducing latency based on fog for IoT applications.

Reference	Solution	Advantages
[33]	A workload-balancing scheme associating the appropriate base stations with user devices	Minimizes the data flow latency for data communication and task processing
[34]	An adaptive computation offloading method for the prospective 5G-driven IoCV	Optimizes the task response time and resource utilization efficiency with the optimal solution obtained by utility evaluation
[35]	A delay estimation framework for IoT based on fog	Accurately predicts the end-to-end delay in cloud-fog-things continuum
[36]	A data placement strategy for fog architecture	Solves data layout problem with generalized assignment problem and develops two solutions
[37]	A smart city network architecture based on fog	Divides the communication between devices into three categories to satisfy QoS
[38, 39]	Detection algorithms and fog-based medical information systems	Detects individual falls in time
[40]	A medical cyber-physical system supported by fog computing and a heuristic algorithm with two phases	Minimizes communication time by optimizing resource utilization
[41]	A new nature-inspired smart fog architecture	Provides adaptive resource management and low decision latency by simulating the function of the human brain
[42]	A task offloading strategy to reduce service latency	Employs fog-to-fog communication and share workload
[43]	A clustering method for offloading	Groups user devices and nodes and uses a matching game to minimize computing delay
[44]	An aggregated software defined network and a fog/IoT architecture	Reduces the impact of packet blocking on QoS delivery through more fine-grained control
[45]	A greedy scheduling algorithm based on knapsack	Allocates resource to fog nodes considering various network parameters, optimizing time delay and energy consumption

monitor the security status of nearby devices and detect threats immediately to ensure security [3, 55].

In recent years, some malicious code detection techniques have been proposed to solve the problem of security detection in fog environment. Zhang et al. [56] used signature-based detection technique and Martignoni et al. [57] proposed a behavior-based detection technique. However, behavior-based detection has a high cost because of resource constraints of fog nodes. Signature-based detection technology is more effective, but it is still difficult to detect variable malicious code in distributed fog nodes. In this case, a hybrid detection technology combining the two technologies was proposed to solve this problem [58]. The behavior-based detection technology in the cloud is distributed to fog nodes, and suspicious software files are detected and sent to the cloud for analysis. If the malware is new, the analysis result will be saved to the database as a new signature and the malicious signature list of each node will be updated.

Xu et al. [59–61] improved the strength Pareto evolutionary algorithm to obtain offloading schemes. The scheme proposed in paper [59] is privacy-aware, which effectively protects the privacy of training tasks offloaded to fog nodes with maintaining overall network performance, while schemes in paper [60, 61] are trust-aware. After the balanced scheme is obtained, they used the multicriterion decision technique and similarity prioritization technique of ideal solution to determine the optimal solution, which can effectively protect privacy with minimized service latency.

Thota et al. [62] proposed efficient centralized security architecture based on fog environment. Patient's medical data is transmitted seamlessly from sensors to edge devices and finally to the cloud for medical staff to access. The architecture effectively protects the privacy of patients and the security of medical data.

Chi et al. [17] proposed a service recommendation method based on amplified location-sensitive hash in order to ensure privacy security of distributed quality data from multiple platforms during cross-platform communication and proved its feasibility through experiments.

Viejo et al. [63] used new fog choreography concepts to solve the problem of reduction of service response time caused by resource constraints of the IoT. The security and efficient delivery of the service were realized successfully, which provide effective support for expensive encryption technology.

Mukherjee et al. [64] firstly designed and implemented a middleware featuring end-to-end security for cloud-fog communications. The intermittence and flexibility of middleware were proposed, respectively, by dealing with unreliable network connection and customizing the security configuration required by the application. The middleware can provide fast, light-weight, and resource-aware security for a wide variety of IoT applications.

Li et al. [65] proposed a hierarchical data aggregation scheme for efficient privacy protection. By modifying Paillier encryption, this scheme can not only resist external attacks but also prevent personal privacy data from leaking from internal devices.

Daoud et al. [66] designed a security model based on fog-IoT network. Then a comprehensive scheduling process and resource allocation mechanism were proposed based on the model. Through these efforts, they successfully introduced the active security scheme with low latency and ultra-trustworthiness into fog-IoT network.

Table 3: Work summary of security and privacy based on fog for IoT applications.

Reference	Solution	Advantages
[54]	An innovative fog security service	Uses identity-based signature and encryption to effectively protect the sensitive data transmitted to the fog node
[56, 57]	Two detection techniques	Detects malicious code attacks stably
[58]	A hybrid detection technology	Extends cloud-based detection technology to fog nodes
[59–61]	Offloading schemes driven by the improved strength Pareto evolutionary algorithm	Uses the multicriterion decision technique and similarity prioritization technique to protect privacy with minimized service delay
[62]	An efficient centralized security architecture based on fog environment	Protects the privacy of patients and the security of medical data through the device-edge-cloud transmission route
[17]	A service recommendation method based on amplified location-sensitive hash	Ensures privacy security of distributed quality data during cross-platform communication from multiple platforms
[63]	A new fog choreography concept	Realizes the security and efficient delivery of the service
[64]	A middleware featuring end-to-end security for cloud-fog communications	Uses middleware to provide fast, light-weight, and resource-aware security for a wide variety of IoT applications
[65]	A hierarchical data aggregation scheme for efficient privacy protection	Prevents personal privacy leaks while resisting external attacks
[66]	A security model based on fog-IoT network	Introduces the active security scheme with low latency and ultratrustworthiness into fog-IoT network

Table 3 is a summary of reviewed fog-based techniques or methods to protect security and latency in IoT applications, containing each solution and its unique advantages.

5. Challenges and Research Directions

5.1. Heterogeneity. In the fog, a large number of heterogeneous platforms and devices are connected to the Internet, and the services or resource requests they send are often heterogeneous, which requires all nodes in the network to dynamically identify all the request information. In fact, there has been some work to classify nodes using labels [67] or to add descriptions to resources to provide solutions for heterogeneous requests. Most of these solutions are based on static recognition and their work depends on the developers of the architecture program [68, 69], lacking flexibility and generality. In addition, data or service requests sent by IoT devices may be supported by multiple service providers, each using a mostly inconsistent service description model. More complex heterogeneous data and models will be produced for the addition of new providers, which brings about more burden to programmers.

There have been also many algorithms proposed in the fog environment to compute the capabilities of fog nodes, some of which consider the resource constraints of different devices and model the differences of capabilities between nodes, while most of which cannot meet the heterogeneous criteria [31, 70, 71]. Even so, these works are based on the different understanding of heterogeneity of nodes in network.

Therefore, semantic specifications should be defined in a clear form within the fog domain so that IoT devices or clouds connected to the fog network can share information in a commonly understood manner, which will contribute to the homogeneity of heterogeneous data at the edges and the simplification of data transfer protocols [10].

5.2. Mobility. Mobility presents significant challenges for both IoT devices and fog nodes. A large number of IoT devices are now wirelessly connected to the network and are generally mobile. The coverage of each fog domain is limited, which makes it necessary to consider service migration when the connected IoT device falls out of service scope.

How to migrate the service data from the previous fog domain to the new fog domain without interrupting the service is a challenging task. In the simplest case, IoT devices on a vehicle may move between different fog domains, making the service provided to on-board devices unstable. To solve this problem, Hassan et al. [72] recommended that service metadata be stored in the cloud so that it can be downloaded continuously after the device is migrated. But it will undoubtedly take some time to update the service information after the migration because of the difference of data between the cloud and the device. Another option being considered is to extend the existing fog architecture to support device mobility [73], which still needs to address the migration problem further. In fact, if the real-time moving trajectory of the device is obtained, machine learning technology can be used to analyze it and predict the future trajectory. Based on the predicted results, the next fog domain can update the service information of the device before it reaches the edge of the fog domain.

In addition, how to allocate the available resources/tasks that a fog node carries when it joins/leaves a fog domain is a complex issue. Song et al. [74] investigated this problem, hoping to realize dynamic load balancing in the fog region during node migration. Even so, how to achieve load balancing in a short time with affecting fewest nodes is still an optimal-solution problem to be explored.

5.3. Data and Equipment Management. Billions of devices (e.g., mobile phones, computers, palmtops, and smart appliances) are connected to the fog and the scale is growing. Different faults caused by various heterogeneous devices may occur anywhere. But tracking the fault information of hardware and providing software patches for maintenance in time are complex works, which need a sound fault detection and analysis mechanism and can locate the location of the fault in time. In addition, the Internet Data Center (IDC) estimated that the amount of data generated by IoT devices will reach 44 zettabytes in 2020 [75]. How to store such a huge amount of data in fog nodes with limited resources is another problem that must be solved.

Open fog suggests using machine learning technology to develop a framework with fault comprehensive feature detection and fault tolerance [76], especially in systems involving critical applications of life, such as anomaly detection in the medical field. There is also a data management scheme proposed in [67], which uses the labeling method to add labels for different types of data to facilitate access. This method can be used in the management of IoT devices. However, many management schemes do not take the resource availability into account, which has a great impact on the workload that devices can share in fog. Therefore, the expected research scheme should be based on the dynamic update of the connected devices, rather than simply assuming that the devices are fixed.

5.4. QoS Management. SLA management is an unavoidable direction to study QoS management. There are a number of cloud-based SLA management schemes that effectively reduce transmission latency, guarantee transmission bandwidth, and reduce packet loss rates (e.g., [77–82]) by reducing SLA violation rates or improving QoS.

In the fog, however, few SLA-managed schemes have been proposed, which are critical to maintaining desired QoS in the fog, where distributed services are dynamically provided. Among the papers reviewed in this article, only Yousefpour et al. [53] proposed a dynamic service-provisioning framework based on QoS perception for dynamically deploying application services on fog nodes, while it is only sensitive to the measure of delay, which is the same as many other strategies that meet QoS standards.

Although latency is certainly an important indicator of a system, there are many other important performance indicators to consider, such as bandwidth, resource utilization, and energy consumption. These indicators should be integrated as targets for future research strategies rather than as constraints alone. For example, Yang et al. [14] and Du et al. [24] proposed a joint offloading method that takes the link bandwidth and latency as parameters, but their method is only an optimization under time delay constraints rather than multiobjective optimization.

In fact, cloud-based QoS management technology has been relatively mature. The new SLA management solution could be an extension of the cloud-based SLA management technology with additional considerations for the uniqueness of fog [10] (such as resource constraints, low latency, and geographic distribution). Although the services are more diversified due to the large amount of heterogeneous data in fog, it is not difficult to solve the fog-based integrated QoS optimization if the semantic specification mentioned in Section 5.1 can be completed to solve the problem of heterogeneity.

5.5. Security and Privacy. As mentioned in Section 4.5, fog nodes act as agents to provide secure selection for resource-constrained devices and encrypt the transmission before the data leaves the edge [3]. But, to put it in another way, fog nodes are scattered on the edge of network where the environment is much worse than the cloud center [83, 84]. Because many fog nodes are in public places, the safety of physical equipment is difficult to guarantee. Moreover, lack of sufficient resources and computing power makes fog nodes unable to perform some complex security algorithms, so fog systems are more vulnerable to attack, such as session hijacking, session riding, and SQL injection [85].

The security and privacy challenges of the fog can be divided into four points: (1) A trust model and mutual authentication trust mechanism are necessary to guarantee the reliability and security of fog network [86]. (2) Traditional certificate and public key infrastructure (PKI) authentication mechanisms are difficult to be used by resource-constrained devices. [87]. (3) The messages sent by IoT devices cannot be encrypted symmetrically. In addition, asymmetrical encryption technology has great challenges, including resource and environment constraints, overhead constraints, and maintenance of the PKI [86]. (4) Location privacy in fog is vulnerable to leakage. While fog nodes are location-aware, collecting location information of IoT devices has become much easier than before [88]. In addition, frequent interaction among the three layers of fog architecture will increase the possibility of privacy disclosure. Without proper security measures, the performance of fog system may be seriously damaged.

6. Prospects

The rise of 5G technology promotes the application of fog computing technology. We expect fog computing technology based on 5G network to play a key role in the fields of intelligent driving and tactile robots.

Intelligent driving is an important technology to solve traffic congestion in the future, including automatic driving and human intervention. Fog nodes are distributed on the roadside, which can reduce the delay of data transmission between vehicles and nodes. But 4G network-based communication is far from being capable of transmitting the huge amount of data that cars generate while driving, and the delay is far from the requirement of automatic driving. 5G network is the key enabler to realize intelligent driving in fog system, which can provide ultrastable and ultrafast data transmission. When the vehicle is under automatic driving, roadside sensors and on-board equipment collect real-time road and vehicle information and send it to the roadside fog nodes for processing and analysis. Applying sophisticated machine learning techniques to fog nodes is necessary, which can learn to recognize all possible road conditions and send correct response instructions to driving system. Based on the same principle, the attitude detection of the driver during manual intervention can also help to improve the safety of driving.

Traditional robots are usually operated by command. It is difficult to break the technical bottleneck of remote control for transmission delay. But if the robot system architecture is deployed on the fog, the sensing equipment of the remote robot and the control equipment with tactile sensation are taken as the user terminal equipment, and the service nodes supporting the tactile command processing and analysis of the robot are brought into the fog layer. With the support of 5G ultrahigh transmission rate and tactile Internet [89, 90], real-time control of the robot can be realized. This work can be extended to the field of telemedicine and disaster relief with great research prospects. These works need to be based on completing the challenges of Section 5.

7. Conclusion

This article surveys the literature on cloud computing, fog computing, and IoT. Based on the concept and characterization description of cloud computing and fog computing, the comparison and collaboration between them are elaborated and some proposed task offloading methods are introduced emphatically. By surveying proposed techniques and methods, the contribution of fog computing to solving the challenges of IoT applications is introduced. Then the remaining challenges and research directions of fog after integrating into IoT system are discussed. In addition, the key role of 5G-based fog computing in the field of intelligent driving and tactile robots is prospected.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Authors' Contributions

Jielin Jiang, Zheng Li, Yuan Tian, and Najla Al-Nabhan conceived and designed the review. Zheng Li wrote the paper. All authors reviewed and edited the manuscript. All authors read and approved the final manuscript.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61601235 and in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20160972. Meanwhile, the authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no. RG-1441-331.

References

[1] K. Saharan and A. Kumar, "Fog in comparison to cloud: a survey," *International Journal of Computer Applications*, vol. 122, pp. 10–12, 2015.

[2] T. CAI, J. Li, A. S. Mian, R. li, T. Sellis, and J. X. Yu, "Target-aware holistic influence maximization in spatial social networks fluence maximization in spatial social networks," *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2020.

[3] M. Chiang and T. Zhang, "Fog and iot: an overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[4] P. Lai, Q. He, M. Abdelrazek, and F. Chen, "Optimal edge user allocation in edge computing with variable sized vector bin packing," *Service-Oriented Computing*, Springer, Berlin, Germany, pp. 230–245, 2018.

[5] X. Xia, F. Chen, and Q. He, "Cost-effective app data distribution in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 31–44, 2020.

[6] P. M. Mell and T. Grance, *The Nist Definition of Cloud Computing*, CSRC, Beijing, China, 2011.

[7] Z. Li and Y. Wang, "An introduction and comparison of the application of cloud and fog in iot," in *Cloud Computing, Smart Grid and Innovative Frontiers in Telecommunications*, pp. 63–75, Springer, Cham, Switzerland, 2020.

[8] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[9] M. lorga, L. Feldman, and R. Barton, *Fog Computing Conceptual Model*, NIST, Gaithersburg, MD, USA, 2018.

[10] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: state-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2018.

[11] X. Xu, B. Shen, X. Yin et al., "Edge server quantification and placement for offloading social media services in industrial cognitive IoVfication and placement for offloading social media services in industrial cognitive iov," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.

[12] Q. He, G. Cui, X. Zhang et al., "A game-theoretical approach for user allocation in edge computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 3, pp. 515–529, 2020.

[13] P. Vinod, M. Chetan, and K. Sangramsing, "A review-fog computing and its role in the internet of things," *International Journal of Engineering Research and Applications*, vol. 6, pp. 2248–2267, 2016.

[14] Y. Yang, X. Chang, Z. Han, and L. Li, "Delay-aware secure computation offloading mechanism in a fog-cloud framework," in *Proceedings of the 2018 IEEE Intl Conf on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications*, pp. 346–353, Melbourne, Australia, December 2018.

[15] F. Mohamed, G. Osman, and H. Suhaidi, "Fog computing: will it be the future of cloud computing?" in *Proceedings of The Third International Conference on Informatics and Applications*, Kuala Terengganu, Malaysia, October 2014.

[16] C. Systems, *Fog Computing and the Internet of Things: Extend the Cloud to where the Things Are*, ACM, New York, NY, USA, 2015.

[17] X. Chi, C. Yan, H. Wang, W. Rafique, and L. Qi, "Amplified locality-sensitive hashing-based recommender systems with privacy protectionfied locality-sensitive hashing-based recommender systems with privacy protection," *Concurrency and Computation: Practice and Experience*, p. 5681, 2020.

[18] L. Wang, X. Zhang, R. Wang, C. Yan, H. Kou, and L. Qi, "Diversified service recommendation with high accuracy and efficiencyfied service recommendation with high accuracy and efficiency," *Knowledge-Based Systems*, vol. 204, p. 106196, 2020.

[19] W. Zhong, X. Yin, X. Zhang et al., "Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment," *Computer Communications*, vol. 157, pp. 116–123, 2020.

[20] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloudflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2020.

[21] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y.-q. Zhang, "Blockchain empowered arbitrable data auditing scheme for network storage as a service," *IEEE Transactions on Services Computing*, vol. 13, pp. 289–300, 2020.

[22] H. Liu, H. Kou, C. Yan, and L. Qi, "Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph," *Complexity*, vol. 2020, Article ID 2085638, 15 pages, 2020.

[23] H. Sun, H. Yu, G. Fan, and L. Chen, "Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture," *Peer-to-Peer Networking and Applications*, vol. 13, no. 2, pp. 548–563, 2020.

[24] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1594–1608, 2018.

[25] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Transactions on Cloud Computing*, p. 1, 2019.

[26] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "Become: blockchain-enabled computation offloading for iot in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2020.

[27] L. Chen, S. Zhou, and J. Xu, "Energy efficient mobile edge computing in dense cellular networks," 2017, http://arxiv.org/abs/1701.07405.

[28] A. T. Nassar and Y. Yilmaz, "Reinforcement learning-based resource allocation in fog ran for iot with heterogeneous latency requirements," 2018, http://arxiv.org/abs/1806–04582.

[29] S. Chen, Y. Zheng, K. Wang, and W. Lu, "Delay guaranteed energy-efficient computation offloading for industrial iot in fog computing," in *Proceedings of the 2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, May 2019.

[30] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171–1181, 2016.

[31] D. Ye, M. Wu, S. Tang, and R. Yu, "Scalable fog computing with service offloading in bus networks," in *Proceedings of the 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 247–251, Beijing, China, June 2016.

[32] C. Shi, Z. Ren, K. Yang et al., "Ultra-low latency cloud-fog computing for industrial internet of things," in *Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Barcelona, Spain, April 2018.

[33] Q. Fan and N. Ansari, "Towards workload balancing in fog computing empowered iot," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 253–262, 2020.

[34] X. Xu, Q. Wu, L. Qi, W. Dou, S.-B. Tsai, and M. Z. A. Bhuiyan, "Trust-aware service offloading for video surveillance in edge computing enabled internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[35] J. Li, T. Zhang, J. Jin, Y. Yang, D. Yuan, and L. Gao, "Latency estimation for fog-based internet of things," in *Proceedings of the 2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, pp. 1–6, Melbourne, Australia, November 2017.

[36] M. I. Naas, P. R. Parvedy, J. Boukhobza, and L. Lemarchand, "iFogstor: an iot data placement strategy for fog infrastructure," in *Proceedings of the 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, pp. 97–104, Madrid, Spain, May 2017.

[37] P. G. V. Naranjo, Z. Pooranian, M. Shojafar, M. Conti, and R. Buyya, "Focan: a fog-supported smart city network architecture for management of applications in the internet of everything environments," 2019, http://arxiv.org/abs/1710.01801.

[38] R. Craciunescu, A. Mihovska, M. Mihaylov, S. Kyriazakos, R. Prasad, and S. Halunga, "Implementation of fog computing for reliable e-health applications," in *Proceedings of the 2015 49th Asilomar Conference on Signals, Systems and Computers*, pp. 459–463, Pacific Grove, CA, USA, November 2015.

[39] Y. Cao, S. Chen, P. Hou, and D. Brown, "Fast: a fog computing assisted distributed analytics system to monitor fall for stroke mitigation," in *Proceedings of the 2015 IEEE International Conference on Networking, Architecture and Storage (NAS)*, pp. 2–11, Boston, MA, USA, August 2015.

[40] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108–119, 2017.

[41] D. Kimovski, H. Ijaz, N. Saurabh, and R. Prodan, "Adaptive nature-inspired fog architecture," in *Proceedings of the 2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)*, pp. 1–8, Washington, DC, USA, May 2018.

[42] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing iot service delay via fog offloading," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 998–1010, 2018.

[43] M. S. Elbamby, M. Bennis, and W. Saad, "Proactive edge computing in latency-constrained fog networks," in *Proceedings of the 2017 European Conference on Networks and Communications (EuCNC)*, pp. 1–6, Oulu, Finland, June 2017.

[44] A. A. Diro, H. T. Reda, and N. Chilamkurti, "Differential flow space allocation scheme in SDN based fog computing for IoT applicationsfferential flow space allocation scheme in sdn based fog computing for iot applications," *Journal of Ambient Intelligence and Humanized Computing*, 2018.

[45] D. Rahabri and M. Nickray, "Low-latency and energy-efficient scheduling in fog-based iot applications," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27, pp. 1406–1427, 2019.

[46] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[47] R. Kelly, "Internet of things data to top 1.6 zettabytes by 2022," *Campus Technology*, vol. 9, pp. 1536–1233, 2016.

[48] L. Mearian, "Self-driving cars could create 1gb of data a second," *Computerworld*, vol. 23, 2013.

[49] N. Cochrane, "US smart grid to generate 1000 petabytes of data a year," *Expert Systems with Applications*, 2016.

[50] V. Mushunuri, A. Kattepur, H. K. Rath, and A. Simha, "Resource optimization in fog enabled iot deployments," in *Proceedings of the 2017 Second International Conference on Fog and Mobile Edge Computing*, pp. 6–13, Valencia, Spain, May 2017.

[51] C. Zhou, A. Li, A. Hou et al., "Modeling methodology for early warning of chronic heart failure based on real medical big data," *Expert Systems with Applications*, vol. 151, Article ID 113361, 2020.

[52] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, "Community-diversified influence maximization in social networksfied influence maximization in social networks," *Information Systems*, vol. 92, p. 101522, 2020.

[53] A. Yousefpour, A. Patil, G. Ishigaki, I. Kim, and X. e.a. Wang, "Qos-aware dynamic fog service provisioning," 2018, http://arxiv.org/abs/1802.00800.

[54] N. Abbas, M. Asim, N. Tariq, T. Baker, and S. Abbas, "A mechanism for securing iot-enabled applications at the fog layer," *Journal of Sensor and Actuator Networks*, vol. 8, no. 1, p. 16, 2019.

[55] Y. Xu, C. Zhang, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, "Blockchain-Enabled accountability mechanism against information leakage in vertical industry services," *IEEE Transactions on Network Science and Engineering*, p. 1, 2020.

[56] M. Zhang, Y. Duan, H. Yin, and Z. Zhao, "Semantics-aware android malware classification using weighted contextual API dependency graphsfication using weighted contextual api dependency graphs," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security-CCS'14*, pp. 1105–1116, Scottsdale, AZ, USA, November 2014.

[57] L. Martignoni, R. Paleari, and D. Bruschi, "A framework for behavior-based malware analysis in the cloud," in *Information Systems Security*, pp. 178–192, Springer, Berlin, Heidelberg, 2009.

[58] K. Lee, D. Kim, D. Ha, U. Rajput, and H. Oh, "On security and privacy issues of fog computing supported internet of things environment," in *Proceedings of the 2015 6th International Conference on the Network of the Future (NOF)*, pp. 1–3, Montreal, Canada, September 2015.

[59] X. Xu, X. Liu, X. Yin, S. Wang, Q. Qi, and L. Qi, "Privacy-aware offloading for training tasks of generative adversarial network in edge computing," *Information Sciences*, vol. 532, pp. 1–15, 2020.

[60] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[61] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, and L. Qi, "Trust-oriented iot service placement for smart cities in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4084–4091, 2020.

[62] C. Thota, R. Sundarasekar, G. Manogaran, R. Varatharajan, and M. K. Priyan, "Centralized fog computing security platform for IoT and cloud in healthcare system," in *Fog Computing*, pp. 365–378, Springer, Berlin, Heidelberg, 2018.

[63] A. Viejo and D. Sánchez, "Secure and privacy-preserving orchestration and delivery of fog-enabled iot services," *Ad Hoc Networks*, vol. 82, pp. 113–125, 2019.

[64] B. Mukherjee, R. Neupane, and P. Calyam, "End-to-end Iot security middleware for cloud-fog communication," in *Proceedings of the 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 151–156, 2017.

[65] Y. Li, S. Chen, C. Zhao, and W. Lu, "Layered data aggregation with efficient privacy preservation for fog-assisted Iot," *International Journal of Communication Systems*, vol. 33, no. 9, p. 4381, 2020.

[66] W. B. Daoud and M. S. Obaidat, "Tacrm: trust access control and resource management mechanism in fog computing," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, p. 28, 2019.

[67] N. K. Giang, M. Blackstock, R. Lea, and V. C. M. Leung, "Developing iot applications in the fog: a distributed dataflow approach," in *Proceedings of the 2015 5th International Conference on the Internet of Things (IOT)*, pp. 155–162, Seoul, South Korea, October 2015.

[68] M. Zhanikeev, "A cloud visitation platform to facilitate cloud federation and fog computing," *Computer*, vol. 48, no. 5, pp. 80–83, 2015.

[69] T. N. Gia, M. Jiang, A. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing in healthcare internet of things: a case study on ecg feature extraction," in *Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 356–363, Liverpool, UK, October 2015.

[70] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded systemfined embedded system," *IEEE Transactions on Computers*, vol. 65, no. 12, pp. 3702–3712, 2016.

[71] S. Sarkar and S. Misra, "Theoretical modelling of fog computing: a green computing paradigm to support iot applications," *IET Networks*, vol. 5, no. 2, pp. 23–29, 2016.

[72] M. A. Hassan, M. Xiao, Q. Wei, and S. Chen, "Help your mobile applications with fog computing," in *Proceedings of the 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking-Workshops (SECON Workshops)*, pp. 1–6, Seattle, WA, USA, June 2015.

[73] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing," in *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 1202–1207, Ottawa, Canada, May 2015.

[74] S. Ningning, G. Chao, A. Xingshuo, and Z. Qiang, "Fog computing dynamic load balancing mechanism based on graph repartitioning," *China Communications*, vol. 13, no. 3, p. 156, 2016.

[75] C. MacGillivray, L. Lamy, R. Segal, and M. Torchia, *Idc Futurescape: Worldwide Internet of Things 2017 Predictions*, IDC, Framingham, MA, USA, 2016.

[76] A. V. Dastjerdi and R. Buyya, "Fog computing: helping the internet of things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, 2016.

[77] S. Singh, I. Chana, and R. Buyya, "Star: sla-aware autonomic management of cloud resources," *IEEE Transactions on Cloud Computing*, p. 1, 2017.

[78] S. Mustafa, K. Bilal, S. U. R. Malik, and S. A. Madani, "Sla-aware energy efficient resource management for cloud environments," *IEEE Access*, vol. 6, pp. 15004–15020, 2018.

[79] Y. Wang, Q. He, D. Ye, and Y. Yang, "Formulating criticality-based cost-effective fault tolerance strategies for multi-tenant service-based systems effective fault tolerance strategies for multi-tenant service-based systems," *IEEE Transactions on Software Engineering*, vol. 44, no. 3, pp. 291–307, 2018.

[80] L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, and X. Xu, "A qos-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems," *World Wide Web*, vol. 23, no. 2, p. 1275, 2019.

[81] J. Bi, H. Yuan, M. Tie, and W. Tan, "Sla-based optimisation of virtualised resource for multi-tier web applications in cloud data centres," *Enterprise Information Systems*, vol. 9, no. 7, pp. 743–767, 2015.

[82] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for iot systems: a computation offloading

game," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, 2018.

[83] I. Stojmenovic and S. Wen, "The fog computing paradigm: scenarios and security issues," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, pp. 1–8, Szczecin, Poland, September 2014.

[84] J. Li, J. Jin, D. Yuan, M. Palaniswami, and K. Moessner, "Ehopes: data-centered fog platform for smart living," in *Proceedings of the 2015 International Telecommunication Networks and Applications Conference (ITNAC)*, pp. 308–313, Sydney, Australia, November 2015.

[85] A. Dasgupta and A. Q. Gill, "Fog computing challenges: a systematic review," in *Proceedings of the Australasian Conference on Information Systems*, Hobart, Australia, December 2017, https://aisel.aisnet.org/acis2017/79.

[86] Y. Hong, W. M. Liu, and L. Wang, "Privacy preserving smart meter streaming against information leakage of appliance status," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, pp. 2227–2241, 2017.

[87] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, pp. 37–42, New York, NY, USA, June 2015.

[88] L. M. Vaquero and L. Rodero-Merino, "Finding your Way in the Fogfinition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[89] A. Aijaz, M. Dohler, A. H. Aghvami, V. Friderikos, and M. Frodigh, "Realizing the tactile internet: haptic communications over next generation 5g cellular networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 82–89, 2017.

[90] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5g-enabled tactile internet," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 460–473, 2016.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8824430, 9 pages
https://doi.org/10.1155/2020/8824430

Research Article

An Improved Elman Network for Stock Price Prediction Service

Bo Liu , Qilin Wu, and Qian Cao

School of Information Engineering, Chaohu University, Chaohu 238000, China

Correspondence should be addressed to Qian Cao; 19875069@qq.com

Received 16 June 2020; Revised 8 August 2020; Accepted 19 August 2020; Published 3 September 2020

Academic Editor: Xiaolong Xu

The rapid development of edge computing drives the rapid development of stock market prediction service in terminal equipment. However, the traditional prediction service algorithm is not applicable in terms of stability and efficiency. In view of this challenge, an improved Elman neural network is proposed in this paper. Elman neural network is a typical dynamic recurrent neural network that can be used to provide the stock price prediction service. First, the prediction model parameters and build process are analysed in detail. Then, the historical data of the closing price of Shanghai composite index and the opening price of Shenzhen composite index are collected for training and testing, so as to predict the prices of the next trading day. Finally, the experiment results validate that it is effective to predict the short-term future stock price by using the improved Elman neural network model.

1. Introduction

The stock market can be regarded as a complex nonlinear system, and there are many factors that affect the stock price, especially the recent historical stock price, which has a great influence on the future short-term stock price. So, it is difficult, but valuable, to provide stock price prediction service. Fortunately, with the development of edge computing and neural network technologies, commercial service providers can benefit from the low-latency edge resources and nonlinear expression ability of neural network to provide their users with more efficient service acquisition for stock price prediction. Based on this well-known cognition, we can design a neural network to predict the stock price of the next period based on the historical stock price [1–4]. In this paper, we will use the historical data of the closing price of the Shanghai composite index to predict the closing price of the Shanghai composite index in the next trading day, and the historical data of the opening price of the Shenzhen composite index to predict the opening price of the Shenzhen composite index in the next trading day. In addition, our research could make stock prediction algorithms deployed on edge terminals more efficient.

Over the years, although many scholars have established a large number of mathematical models to predict the stock price, they have not achieved good results and have little impact. However, the rise of big data technology and artificial intelligence technology will provide another effective solution for stock price prediction. This is the motivation of our research. Specifically, we hope to establish a reasonable artificial intelligence model and make a more accurate prediction of the future short-term stock price by inputting the latest stock price history. We expect this model can offer a reference for people of stock investment.

In this paper, we propose an improved Elman neural network model to predict stock price, and our main contributions include the following:

In order to apply traditional stock prediction algorithms to terminal devices such as edge computing and mobile phones, we build a stock price prediction model based on an improved Elman network with the aim to predict the stock price simpler and more stable. We give the specific model parameters and build process.

In order to reflect the latest stock market situation, we trained and tested the proposed model with the latest dataset, namely, the Shanghai composite index and Shenzhen composite index in 2018, 2019, and 2020; the latest datasets are used to better reflect the current stock market.

> To analyse the new algorithm model more clearly, we quantitatively analysed the performance of the model with a variety of mathematical tools and error analysis methods. In addition, a large number of diagrams and tables are provided to further clarify the model.

The rest of this paper is organized as follows. Section 2 reviews and summarizes the related work, on this basis, to clarify the significance of this study. Section 3 is preliminaries in which the principle of Elman neural network is clarified. In Section 4, we proposed our model, and the specific model construction procedures are introduced in detail. Section 5 is experiments in which the model is built, trained, and tested. In addition, we devoted a great deal of space to the analysis of the results in this section. Finally, Section 6 concludes this paper.

2. Preliminaries

Elman neural network is a typical feedback neural network model widely used, which is generally divided into four layers: input layer, hidden layer, bearing layer, and output layer [5, 6].

Figure 1 shows the basic structure of an Elman neural network; the connection of the input layer, hidden layer, and output layer are similar to the feedforward network. The input layer unit only plays the role of signal transmission, and the output layer unit plays the role of weighting. There are linear and nonlinear excitation functions in the hidden layer element, and the excitation function usually takes the nonlinear function of Sigmoid. The bearing layer is used to remember the output value of the hidden layer unit at the previous time, which can be considered as a delay operator with one-step delay. The output of the hidden layer is self-linked to the input of the hidden layer by accepting the delay and storage of the layer, which makes it sensitive to historical data. In other words, Elman neural network adds a bearing layer to the hidden layer as a one-step delay operator to achieve the purpose of memory so that the system has the ability to adapt to time-varying characteristics and enhance the global stability of the network [7–9]. The mathematical expression of its network is

$$\begin{aligned} y(k) &= g\left(w_3 x(k)\right), \\ x(k) &= f\left(w_1 x_c(k) + w_2\left(u(k-1)\right)\right), \\ x_c(k) &= x(k-1), \end{aligned} \tag{1}$$

where $y(t)$ is the output node vector, $x(t)$ is the nodal element vector of the hidden layer, $u(t-1)$ is the input vector, $x_c(t)$ is the feedback state vector, w_3 is the connection weight from the hidden layer to the output layer, w_2 is the connection weight from the input layer to the hidden layer, w_1 is the connection weight of the connecting layer to the hidden layer, g is the transfer function of the output neuron and the linear combination of the output of the hidden layer, and f is the transfer function of the hidden layer neuron, usually using the S function.

This research takes MATLAB as the experimental platform. And the two datasets used in this study are the closing prices of 490 trading days of the Shanghai composite index from September 26, 2017, to September 30, 2019, and the opening prices of 420 trading days from August 15, 2018, to May 12, 2020. We will use the same model for training and testing based on these two datasets.

3. Related Work

In fact, many researchers have been studying stock price forecasts for years, some of these studies have improved the existing models and some have further processed the data. However, these studies are not perfect, and some of the models are too complex and some of the processing procedures are tedious. These shortcomings will increase the instability of the models and limit the application and extension of the research results.

Shi et al. considered that traditional stock forecasting methods could not fit and analyse highly nonlinear and multifactors of stock market well, so there are problems such as low prediction accuracy and slow training speed. Therefore, they proposed a prediction method of the Elman neural network model based on the principal component analysis method. In order to compare the results better, BP network and Elman network with the same structure are established to predict the stock data [10]. Yu et al. used an improved Elman neural network as the forecasting model and the market price of Zhongji company (No. 000039) in Shenzhen stock market is forecasted; their experiment results get higher precision, steadier forecasting effect, and more rapid convergence speed [11]. Zheng et al. studied the forecast of opening stock price based on Elman neural network in 2015, and they selected the opening prices of Shanghai stock index of 337 trading days from December, 2012 to April, 2014 as the raw data for stimulated forecast, and the result proves the validity of their forecast model [12]. Zhang et al. successfully applied Elman regression neural network to the prediction of stock opening price. Specifically, the authors described the Particle Swarm Optimization (PSO) algorithm for learning optimization of the Elman Recurrent Neural Network, and the results showed that the model based on LSTM was more accurate than other machine learning models [13]. Jun used Adaptive Whale Optimization Algorithm and Elman neural network to predict the stock price and achieved better results based on their experiments [14]. Javad Zahedi et al. used the artificial neural network model and principal component analysis to evaluate the predictability of stock price in Teheran stock exchange with 20 accounting variables. Finally, the goodness of fit of principal component analysis was determined by actual values, and the effective factors of Teheran stock exchange price were accurately predicted and modelled by a new model composed of all variables [15]. Han et al. designed a three-ply BP network and the corresponding mathematical model. Therefore, using 140 days actual price of the stock 600688 as a sample, the network was trained through MATLAB; thereby, the 10 days predictions of the stock price and the dispersion $Q = 0.\ 0146$ to the practical data were made [16].

Although scholars have made outstanding contributions in using artificial intelligence to predict stock prices, neither the stability of the models nor the accuracy of the predictions

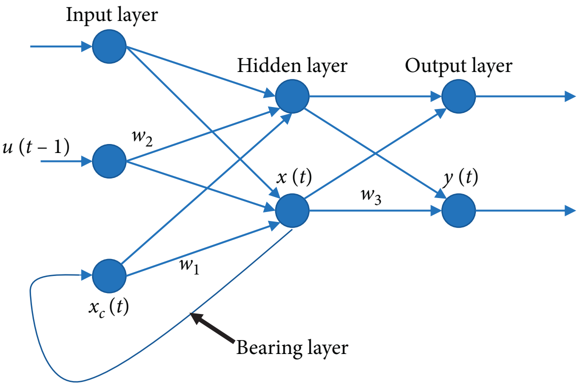


Figure 1: The basic structure of an Elman neural network.

is satisfactory. Based on this fact, this study seeks to exploit the neural network model for the prediction of stock price based on Elman network with balancing the simplicity, stability, and accuracy.

4. Supposed Model

The general steps to build the supposed model of this study include data collection, data load, sample set construction, division of sample set and training set, construction of Elman neural network, and training of the neural network model. The specific flow chart is shown in Figure 2.

4.1. Construction of Sample Set. The stock price prediction problem in this study is actually a time series problem, which can be expressed by the following formula:

$$x_n = f\left(x_{n-1}, x_{n-2}, \cdots, x_{n-N}\right). \tag{2}$$

This formula means that the closing price of the previous N trading day can be used to predict the closing price of the next trading day. The data of 490 closing prices were divided into training samples and test samples; for the training samples, $x_1 \sim x_N$ are selected to form the first sample, where $(x_1, x_2, \cdots, x_{N-1})$ are the independent variable and x_N is the dependent variable, and $x_2 \sim x_{N+1}$ are selected to form the second sample, where $(x_2, x_3, \cdots, x_N)$ are the independent variable and x_{N+1} is the dependent variable; finally, a training matrix is formed as follows:

$$\begin{pmatrix} x_1 & x_2 & x_i \\ x_2 & x_3 & x_{i+1} \\ \cdots & \cdots & \cdots \\ x_{N-1} & x_N & \\ x_N & x_{N+1} & \end{pmatrix}. \tag{3}$$

In this matrix, each column is a sample, and the last row is the expected output. These samples are fed into the Elman neural network for training, and then the network model can be obtained [17–19].

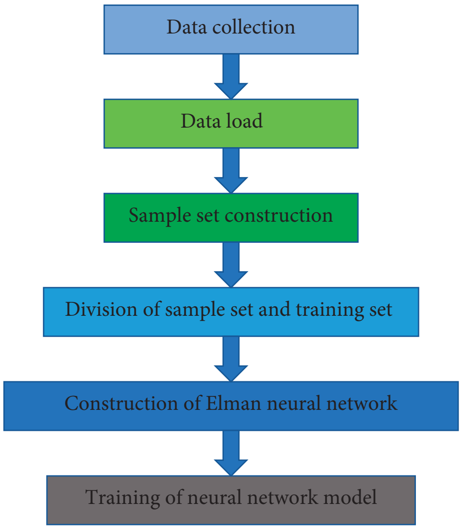

Data collection
Data load
Sample set construction
Division of sample set and training set
Construction of Elman neural network
Training of neural network model

Figure 2: Elman neural network model construction steps.

In this study, $x_1 \sim x_N$ are selected to form the first sample, and $x_2 \sim x_{N+1}$ are selected to form the second sample; the rest can be carried out in the same manner. Here, N is randomly set to 8, which means that the closing price of the day is determined by the closing price of the previous seven trading days.

Take the Shanghai composite index dataset as an example, the closing prices of the first eight trading days are 3,343.58, 3345.27, 3339.64, 3348.94, 3374.38, 3382.99, 3388.28, and 3386.10, which means 3,343.58, 3345.27,3339.64, 3348.94, 3374.38, 3382.99, and 3388.28 will be used to forecast the eighth data 3388.28 which we have already obtained. The closing prices of the first eight trading days are 2999.28, 3006.45, 2977.08, 2985.34, 2955.43, 2929.09, 2932.17, and 2905.19, and the same principle, 2999.28, 3006.45, 2977.08, 2985.34, 2955.43, 2929.09, and 2932.17, will be used to forecast the eighth data 2905.19 which we have already obtained. Therefore, 490 pieces of data will be converted into a 8 × 483 matrix; 483 columns mean 483 samples, in which the first 7 data in each column are independent variables and the eighth data is the data to be predicted. The 8 × 483 matrix is shown as follows:

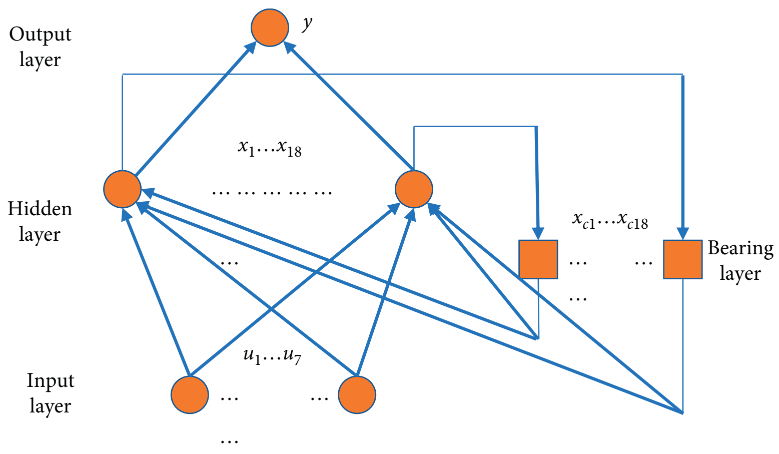


Figure 3: Model structure.

$$\begin{pmatrix} 3343.58 & 3390.52 & \cdots & 2999.28 \\ 3345.27 & 3378.47 & \cdots & 3006.45 \\ 3339.64 & 3372.04 & \cdots & 2977.08 \\ 3348.94 & 3381.79 & \cdots & 2985.34 \\ 3374.38 & 3370.17 & \cdots & 2955.43 \\ 3382.99 & 3378.65 & \cdots & 2929.09 \\ 3388.28 & 3380.7 & \cdots & 2932.17 \\ 3386.1 & 3388.25 & \cdots & 2905.19 \end{pmatrix}. \quad (4)$$

The Shenzhen composite index dataset is 8786.3497, 8470.9094, 8573.5693, 8355.0002, 8419.7868, 8533.4289, 8446.9836, 8480.2244, 8511.3743, 8731.6394, 8716.8172, 8666.9025, 8509.2723, 8440.9528, 8454.1357, 8519.5698, ……, 10477.7614, 10460.9947, 10575.5242, 10618.1651, 10899.9169, 10923.6123, 11053.8157, 10972.0503. In the same way, the Shenzhen composite index dataset is formed as a 8×413 matrix, which is as follows:

$$\begin{pmatrix} 8786.3497 & 8511.3743 & \cdots & 10477.7614 \\ 8470.9094 & 8731.6394 & \cdots & 10460.9947 \\ 8573.5693 & 8716.8172 & \cdots & 10575.5242 \\ 8355.0002 & 8666.9025 & \cdots & 10618.1651 \\ 8419.7868 & 8509.2723 & \cdots & 10899.9169 \\ 8533.4289 & 8440.9528 & \cdots & 10923.6123 \\ 8446.9836 & 8454.1357 & \cdots & 11053.8157 \\ 8480.2244 & 8519.5698 & \cdots & 10972.0503 \end{pmatrix}. \quad (5)$$

413 columns mean 413 samples, in which the first 7 data in each column are independent variables and the eighth data is the data to be predicted.

4.2. Construction of Elman Neural Network. Figure 3 shows the proposed model structure, where $u_1, \cdots, u_7$ are input data, $x_1, \cdots, x_{18}$ are hidden-layer data, and $x_{c1}, \cdots, x_{c18}$ are bearing-layer data. With the help of MATLAB neural network toolbox, Elman neural network can be easily built. To be specific, the MATLAB neural network toolbox provides an Elmannet function, and Elman network construction can be completed by setting three parameters in the Elmannet function, which are the delay time, the number of hidden layer neurons, and the training function, respectively. In this case, the number of hidden-layer neurons is set to be 18, and TRAINGDX is chosen to be the training function [20–22]. TRAINGDX, which is named gradient descent with momentum and adaptive learning rate backpropagation, is a network training function that updates weight and bias values according to gradient descent momentum and an adaptive learning rate. It will return a trained net and the training record. In addition, the maximum number of iterations in the training is set to 3000, the maximum number of validation failures is set to 6, and the error tolerance is set to 0.00001, which means that the training can be stopped if the error value is reached [23, 24]. Figure 4 shows the model structure graphic automatically generated by MATLAB.

To construct the Elman neural network, the MATLAB code can be like this. Firstly, three parameters in the Elmannet function are set, and codes are as follows:

$$\text{net} = \text{Elmannet}\,(1:2, 18, \text{'TRAINGDX'}). \quad (6)$$

Secondly, the maximum number of iterations in the training is set to 3000, and codes are as follows:

$$\text{net.trainParam.epochs} = 3000. \quad (7)$$

Thirdly, the maximum number of validation failures is set to 6, and the error tolerance is set to 0.00001, and codes are as follows:

$$\begin{gathered} \text{net.trainParam.}\max\text{\_fail} = 6, \\ \text{net.trainParam.goal} = 0.00001. \end{gathered} \quad (8)$$

Finally, initialize the network, and codes are as follows:

$$\text{net} = \text{init}\,(\text{net}). \quad (9)$$

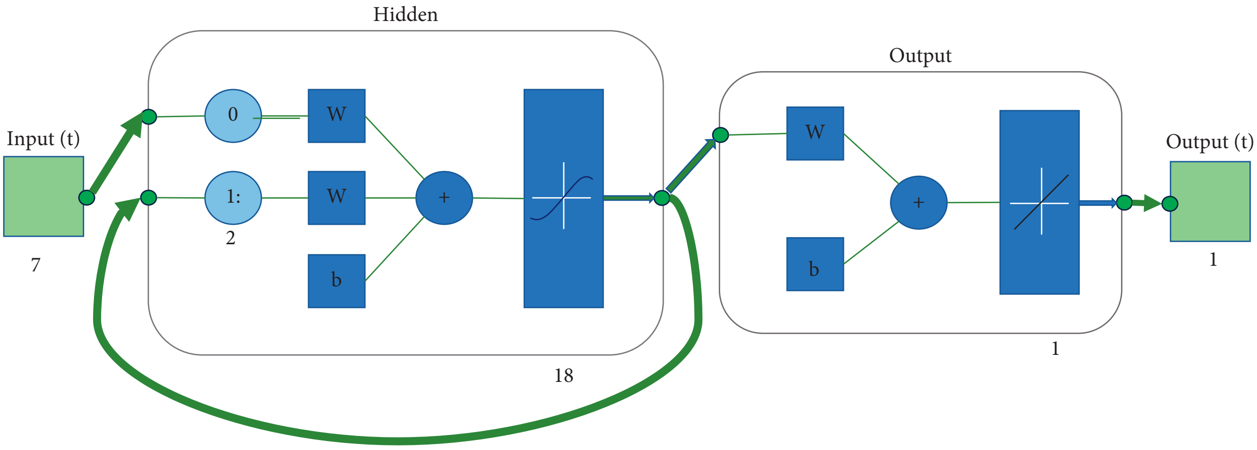


Figure 4: The model structure graphic generated by MATLAB.

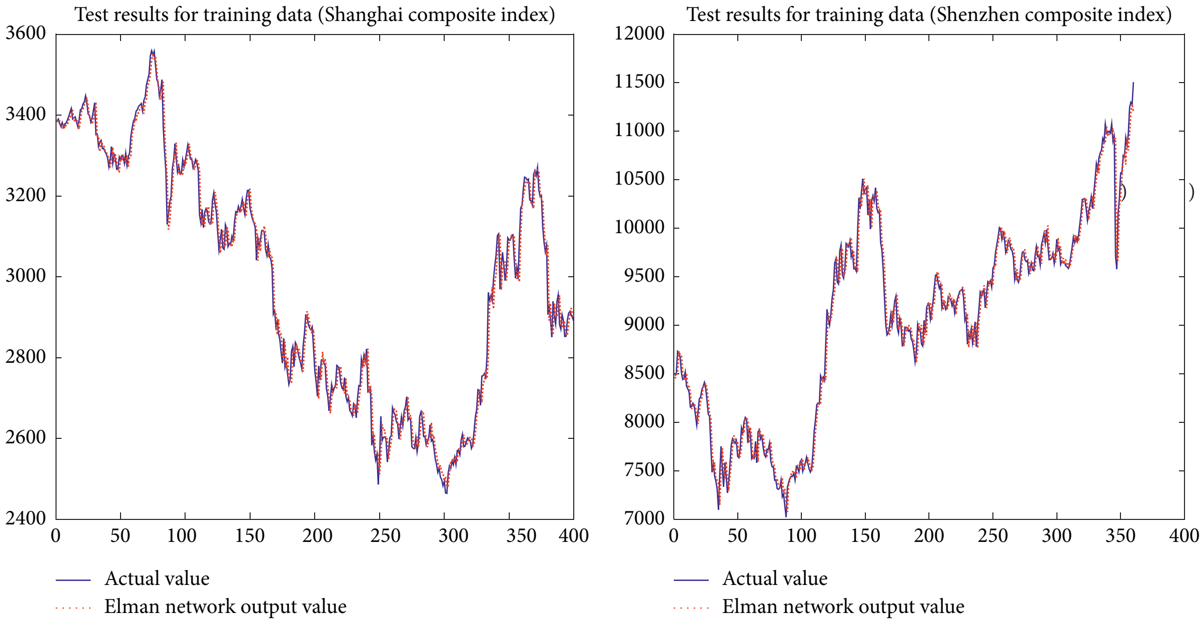


Figure 5: Test results for training data.

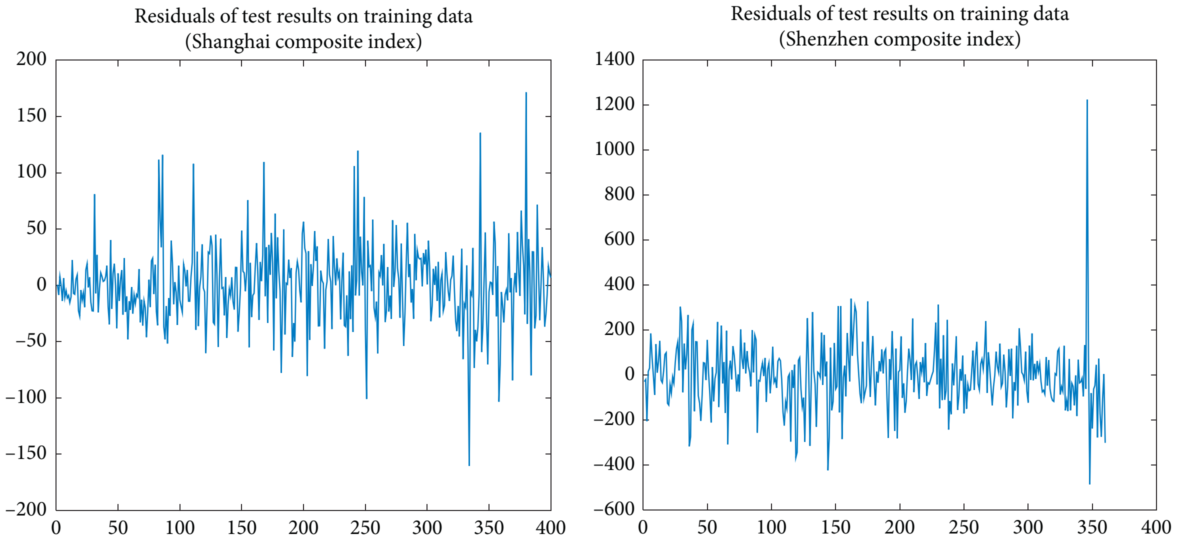


Figure 6: Residuals of test results on training data.

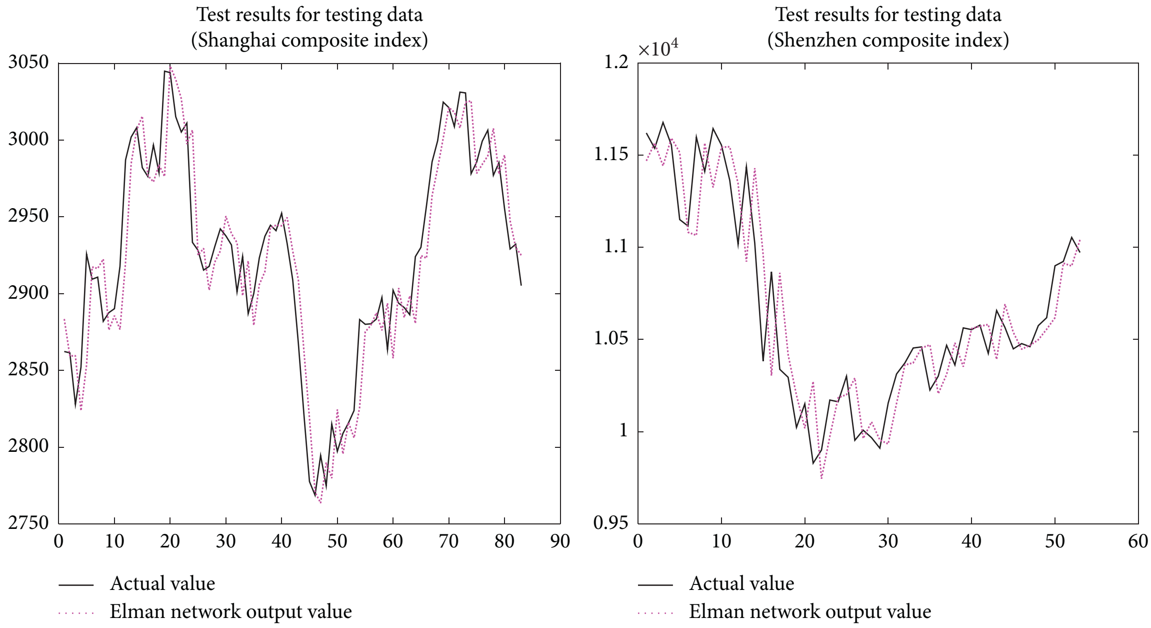


Figure 7: Test results for testing data.

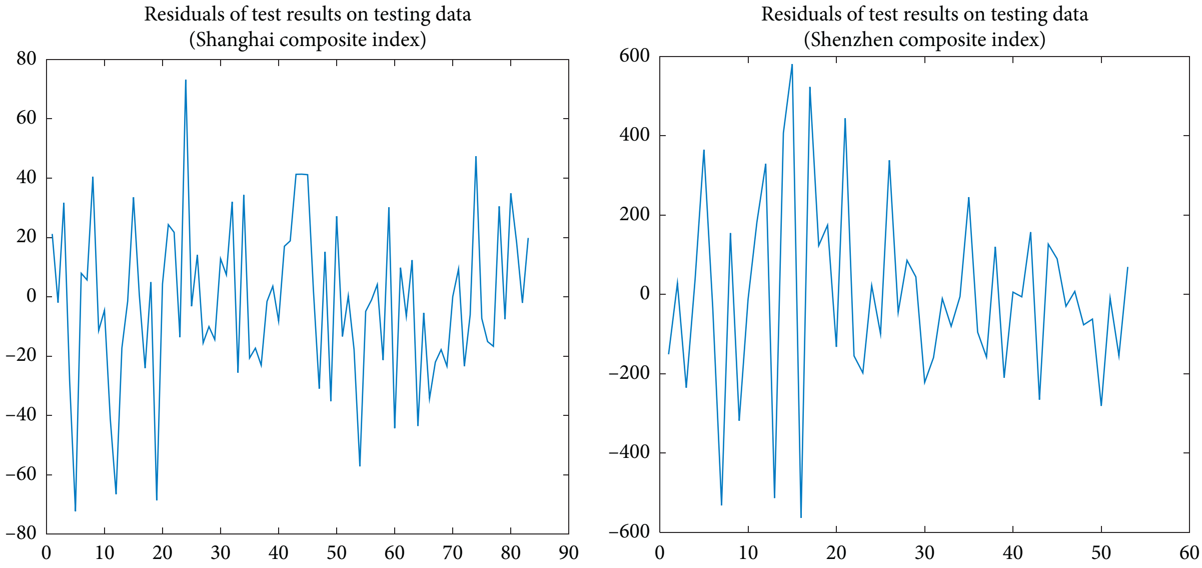


Figure 8: Residuals of test results on testing data.

Table 1: Relative error of test (Shanghai composite index).

Number	1	2	3	4	5	6	7
Relative error	0.006674	−0.001090	0.010719	−0.010057	−0.025153	0.001690	0.000677
Number	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Relative error	0.012402	−0.003898	−0.002381	−0.015004	−0.023218	−0.007697	−0.002238
Number	**15**	**16**	**17**	**18**	**19**	**20**	**21**
Relative error	0.009203	−0.000263	−0.009339	0.000532	−0.022964	−0.000520	0.007789
Number	**22**	**23**	**24**	**25**	**26**	**27**	**28**
Relative error	0.006250	−0.004937	0.023693	−0.000488	0.004250	−0.005476	−0.003717
Number	**29**	**30**	**31**	**32**	**33**	**34**	**35**
Relative error	−0.005492	0.003612	0.001676	0.010246	−0.009194	0.010889	−0.007296
Number	**36**	**37**	**38**	**39**	**40**	**41**	**42**
Relative error	−0.006752	−0.008459	−0.001553	−0.000139	−0.003553	0.004742	0.005815
Number	**43**	**44**	**45**	**46**	**47**	**48**	**49**

Table 1: Continued.

Relative error	0.013786	0.014388	0.014745	0.000828	−0.010776	0.005480	−0.012749
Number	**50**	**51**	**52**	**53**	**54**	**55**	**56**
Relative error	0.009279	−0.004690	0.000227	−0.006596	−0.019610	−0.002958	−0.001138
Number	**57**	**58**	**59**	**60**	**61**	**62**	**63**
Relative error	0.000246	−0.007637	0.009974	−0.016051	0.002405	−0.002644	0.003361
Number	**64**	**65**	**66**	**67**	**68**	**69**	**70**
Relative error	−0.015557	−0.002587	−0.012756	−0.009015	−0.007153	−0.009234	−0.001688
Number	**71**	**72**	**73**	**74**	**75**	**76**	**77**
Relative error	0.002176	−0.008728	−0.003110	0.015200	−0.002282	−0.005979	−0.006169
Number	**78**	**79**	**80**	**81**	**82**	**83**	
Relative error	0.009414	−0.002398	0.010743	0.006008	−0.001105	0.006332	

Table 2: Relative error of test (Shenzhen composite index).

Number	1	2	3	4	5	6	7
Relative error	−0.013747	0.002825	−0.017177	0.004370	0.030084	−0.001853	−0.047231
Number	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Relative error	0.017591	−0.017927	0.007011	0.010986	0.026179	−0.044038	0.035607
Number	**15**	**16**	**17**	**18**	**19**	**20**	**21**
Relative error	0.061603	−0.043408	0.047926	0.008911	0.028250	−0.016142	0.039523
Number	**22**	**23**	**24**	**25**	**26**	**27**	**28**
Relative error	−0.012140	−0.021609	0.001765	−0.010977	0.036256	−0.005063	0.006179
Number	**29**	**30**	**31**	**32**	**33**	**34**	**35**
Relative error	0.003674	−0.022697	−0.014897	−0.002466	−0.006432	0.001260	0.023451
Number	**36**	**37**	**38**	**39**	**40**	**41**	**42**
Relative error	−0.009376	−0.016781	0.009971	−0.019686	0.001902	−0.002325	0.014443
Number	**43**	**44**	**45**	**46**	**47**	**48**	**49**
Relative error	−0.024677	0.009285	0.008341	−0.003239	−0.000341	−0.010959	−0.005216
Number	**50**	**51**	**52**	**53**	**54**	**55**	**56**
Relative error	−0.026052	−0.000483	−0.012797	0.007071			

After all the above steps, construction of Elman neural network is completed [25–27].

5. Experiments

5.1. Training of the Supposed Model. When the Elman neural network is built, the model can be trained, but all the data has to be normalized first considering of the performance and stability of the model. The normalization operation can use the mapminmax function provided by MATLAB toolbox, and the default normalization interval of mapminmax function is $[-1, 1]$. The detailed MATLAB code is as follows:

$$\begin{aligned}\left[\mathrm{train}\, x_1, \mathrm{st}_1\right] &= \mathrm{map}\min\max\left(\mathrm{train}\, x\right),\\ \mathrm{train\_ty}_1 &= \mathrm{sim}\left(\mathrm{net}, \mathrm{train}\, x_1\right),\\ \mathrm{train\_ty} &= \mathrm{map}\min\max\left(\text{‘reverse’}, \mathrm{train\_ty}_1, \mathrm{st}_2\right).\end{aligned} \tag{10}$$

After the normalized operation of training data, trainx and trainx1 were obtained. The normalized training data (trainx1) were input into the network model to obtain the current network output (train_ty1) and then reversely normalized into normal data to obtain train_ty, which is the corresponding stock price of the training data. What we want to emphasize is that the data used in the test should be normalized first and then the output should be unnormalized.

5.2. The Test Results and the Quantitative Analysis. Figure 5 shows a graph of the actual and predicted values; the blue solid line is the actual value and the red dotted line represents the Elman network output value. Apparently, the model fits the training data well. In addition, we further calculated the residuals of test results on training; Figure 6 shows the residuals of training results on training data, and residual in mathematical statistics refers to the difference between the actual observed value and the estimated value (the fitting value).

Figure 7 shows a graph of the actual and predicted values; the black solid line is the actual value and the red dotted line represents the Elman network output value. In addition, we further calculate the residuals of test results on testing data; Figure 8 shows the residuals of test results on testing data. And, the relative errors of each prediction are also calculated for further study and analysis. All relative error values are shown in Tables 1 and 2. By analysing these graphs and data, it is clear that the prediction effect of the model is pretty good.

6. Conclusions

This study is based on a basic premise that the historical stock price will have a great impact on the future short-term stock price. On this premise, we established an improved Elman model and collected the historical data of the Shanghai composite index and the Shenzhen composite index as a dataset for the experiment. As for dataset processing, we divided two datasets, one for training and the other for testing. In addition, the data were normalized. Regarding model building, we take MATLAB as the platform, and set the number of hidden-layer neurons to be 18. TRAINGDX is chosen to be the training function. In terms of training, the maximum number of iterations in the training is set to 3000, the maximum number of validation failures is set to 6, and the error tolerance is set to 0.00001. Finally, we use the model to test the training data and the test data. In order to analyse the experimental results, we also calculated relative error and the residuals and drew a picture to show them. Based on Elman network, this study predicted the short-term stock price in the future and achieved a good prediction effect. However, it is unrealistic to predict the long-term stock price in the future, which is difficult to achieve [28–30]. This study provides an effective experimental method for predicting the near future stock price.

Data Availability

All of the data used in this study are already available on the Internet and is easily accessible.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the key project of the Natural Science Research of Higher Education Institutions in Anhui Province (Grant no. KJ2018A0461); the Anhui Province Key Research and Development Program Project (Grant no. 201904a05020091); the Provincial Quality Engineering Project from Department of Education Anhui Province (Grant no. 2019mooc283); and the Domestic and Foreign Research and Study Program for Outstanding Young Backbone Talents in Colleges and Universities (Grant no. Gxgnfx2019034).

References

[1] Q. Shayea, *Neural Networks to Predict Stock Market Price*, World Congress on Engineering and Computer Science, San Francisco, CA USA, 2017.

[2] X. Xu, B. Shen, X. Yin et al., "Edge server quantification and placement for offloading social media services in industrial cognitive IoV," *IEEE Transactions on Industrial Informatics*, 2020.

[3] V. Rohit, C. Pkumar, and S. Upendra, "Neural networks through stock market data prediction," in *Proceedings of the 2017 International Conference of Electronics*, Coimbatore, India, April 2017.

[4] D. Das, A. S. Sadiq, N. B. Ahmad, and J. Lloret, "Stock market prediction with big data through hybridization of data mining and optimized neural network techniques," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 29, no. 1-2, pp. 157–181, 2017.

[5] J. Zahedi and M. Rounaghi, "Application of artificial neural network models and principal component analysis method in predicting stock prices on tehran stock exchange," *Physica A: Statistical Mechanics and its Application*, vol. 38, pp. 178–187, 2015.

[6] X. Han, "Stock price prediction with neural network based on MATLAB," *Systems Engineering*, 2003.

[7] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2020.

[8] R. Mahanta, T. N. Pandey, A. K. Jagadev, and S. Dehuri, "Optimized radial basis functional neural network for stock index prediction," in *Proceedings of the IEEE Conference Publications*, Chennai, India, March 2016.

[9] https://www.mathworks.com/help/deeplearning/ref/Elmannet.html.

[10] Y. Zhang, G. Cui, S. Deng et al., "Efficient query of quality correlation for service composition," *IEEE Transactions on Services Computing*, p. 1, 2018.

[11] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. Alam Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned Internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[12] Y. Zhang, C. Yin, Q. Wu et al., "Location-aware deep collaborative filtering for service recommendation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems (TSMC)*, 2019.

[13] J. Yu and P. Guo, "Stock price forecasting Model Based on Improved Elman Neural Network," *Computer Technology and Development*, 2008.

[14] H. Shi and X. Liu, "Application on stock price prediction of Elman neural networks based on principal component analysis method," in *Proceedings of the 2014 11th International Computer Conference on Wavelet Active Media Technology & Information Processing*, Chengdu, China, December 2014.

[15] X. Zhang, S. Qu, J. Huang, B. Fang, and P. Yu, "Stock market prediction via multi-source multiple instance learning," *IEEE Access*, vol. 6, no. 99, pp. 50720–50728, 2018.

[16] M. Billah, S. Waheed, and A. Hanifa, "Predicting closing stock price using artificial neural network and adaptive neuro fuzzy inference system (ANFIS: the case of the Dhaka stock exchange," *International Journal of Computer Applications*, vol. 129, no. 11, pp. 1–5, 2015.

[17] https://www.mathworks.com/help/deeplearning/index.html?s_tid=CRUX_lftnav.

[18] Z. Zhang, Y. Shen, and G. Zhang, "Short-term prediction for opening price of stock market based on self-adapting variant PSO-elman neural network," in *Proceedings of the IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, November 2017.

[19] V. Andrea and L. Karel, "MatConvNet-convolutional neural networks for MATLAB," in *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 689–692, Brisbane, Australia, October 2015.

[20] L. Ren, Y. Liu, Z. Rui, H. Li, and R. Feng, "Application of elman neural network and MATLAB to load forecasting," in *Proceedings of the International Conference on Information Technology and Computer Science*, Kiev, Ukraine, July 2009.

[21] K. Kim and W. Lee, "Stock market prediction using artificial neural networks with optimal feature transformation," *Neural Computing & Applications*, vol. 13, no. 3, pp. 255–260, 2004.

[22] H. Grigoryan, "Stock market prediction using artificial neural networks. Case Study of TAL1T, Nasdaq OMX Baltic Stock," *Database Systems Journal*, 2015.

[23] S. Nayak, B. Misra, and H. Behera, "An adaptive second order neural network with genetic-algorithm-based training (ASONN-GA) to forecast the closing prices of the stock market," *International Journal of Applied Metaheuristic Computing*, vol. 7, no. 2, pp. 39–57, 2016.

[24] Y. Zhang, K. Wang, Q. He et al., "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Transactions on Services Computing*, 2019.

[25] P. Kai, H. Huang, S. Wan, and V. Leung, "End-edge-cloud collaborative computation offloading for multiple mobile users in heterogeneous edge-server environment," *Wireless Network*, vol. 2020, 2020.

[26] C. Goutami and S. Chattopadhyay, "Monthly sunspot number time series analysis and its model construction through autoregressive artificial neural network," *The European Physical Journal Plus*, vol. 127, no. 4, 2012.

[27] Z. Guo, J. Wu, H. Lu, and J. Wang, "A case study on a hybrid wind speed forecasting method using BP neural network," *Knowledge-Based Systems*, vol. 24, no. 7, pp. 1048–1056, 2011.

[28] Y. Zhang and L. Wu, "Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network," *Expert Systems with Applications*, vol. 36, no. 5, pp. 8849–8854, 2008.

[29] X. Han, "Stock Price Prediction with Neural Network Based on MATLAB," *Systems Engineering*, vol. 2003, 2003.

[30] K. Peng, B. Zhao, S. Xue, and Q. Huang, "Energy- and resource-aware computation offloading for complex tasks in edge environment," *Complexity*, vol. 2020, Article ID 9548262, 14 pages, 2020.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8892954, 13 pages
https://doi.org/10.1155/2020/8892954

Research Article

A Dynamic Privacy Protection Mechanism for Spatiotemporal Crowdsourcing

Tianen Liu,[1] Yingjie Wang,[1] Zhipeng Cai,[2] Xiangrong Tong,[1] Qingxian Pan,[1] and Jindong Zhao[1]

[1] *School of Computer and Control Engineering, Yantai University, Yantai 264005, China*
[2] *Department of Computer Science, Georgia State University, Atlanta 30303, GA, USA*

Correspondence should be addressed to Yingjie Wang; towangyingjie@163.com

Received 25 May 2020; Revised 26 June 2020; Accepted 29 July 2020; Published 28 August 2020

Academic Editor: Xiaolong Xu

In spatiotemporal crowdsourcing applications, sensing data uploaded by participants usually contain spatiotemporal sensitive data. If application servers publish the unprocessed sensing data directly, it is easy to expose the privacy of participants. In addition, application servers usually adopt the static publishing mechanism, which is easy to produce problems such as poor timeliness and large information loss for spatiotemporal crowdsourcing applications. Therefore, this paper proposes a spatiotemporal privacy protection (STPP) method based on dynamic clustering methods to solve the privacy protection problem for crowd participants in spatiotemporal crowdsourcing systems. Firstly, the working principles of a dynamic privacy protection mechanism are introduced. Then, based on k-anonymity and l-diversity, the spatiotemporal sensitive data are anonymized. In addition, this paper designs the dynamic k-anonymity algorithm based on the previous anonymous results. Through extensive performance evaluation on real-world data, compared with existing methods, the proposed STPP algorithm could effectively solve the problem of poor timeliness and improve the privacy protection level while reducing the information loss of sensing data.

1. Introduction

With the widespread use of wireless communication technologies and smart mobile terminals, location-based services (LBS) are becoming more and more popular [1, 2]. In many spatiotemporal crowdsourcing applications, participants receive corresponding rewards by submitting their own sensing tasks to crowdsourcing application servers [3]. However, the submitted sensing data contain the participants' spatiotemporal data [4, 5]. If the crowdsourcing application server publishes these spatiotemporal data without processing, the participant's privacy information will be obtained by attackers [6, 7]. More importantly, attackers can infer the participant's recent medical service or entertainment venue by locating his spatiotemporal information to understand his health status, preferences, time, and scope of the outing [8]. Therefore, in a spatiotemporal crowdsourcing application, it is especially important to protect the spatiotemporal information of participants. The privacy protection technology based on spatiotemporal crowdsourcing has also become a research hotspot in the field of spatiotemporal crowdsourcing systems [9].

In order to ensure that participants' spatiotemporal private information is not leaked when publishing data, a large amount of work on spatiotemporal privacy protection is devoted to disturbing and anonymizing the spatiotemporal data that may reveal personal whereabouts. To et al. [10] proposed a protection framework based on differential privacy. The workers in spatiotemporal crowdsourcing first submit their real location information to the truthful mobile service provider, and the mobile service provider uses a grid-based method to construct the private spatial decompositions (PSDs) for the original location information and adds Laplace noise to process workers' real location data for privacy protection purposes. Vu et al. [11] proposed a privacy protection mechanism based on local sensitive hashing to group participants' positions. Each group contains at least k participants to achieve spatial anonymity. The ideal partition of spatial data under low time complexity is realized, and participants' location information is protected in a spatiotemporal crowdsourcing application scenario. In [12], the problem of insufficient diversity of k-anonymity algorithm to the participants' sensitive locations is solved, and the probability of participants' access to the sensitive locations is limited or the probability analysis based on adversary knowledge is used to ensure the location diversity.

However, most researchers currently only consider the data publishing in static scenario. The attackers use historical publishing results to reveal sensitive information during static publishing, for example, to compare with the results of the previous publishing. In spatiotemporal crowdsourcing applications, many data analysis applications actually involve dynamic data publishing. For example, in order to plan travel routes for special vehicles (cash trucks, ambulances, fire engines, etc.), it is necessary to issue a sensing task to collect road traffic jams [13, 14]. For such a spatiotemporal sensitive task, application server needs to dynamically publish sensing data submitted by participants to improve the timeliness of the task. Dynamic sensing data change constantly over time, so it is often necessary to anonymize and dynamically publish sensing data at different times. However, most anonymity algorithms are invalid when dealing with the dynamic publishing of spatiotemporal data [15]. The previous anonymity result cannot be effectively utilized. Because of the big data scenario, the time complexity of algorithms is high, and the timeliness is poor [16]. Moreover, most researchers proposed privacy preserving for participant's location information but failed to consider that attackers can also infer other private information based on participant's spatiotemporal information. According to these problems, we research the privacy protection for spatiotemporal privacy information in spatiotemporal crowdsourcing systems, and the following issues should be improved further:

(1) In the process of dynamic data publishing, the results after anonymization should be effectively utilized instead of unifying the anonymization of incremental data with previous data to improve the timeliness of dynamic publishing of big data

(2) In the process of anonymizing the location attribute of participants, the time attribute is added to effectively avoid the background knowledge attack and homogeneity attack against the location attribute

In order to solve the above problems, we propose a spatiotemporal privacy protection method for spatiotemporal crowdsourcing systems. The contributions of this paper are shown as follows:

(1) A dynamic publishing algorithm based on spatiotemporal data privacy protection is designed by improving k-anonymity. When incremental data arrive, the anonymization result of the last time will be utilized to solve the timeliness problem of dynamic publishing.

(2) Based on the traditional position coordinate, a time axis is added to form the spatiotemporal information of participants, and the anonymization of participants' spatiotemporal information is carried out by applying k-anonymity and l-diversity methods, so as to solve the background knowledge attack and homogeneity attack problems.

(3) In order to verify the effectiveness of the proposed privacy protection method, the comparison experiments with k-anonymity and variable centroid location aggregation (VCLA) [17] algorithms are conducted on two real-world datasets.

The structure of the paper is as follows. Section 2 introduces the related works of spatiotemporal privacy protection. Section 3 introduces the proposed spatiotemporal privacy protection method for spatiotemporal crowdsourcing systems. In Section 4, the real-world datasets and the existing anonymity algorithms are used for evaluating the performance of the proposed method. Section 5 concludes the paper and presents the future work.

2. Related Works

In this section, we will introduce the related works about privacy protection methods for spatiotemporal data and dynamic publishing of sensitive data under a participatory sensing environment. Participatory sensing (PS) refers to the formation of a mobile Internet through daily mobile devices, where data are sensed, collected, analyzed, or screened by the public and professional users and then uploaded to the participatory sensing network [18]. With the popularization of mobile terminals and the rapid development of wireless sensor technology, the application of PS is becoming more and more common in real life. For example, in [19], Chen et al. studied the energy-efficient task offloading in mobile edge computing (MEC). However, in the process of task offloading, the privacy of participants will be exposed. In order to deal with the problems that participants' privacy will be exposed during the task offloading process, Xu et al. [20] put forward a two-phase offloading optimization strategy for joint optimization of offloading utility and privacy in edge computing. Further, Xu et al. [21] discussed the problem that transmitted information is vulnerable to attack and may cause incomplete data during task offloading. A blockchain-enabled computation offloading method was proposed to ensure data integrity. In the implementation process of these participatory sensing applications, sensing tasks uploaded by participants will mark personal spatiotemporal data, which brings great risks to the privacy security and personal safety of participants. Therefore, while people enjoy the convenience brought by LBS, their privacy is also at risk of being exposed [22].

In LBS, using anonymous technology to solve the location privacy problem of participants has been widely studied [23]. The k-anonymity technology was firstly proposed by Samarati and Sweeney [24]. The parameter k specifies the maximum risk of information disclosure that users can bear. It requires at least k indistinguishable records on the quasi-identifier in published data, so that attackers cannot identify the specific individual that the privacy information belongs to, so as to protect personal privacy. In

[25], the clustering-based k-anonymity strategy is adopted to protect the privacy disclosure of wearable owners when they upload sensing data. In [26], a k-anonymous location privacy protection method based on coordinate transformation was proposed for the problem that the third-party truthful server (TTP) was often untruthful in real life [27]. The anonymous server receives the coordinate-converted participant location and constructs an anonymous area without knowing the user's actual location, thereby protecting the participant's location privacy. In [28], the optimal k value of the current user is determined according to the user's environment and social attributes, and a location protection k-anonymity method based on the truthful chain was proposed to protect the location privacy of participants while ensuring the quality of service.

However, k-anonymity cannot cope with the background knowledge attack and homogeneity attack. Machanavajjhala et al. [29] firstly proposed l-diversity to improve k-anonymity. Each k-anonymity group in the published data sheet contains at least l different sensitive attribute values, so that the probability that an attacker infers a certain record privacy information will be less than $1/l$. In [30], considering the identity attributes of participants, it is ensured that each anonymous set at least has $k - 1$ participants, and each anonymous set has p different sensitive values. In [31], k-anonymity and l-diversity were adopted as privacy models, and an anonymization method based on genetic algorithm clustering was proposed. The basic operator of genetic algorithm is improved to protect the personal sensitive information contained in the published report.

However, when requesting LBS services, the location of most participants is always related to time [32]. The above works only protect the location attribute of participants but do not associate the location attribute of participants with the time attribute. Trajectory anonymity refers to the sequence of user location information in a continuous period of time, which anonymizes and protects the user's location attribute and time attribute together. In [33], the trajectory privacy protection method based on user demand was proposed. By dividing different time intervals and setting different privacy protection parameters for different trajectories, the anonymous trajectory equivalence class is constructed. In [34], the Hilbert curve was used to extract the distribution characteristics of trajectory data each time, and the personalized differential privacy publishing mechanism was designed according to the individual needs for different degrees of privacy. In [35], a collaborative trajectory privacy protection scheme for continuous query was proposed to confuse attackers by issuing false query, thus confusing users' actual trajectory. In [36], a trajectory privacy protection algorithm based on trajectory shape diversity was proposed by combining k-anonymity and l-diversity to solve the trajectory privacy leakage problem that may be caused by the high similarity between trajectories in the anonymous set.

In the research of privacy protection data publishing (PPDP), the first proposed model was mainly used for static publishing, that is, only considering the one-time publishing of data, and the above research was mainly conducted for the static data publishing [37]. However, in many spatiotemporal crowdsourcing applications, a large amount of data stays in a changing state, and dynamic data publishing occurs from time to time [38]. In order to solve the problem that static publishing cannot resist link attacks and critical missing attacks, Wang and Fung [39] firstly studied the possible privacy leaks of data redistribution and proposed a method to prevent privacy leakage. The main idea of this method is to hide the true connection relationship between the two publishing versions, thereby weakening the global quasi-identifier. Xiao [40] firstly proposed the privacy protection model m-invariance for dynamic data publishing, whose key is to introduce pseudogeneralization technology to ensure that any QI group records in different data publishing versions have the same sensitive attribute value. In [41], because of the problem that the privacy protection association rule mining algorithm is not applicable to the dynamic change database, the incremental privacy protection data mining algorithm based on granularity calculation was proposed, and the incremental update algorithm was used to solve the problem of frequent item set calculation of incremental transaction database. In [42], a differential privacy histogram publishing method based on fractal dimension mining technology was proposed. The method used fractal dimension to cluster datasets and counted the values of each class. Laplace noise was added to data before publishing to achieve differential privacy. However, the above methods cannot cope with the privacy protection for spatiotemporal information, and it is difficult to adapt to the issue of dynamic privacy protection in spatiotemporal crowdsourcing applications. Even if the above methods consider participants' location information, attackers could also infer participants' privacy through time information. More importantly, the above methods are invalid for real-time data tasks.

Based on the above discussions, a dynamic publishing method for spatiotemporal privacy protection under the participatory sensing environment is proposed. By combining k-anonymity and l-diversity, the proposed dynamic publishing method could protect the privacy information of participants and reduce the information loss.

3. Dynamic Privacy Protection Algorithm

In this section, a dynamic privacy protection mechanism for spatiotemporal sensitive information is researched, and the working principles of the three main parts of the dynamic privacy protection mechanism are introduced. The proposed algorithms and corresponding explanations are given through an example.

3.1. Dynamic Privacy Protection Mechanism.

The mechanism is divided into three parts: participants, TTPs, and application server.

(i) Participants: in spatiotemporal crowdsourcing applications, participants are responsible for the collection and uploading of sensing data [43]. Sensing data uploaded by participant p_i, $1 \le i \le n$, and p_i include following attributes: $< \text{id}_i, \text{data}_i, \text{time}_i, x_i, y_i >$, where id_i is the identity attribute of p_i, data_imeans completed sensing tasks uploaded by p_i, and $< \text{time}_i, x_i, y_i >$ indicates real-time attribute and location attribute contained in data_i, denoted by d_i. It is a sensitive attribute and requires to anonymize. In the dynamic privacy protection publishing mechanism, participants submit sensing data in batches.

(ii) TTPs: in this mechanism, participants firstly upload sensing data to TTPs, and TTPs preprocesses the sensing data to extract sensitive data (i.e., participants' real spatiotemporal data) [44]. Then, using k-anonymity, the real spatiotemporal data are anonymized. The sensing data that do not satisfy the anonymity condition are stored in buffer pool and anonymize with the next incremental data. More importantly, when incremental data arrive, the corresponding equivalence classes will be added if the adaptive threshold is satisfied by utilizing the previous anonymity results. Finally, the cluster center value $\overline{u_i}$, $1 \le i \le r$, is sent to the application server.

(iii) Application server: for avoiding the background knowledge attack and the homogeneity attack against k-anonymity, cluster center values need to be clustered again based on l-diversity idea. Application server anonymizes u_i according to the time attribute. Each cluster contains at least l cluster center values, and then the newly generated cluster center value $\overline{t_i}$, $1 \le i \le c$, is published. After anonymization processing on TTPs and application server, the results are shown in Table 1, where $U = U_{i=1}^{r} M_i$, $M_i = U_{j=1}^{r_i} u_{ij}$, $L = U_{i=1}^{c} T_i$, and $T_i = U_{j=1}^{c_i} t_{ij}$, both r and c, respectively, represent the number of position clusters and time clusters.u_i represents a cluster containing spatiotemporal sensitive data. r_i and c_i represent the number of spatiotemporal sensitive data in the ith cluster. u_{ij} andt_{ij} represent the spatiotemporal sensitive data included in a cluster.

The dynamic publishing privacy protection mechanism proposed in this paper is different from the traditional spatiotemporal crowdsourcing process. In the process of traditional spatiotemporal crowdsourcing, requesters firstly publish tasks and then recruit participants to complete the task. In the process of uploading tasks by the participants, traditional spatiotemporal crowdsourcing does not consider the privacy of participants. More importantly, the static publish of tasks will reduce users' experience. The working process of the proposed dynamic publishing privacy protection mechanism is shown in Figure 1. In Step 1, participants send collected sensing data (including spatiotemporal sensitive information) to TTPs by secure wireless networks. In Step 2, TTPs reprocess the sensing data and extract spatiotemporal sensitive data. The spatiotemporal sensitive data are used by k-anonymity to anonymize. If the clustering condition is not met, Step 3 is performed to temporarily store the corresponding spatiotemporal sensitive data into buffer pool. If the clustering condition is met, Step 4 is performed, and TTP sends the anonymity result to the application server. In Step 5, application server clusters based on l-diversity for the time attribute of anonymity results. In Step 6, application server publishes the sensing data containing anonymity sensitive spatiotemporal data. In Step 7, when other participants submit sensing data, incremental data are sent to TTPs together with the sensing data temporarily stored in buffer pool. In Step 8, sensing data are dynamically anonymized by utilizing the previous anonymity results. Perform the above process until participants no longer submit sensing data.

3.2. Static Publishing Anonymous Protection.

3.2. Static Publishing Anonymous Protection. In order to protect the spatiotemporal privacy of participants, k-anonymization is used to anonymize participants' time and location attributes together. In the spatiotemporal crowdsourcing application, because of the different dimensions of time and location attributes of participants, we standardize the spatiotemporal sensitive data d_i, $1 \le i \le n$, by using the standard deviation method expressed by equation (1). d_{ik} represents the real spatiotemporal information of the kth dimension of the ith data. d'_{ik} represents normalized spatiotemporal information of d_{ik} that is shown by the following equation:

$$d'_{ik} = \frac{d_{ik} - \min\{d_{jk}\}}{\max\{d_{jk}\} - \min\{d_{jk}\}}, \quad 10 \le j \le n, 1 \le k \le 3. \tag{1}$$

The distance between participants p_i and p_j is calculated by equation (2). The distance includes spatial distance and temporal distance between participants p_i and p_j:

$$\text{dis}(p_i, p_j) = \sqrt{\sum_{k=1}^{3} \left(d'_{ik} - d'_{jk}\right)^2}, \quad 1 \le i, j \le n. \tag{2}$$

In order to easily find the center points of position cluster and time cluster, we calculate the global centroid $\overline{d}$ of the actual spatiotemporal dataset for anonymization by the following equation:

$$\overline{d} = < \frac{\sum_{i=1}^{n} \text{time}_i}{n}, \frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n} > \tag{3}$$

In order to reduce the information loss and increase the privacy protection, we set the adaptive threshold expressed as follows:

$$\text{ave}_j = \sum_{k=1}^{3} \sum_{i=1}^{r} \left(d'_{ik} - \overline{d'_{jk}}\right), \quad 1 \le j \le |G| \text{ or } |L|, \tag{4}$$

where r indicates that there is r spatiotemporal data in the cluster. The static publishing anonymity protection based on k-anonymity is shown in Algorithm 1.

Algorithm 1 describes that participants send sensing data to TTPs. The TTPs firstly process the sensing data and extract participant's real spatiotemporal information (represented by set A) as sensitive data for privacy

Table 1: Anonymization results.

Anonymization results	TTPs	Application server
Clustering result set	U	L
Each cluster in result set	$M_i, 1 \le i \le r$	$T_i, 1 \le i \le c$
Spatiotemporal sensitive data included in a cluster	$u_{ij}, 1 \le j \le r_i$	$t_{ij}, 1 \le j \le c_i$
Cluster center value	$\overline{u_i}, 1 \le i \le r$	$\overline{t_i}, 1 \le i \le c$

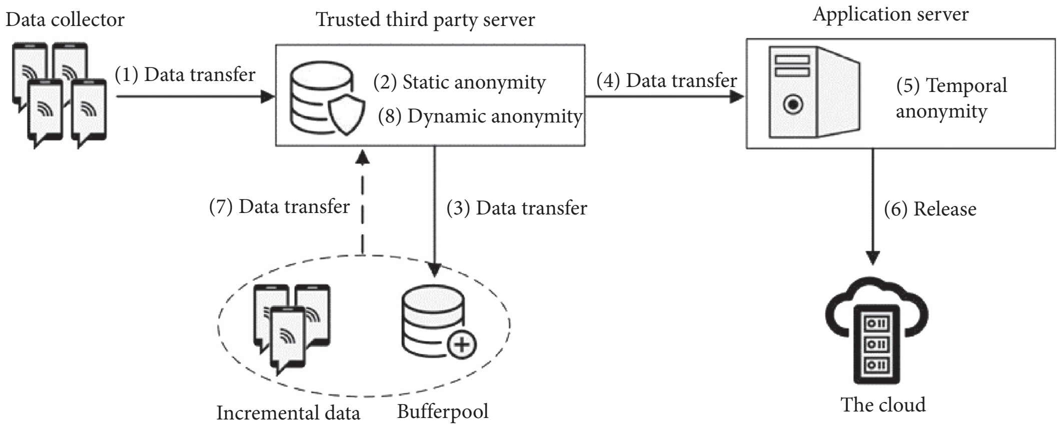


Figure 1: The framework of a dynamic publishing mechanism for privacy protection.

protection. The input of Algorithm 1 is k-anonymity-specified parameter k and the participant's real spatiotemporal dataset A. The output of Algorithm 1 is the anonymity result set U and buffer pool dataset B. Calculate the global centroid $\overline{d}$ in Step 1. Step 2 initializes parameters, and count is the number of new split clusters. Steps 4–11 describe that the number of points in the new split cluster is k. Step 5 selects a point d_{sma} with the largest distance to the global center point $\overline{d}$. d_{sma} indicates the new cluster center point and will be deleted from A (Step 6). Steps 7–11 select $k-1$points that have the smallest distance with $d_{\text{sma}}$to form a new cluster N_{count}. Update the center point d_{count} of N_{count} (Step 8), and select the point d_{smi} with the smallest distance to d_{count}(Step 9). Step 10 adds d_{smi} to cluster N_{count} and removes it from A. Steps 12–19 extend the cluster N_{count}, and in order to reduce information loss, we set the adaptive threshold *ave* (Step 13). If the point d_{cmi} (Step 14) in A satisfies the adaptive threshold, it will be added to the cluster N_{count} (Step 16), and the centroid d_{count} of N_{count} (Step 17) is updated. In Step 20, N_{count} is added to U, and the number of cluster is updated. If there are remaining data in A, it is stored in the buffer pool B (Step 22). In Step 23, the output of Algorithm 1 is returned.

The real spatiotemporal information contained in the sensing data uploaded by participants is anonymized by TTPs and returned to anonymity result set U. Send the center point $\overline{u_i}$ to the application server. Then, we illustrate the anonymity process of spatiotemporal data more visually by some data examples in experiments. As shown in Table 2, the first column is the class ID being run by Algorithm 1, the second and third columns are the real spatiotemporal information of participants, and the fourth and fifth columns are the anonymity spatiotemporal information. We can see that each equivalence class contains at least 3 points.

3.3. Improved Static Publishing Anonymity Protection Based on l-Diversity.

However, k-anonymity is vulnerable to background knowledge attack and homogeneity attack. Therefore, when the application server publishes anonymity results, we adopt l-diversity to improve the algorithm. Application server receives the cluster center value u_i sent by TTPs, anonymizes the time attribute based on l-diversity, and calculates the time center value by the following equation:

$$\overline{t} = \frac{\sum_{i=1}^{m} \text{time}_i}{m}, \tag{5}$$

where m refers to the number of spatiotemporal data anonymized by TTPs, i.e., $m = \sum_{i=1}^{r} |M_i|$.

Algorithm 2 describes the anonymous releasing based on l-diversity for time attribute. The input is l-diversity parameter l and the output set U of Algorithm 1, and the output is anonymous set L. Step 1 and Step 2 take the time set T and the position set O out of U, respectively. Step 3 calculates the global central value $\overline{t}$ of time set T, and the number of initial clusters is count = 1. Steps 4–15 describe that the number of points in the newly generated cluster is l. Step 5 initializes time cluster T_{count} and location cluster O_{count}. Step 6 finds the t_{tma} with the largest distance to the global central value $\overline{t}$ by equation (2). Step 7 adds t_{tma} to the new cluster T_{count}, and the coordinate cluster O_{tma}corresponding to the subscript t_{ma} is added to the new cluster O_{count}. Then, t_{tma} is deleted from the time set T. The $l-1$ points with the smallest distance are selected to join the cluster (steps 8–12). Step 13 updates the time center

```
Input: $k$-anonymous parameter $k$, the actual spatiotemporal dataset $A$ from participants
Output: aggregation result $U$, buffer pool dataset $B$
Calculate the global centroid $\overline{d}$ of $A$ by equation (3)
count = 1, $U = \varphi$
while $|A| \geq k$ do
    $N_{\text{count}} = \varphi$
    sma = $\text{argmax}_{i \in A} \text{dis}(d_i, \overline{d})$
    $N_{\text{count}} = N_{\text{count}} \cup d_{\text{sma}}$, $A = A/d_{\text{sma}}$
    for $j \longleftarrow 1$ to $k - 1$ do
        Update the centroid $\overline{d_{\text{count}}}$ of $N_{\text{count}}$ by equation (3)
        smi = $\text{argmin}_{i \in A} \text{dis}(d_i, \overline{d_{\text{count}}})$
        $N_{\text{count}} = N_{\text{count}} \cup d_{\text{smi}}$, $A = A/d_{\text{smi}}$
    end for
    while $|N_{\text{count}}| < 2k - 1$ do
        Calculate the average distance $\text{ave}_{\text{count}}$ of $N_{\text{count}}$ by equation (4)
        cmi = $\text{argmin}_{i \in A} \text{dis}(d_i, \overline{d_{\text{count}}})$
        if $\text{dis}(d_{\text{cmi}}, \overline{d_{\text{count}}}) < \text{ave}_{\text{count}}$ then
            $N_{\text{count}} = N_{\text{count}} \cup d_{\text{cmi}}$, $A = A/d_{\text{cmi}}$
            Update the centroid $d_{\text{count}}$ of $N_{\text{count}}$ by equation (3)
        end if
    end while
    $U = U \cup N_{\text{count}}$, count = count + 1
end while
$B = \text{A}$
return $U$, $B$
```

Algorithm 1: Static publishing anonymity protection based on k-anonymity.

Table 2: Three anonymous examples.

Class ID	Time	Location	Anonymized time	Anonymized location
1	20:47:36	(21.3675, −157.9388)	20:38:22	(21.3166, −157.8616)
1	21:46:33	(21.2866, −157.8129)	20:38:22	(21.3166, −157.8616)
1	19:20:57	(21.2958, −157.8331)	20:38:22	(21.3166, −157.8616)
2	23:31:00	(45.5894, −122.7524)	23:00:57	(46.3272, −122.5448)
2	22:00:00	(45.7801, −122.5400)	23:00:57	(46.3272, −122.5448)
2	23:31:50	(47.6122, −122.3419)	23:00:57	(46.3272, −122.5448)
3	18:54:05	(30.4810, −97.8295)	19:15:09	(32.0273, −97.4996)
3	19:33:26	(32.7368, −97.3271)	19:15:09	(32.0273, −97.4996)
3	19:17:48	(32.8640, −97.3421)	19:15:09	(32.0273, −97.4996)
4	18:46:48	(30.2016, −97.6671)	19:02:11	(31.5155, −97.4498)
4	19:09:06	(32.6804, −97.3746)	19:02:11	(31.5155, −97.4498)
4	19:03:47	(32.8382, −97.0045)	19:02:11	(31.5155, −97.4498)
4	19:09:03	(30.3417, −97.7530)	19:02:11	(31.5155, −97.4498)
5	19:50:21	(59.3238, 18.0977)	17:25:48	(59.3232, 18.0543)
5	15:49:08	(59.3457, 18.0587)	17:25:48	(59.3232, 18.0543)
5	16:38:04	(59.3055, 17.9892)	17:25:48	(59.3232, 18.0543)
5	17:28:22	(59.3122, 18.0796)	17:25:48	(59.3232, 18.0543)
5	17:23:05	(59.3288, 18.0461)	17:25:48	(59.3232, 18.0543)

value $\overline{t_{\text{count}}}$ of cluster T_{count}. The output of Algorithm 2 in Step 14 is L_{count}, and the Cartesian product of time center value sets $\overline{t_{\text{count}}}$ and position cluster O_{count}. Steps 16–23 describe that if there is any remaining point in time set T, the cluster with the smallest distance (Step 18) is found by equation (2) and added to the cluster (Step 19), then the time center value $\overline{t_{\text{lmi}}}$ of the cluster T_{lmi} is updated (Step 20). In Step 24, the output of Algorithm 2 is returned.

The following is a more visual illustration of releasing spatiotemporal data based on l-diversity. As shown in Table 3, the first column is the group ID being run by Algorithm 2, the second column is the class ID being run by Algorithm 1, the third and fourth columns are the anonymous spatiotemporal information being run by Algorithm 1, and the fifth and sixth columns are the anonymous spatiotemporal information anonymized by the application server. We can see that each 2-equivalence group ($l = 2$) contains at least two 3-equivalence classes ($k = 3$), where the anonymous time attribute is the same and the anonymous location attribute is different.

```
Input: $l$-diversity parameter $l$, aggregation result $U$ from Algorithm 1
Output: aggregation result $L$
Take time set $T$ out of $U$
Take location set $O$ out of $U$
Calculate the global centroid $\bar{t}$ by equation (5), count = 1
while $|T| \geq l$do
    $T_{\text{count}} = \varphi$, $O_{\text{count}} = \varphi$
    $\text{tma} = \text{argmax}_{i \in T} \text{dis}(t_i, \bar{t})$
    $T_{\text{count}} = T_{\text{count}} \cup t_{\text{tma}}$, $O_{\text{count}} = O_{\text{count}} \cup O_{\text{tma}}$, $T = T / t_{\text{tma}}$
    for $j \longleftarrow 1$ to $l - 1$ do
        Update the centroid $\overline{t_{\text{count}}}$ of $T_{\text{count}}$ by equation (3)
        $\text{tmi} = \text{argmin}_{i \in T} \text{dis}(t_i, \overline{t_{\text{count}}})$
        $T_{\text{count}} = T_{\text{count}} \cup t_{\text{tmi}}$, $O_{\text{count}} = O_{\text{count}} \cup O_{\text{tmi}}$, $T = T / t_{\text{tmi}}$
    end for
    Update the centroid $\overline{t_{\text{count}}}$ of $T_{\text{count}}$ by equation (3)
    $L_{\text{count}} = \overline{t_{\text{count}}} \times O_{\text{count}}$, count = count + 1
end while
while $|T| > 0$do
    for $i \in |T|$do
        $\text{lmi} = \text{argmin}_{j \in L} \text{dis}(t_i, \overline{t_j})$
        $T_{\text{lmi}} = T_{\text{lmi}} \cup t_i$, $O_{\text{lmi}} = O_{\text{lmi}} \cup O_i$
        Update the centroid $\overline{t_{\text{lmi}}}$of $T_{\text{lmi}}$
        $L_{\text{lmi}} = \overline{t_{\text{lmi}}} \times O_{\text{lmi}}$
    end for
end while
return $L$
```

Algorithm 2: Static publishing anonymity protection based on l-diversity.

Table 3: 3-Anonymity, 2-diversity examples.

Group ID	Class ID	Time	Location	Anonymized time	Anonymized location
1	3	19:15:09	(32.0273, −97.4996)	18:34:23	(32.0273, −97.4996)
1	3	19:15:09	(32.0273, −97.4996)	18:34:23	(32.0273, −97.4996)
1	3	19:15:09	(32.0273, −97.4996)	18:34:23	(32.0273, −97.4996)
1	4	19:02:11	(31.5155, −97.4498)	18:34:23	(31.5155, −97.4498)
1	4	19:02:11	(31.5155, −97.4498)	18:34:23	(31.5155, −97.4498)
1	4	19:02:11	(31.5155, −97.4498)	18:34:23	(31.5155, −97.4498)
1	4	19:02:11	(31.5155, −97.4498)	18:34:23	(31.5155, −97.4498)
1	5	17:25:48	(59.3232, 18.0543)	18:34:23	(59.3232, 18.0543)
1	5	17:25:48	(59.3232, 18.0543)	18:34:23	(59.3232, 18.0543)
1	5	17:25:48	(59.3232, 18.0543)	18:34:23	(59.3232, 18.0543)
1	5	17:25:48	(59.3232, 18.0543)	18:34:23	(59.3232, 18.0543)
1	5	17:25:48	(59.3232, 18.0543)	18:34:23	(59.3232, 18.0543)
2	1	20:38:22	(21.3166, −157.8616)	21:49:40	(21.3166, −157.8616)
2	1	20:38:22	(21.3166, −157.8616)	21:49:40	(21.3166, −157.8616)
2	1	20:38:22	(21.3166, −157.8616)	21:49:40	(21.3166, −157.8616)
2	2	23:00:57	(46.3272, −122.5448)	21:49:40	(46.3272, −122.5448)
2	2	23:00:57	(46.3272, −122.5448)	21:49:40	(46.3272, −122.5448)
2	2	23:00:57	(46.3272, −122.5448)	21:49:40	(46.3272, −122.5448)

3.4. Dynamic Publishing Anonymity Protection. For static one-release mechanisms, k-anonymity and l-diversity are valid. However, in real life, application servers usually publish sensing data dynamically. Therefore, in this section, we improve k-anonymity and l-diversity to accommodate dynamic publishing mechanism. Algorithm 3 describes the dynamic publishing anonymity protection.

Algorithm 3 describes how TTPs use the previous anonymity result to solve the problem of dynamic publishing when participants submit sensing data in different time periods. The input of Algorithm 3 is k-anonymous parameter k, the clustering result U of Algorithm 1, incremental dataset I (that is, the sensing data submitted by participants), and buffer pool dataset B. The output of the algorithm is the clustering result D and buffer pool dataset

```
     Input: $k$-anonymous parameter $k$, aggregation result $U$ from Algorithm 1, incremental dataset $I$, buffer pool dataset $B$
     Output: aggregation result $D$, buffer pool dataset $B'$
(1)  Calculate global dataset $W=I+B$
(2)  for $i \longleftarrow 1$ to $r$ do
(3)    Take the centroid set $\overline{U}$ out of $U$
(4)    $\text{smi} = \text{argmin}_{j\in\overline{U}}\text{dis}(\overline{u_j}, tW_i)$
(5)    $r\text{ ave} = \text{argmin}_{e\in|u_{\text{smi}}|}\text{dis}(u_e, \overline{u_{\text{smi}}})$
(6)    if $\text{dis}(\overline{u_{\text{smi}}}, tW_i) \leq r\text{ ave}$ then
(7)      $M_{\text{smi}} = M_{\text{smi}} \cup W_i$, $W = W/W_i$
(8)      Update the centroid $\overline{u_{\text{smi}}}$ of $u_{\text{smi}}$
(9)      $D = U$
(10)   end if
(11) end for
(12) for $i \longleftarrow 1$to $r$ do
(13)   if $|M_i| > 2k$ then
(14)     Callback Algorithm 4$\longrightarrow$
(15)       input: $k$-anonymous parameter $k$, cluster $M$
(16)       output: aggregation result $G$
(17)   end if
(18) $D = U \cup G$, $B' = W$
(19) end for
(20) return $D$, $B'$
```

Algorithm 3: Dynamic publishing anonymous protection.

B'. The global dataset W is the incremental data I and the buffer pool data B (Step 1). Steps 2–11 describe the process of adding data from dataset W that meets the adaptive threshold condition to the last clustering result, where r represents the number of clusters of U (Table 1). First, the cluster center set $\overline{U} = U_{i=1}^{r}\overline{u_i}$ in the clustering result U is taken out (Step 3), and Step 4 finds the subscript of the smallest cluster center point smi to point W_i. Then, adaptive threshold values r ave are set by equation (4) (Steps 5 and 6), where r ave is the average distance between point u_e and center point $\overline{u_{\text{smi}}}$ in clusteru_{smi}. If the point in dataset W meets the adaptive threshold, join the corresponding cluster and delete the point from W (Step 7), update the central value $\overline{u_{\text{smi}}}$ of cluster u_{smi} (Step 8), and assign the updated U to the output result D in Algorithm 3 (Step 9). Steps 12–19 describe that if the number of points in cluster M_i is greater than or equal to $2k$, then Algorithm 4 is called to split M_i. The clustering result D is denoted by $D = U \cup G$, where U is the number of points in cluster $M_i$less than $2k$, and G is the output of Algorithm 4 and temporarily stores the remaining data in W to buffer pool B'(Step 18). In Step 20, the output of Algorithm 3 is returned.

Algorithm 4 describes that if the number of points in the cluster is greater than or equal to $2k$, the cluster is split. The input of Algorithm 4 is k-anonymous parameter k and cluster M. The output of Algorithm 4 is the clustering result G of the new split. Step 1 calculates the global center point $\overline{d}$ of cluster M by equation (3). Step 2 initializes parameters, count is the number of new split clusters, and G is the output of Algorithm 4. Steps 3–13 describe that the number of points in the new split cluster is k. Step 5 selects a point $\underline{d_{\text{sma}}}$ with the largest distance to the global center point $\overline{d}$. Take d_{sma} as the new cluster center point and delete it from M (Step 6). Steps 7–11 select k-1 points that have the smallest distance with $\underline{d_{\text{sma}}}$ to form a new clusterN_{count}, update the center point $\overline{d_{\text{count}}}$of N_{count} (Step 8), and select the point $d_{\text{smi}}$with the smallest distance to $\overline{d_{\text{count}}}$ (Step 9). Step 10 adds d_{smi} to cluster N_{count} and removes it from M. In Step 12, the newly generated cluster N_{count} is added to the output result G, and the number of clusters increases. If there are remaining points in cluster M, add them to the new cluster closest to them (steps 14–20). Step 16 finds a cluster N_{cmi} having the smallest distance with the remaining point d_i, add $d_i$to N_{cmi} (Step 17), and update the center point $\overline{d_{\text{cmi}}}$ of cluster N_{cmi}. In Step 21, the output G of Algorithm 4 is returned.

4. Experiments and Result Analysis

In this section, we use real-world datasets, including Gowalla's Friendship Network dataset and Kaggle's New York Taxi Travel Time dataset. Table 4 shows the number of attributes and data points and the density of data points contained in datasets. We compare the proposed STPP algorithm with k-anonymity and VCLA algorithms in terms of running time, information loss, and privacy protection. The hardware environment of the experiments is an AMD A8-5550M APU with Radeon (tm) HD Graphics @ 2.10 GHz equipped with 4 GB RAM and running the Win 10 OS.

Datasets are processed to better protect participants' spatiotemporal sensitive data. First, we randomly extract 1000 data from Friendship Network dataset as a segment, a total of five segments, as participant's sensing data to conduct comparison experiments. Then, we randomly extract 3000 data from New York City Taxi Trip dataset as a

```
Input: $k$-anonymous parameter $k$, cluster $M$
Output: aggregation result $G$
Calculate the global centroid $\overline{d}$ of cluster $M$ by equation (3)
count = 1, $G = \varphi$
while $|M| \geq k$ do
    $N_{\text{count}} = \varphi$
    sma = $\text{argmax}_{i\in M}\text{dis}(d_i, \overline{d})$
    $N_{\text{count}} = N_{\text{count}} \cup d_{\text{sma}}$, $M = M/d_{\text{sma}}$
    for $j \longleftarrow 1$ to $k-1$ do
        Update the centroid $\overline{d_{\text{count}}}$ of $N_{\text{count}}$
        smi = $\text{argmin}_{i\in M}\text{dis}(d_i, \overline{d_{\text{count}}})$
        $N_{\text{count}} = N_{\text{count}} \cup d_{\text{smi}}$, $M = M/d_{\text{smi}}$
    end for
    $G = G \cup N_{\text{count}}$, count = count + 1
end while
while $|M| > 0$do
    for $i \in |M|$do
        cmi = $\text{argmin}_{i\in G}\text{dis}(d_i, \overline{d_j})$
        $N_{\text{cma}} = N_{\text{cmi}} \cup d_i$
        Update the centroid $\overline{d_{\text{cmi}}}$ of $N_{\text{cmi}}$
    end for
end while
return G
```

Algorithm 4: Breaking up clusters.

Table 4: Attribute, quantity, and density of datasets.

Datasets	Dimensions	Quantity	Sparseness
Friendship Network dataset	5	6442892	Sparse
New York City Taxi Trip dataset	11	1458644	Dense

segment, a total of five segments, as participant's sensing data to design comparison experiments. Each segment of sensing data is uploaded to TTPs in batches dynamically. Then, the spatiotemporal sensitive data of participants, including time and location attributes, are extracted from sensing data for anonymization.

Figure 2 shows the comparison of experimental results by comparing the proposed STPP algorithm with k-anonymous and VCLA algorithms on running time. Figure 2(a) shows the experimental result on Friendship Network dataset, and Figure 2(b) shows the experimental result on New York City Taxi Trip dataset. The x-coordinate is the number of participants, and the y-coordinate is running time. It can be seen that the STPP algorithm is superior to the other two algorithms, whether it is on a small dataset where participants' spatiotemporal distance is sparse, or on a large dataset with dense spatiotemporal distance. When there are fewer participants submitting tasks, the running time of the three algorithms is not much different. It is because that the three algorithms are improved by k-anonymity algorithms, the STPP algorithm proposed in this paper does not have obvious advantages in terms of running time when there are few participants. However, when the number of participants gradually increases, STPP algorithm could better solve the problem of poor timeliness of data publishing due to the large number of participants in spatiotemporal crowdsourcing applications.

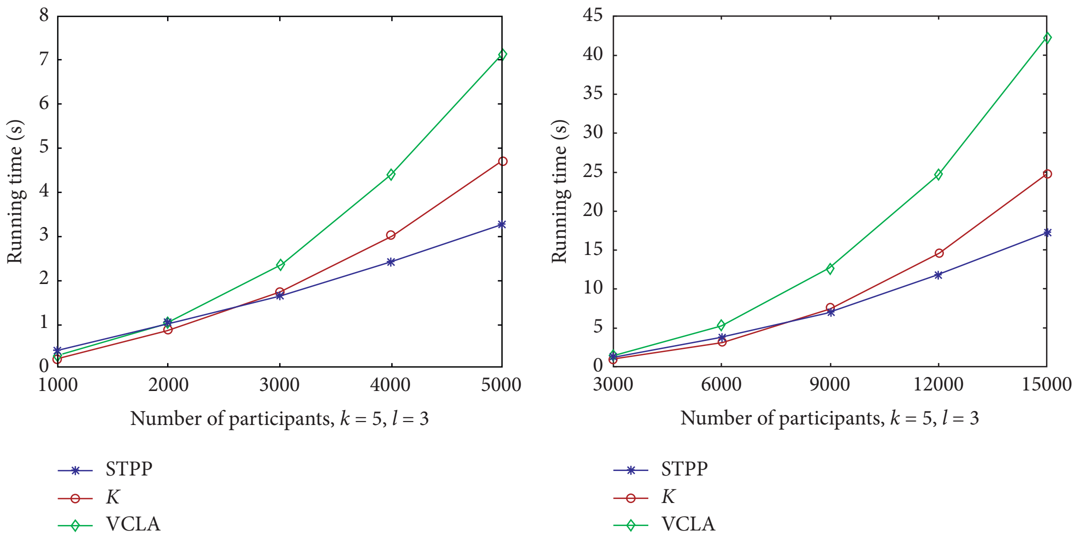
Since anonymized data are used for dynamic publishing, the difference between real spatiotemporal data and anonymized data is seen as the information loss. The information loss is expressed by the following equation:

$$\text{IL} = \sum_{i=1}^{n}\sum_{k=1}^{3} d'_{ik} - \overline{d'_{ik}}, \quad (6)$$

where $\overline{d'_{ik}}$ represents the anonymized information of d'_{ik}. k represents dimension, which includes time dimension and location dimension.

Figure 3 shows the comparison of experimental result by comparing the STPP algorithm with k-anonymous and VCLA algorithm on information loss. The x-coordinate is the number of participants on the Friendship Network dataset, and the y-coordinate is information loss. From the experimental result, it can be seen that the information loss increases with the increase of participants. Moreover, STPP algorithm is obviously better than the comparison algorithms on information loss.

Figure 4 shows the relationship between the parameter k of k-anonymous and the information loss of the STPP algorithm, where different curves represent different amounts of data. The experiments are conducted on the Friendship Network dataset. From the experimental result, it can be inferred that with the increase of k, the information loss increases gradually, which is because that increasing k leads



(a)

(b)

Figure 2: Comparison of experimental results of running time on (a) Friendship Network dataset and (b) New York City Taxi Trip dataset.

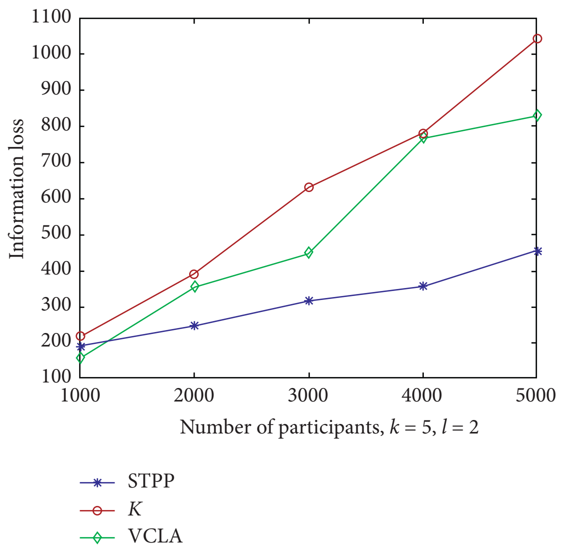


Figure 3: Comparison of experimental result on information loss.

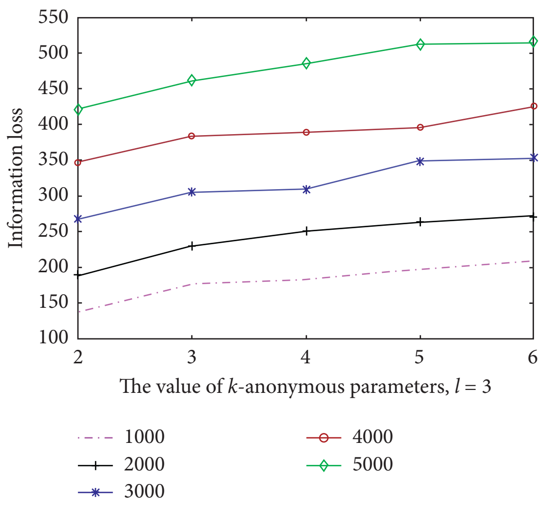


Figure 4: Information loss of STPP algorithm.

to an increase of spatiotemporal sensitive data in clusters, and IL in each cluster will increase correspondingly.

For evaluating the performance of privacy protection, we use the probability of attackers' attack success to quantify and compare, that is, attackers guess the probability of participants' specific spatiotemporal data based on the published sensing data. Suppose that n sensing data are published, and spatiotemporal sensitive data $d_i, 1 \le i \le n$, are aggregated into r location clusters and c time clusters. In this paper, equation 7 is used to quantify privacy protection, where $\sum_{i=1}^{r}(1/|u_i|)/r$ and $\sum_{j=1}^{c}(1/|c_j|)/c$ represent the average probability that attackers can infer real location attribute and time attribute of each sensing data, respectively:

$$p = \frac{1}{n} \times \frac{\sum_{i=1}^{r}\left(1/\left|u_i\right|\right)}{r} \times \frac{\sum_{j=1}^{c}\left(1/\left|c_j\right|\right)}{c}. \tag{7}$$

Figure 5 shows the comparison of experimental result by comparing the STPP algorithm with k-anonymous and VCLA algorithms on privacy protection. The x-coordinate is the number of participants on the New York City Taxi Trip dataset, and the y-coordinate is the probability of attackers to infer specific spatiotemporal data of participants based on the published sensing data. From the experimental result, it can be seen that the privacy protection gradually increases with the increase of participants. It is because that if the number of participants increases, the sensing data published by the application server will increase correspondingly, which reduces the probability of attackers' attack success, since the probability of successful attack without

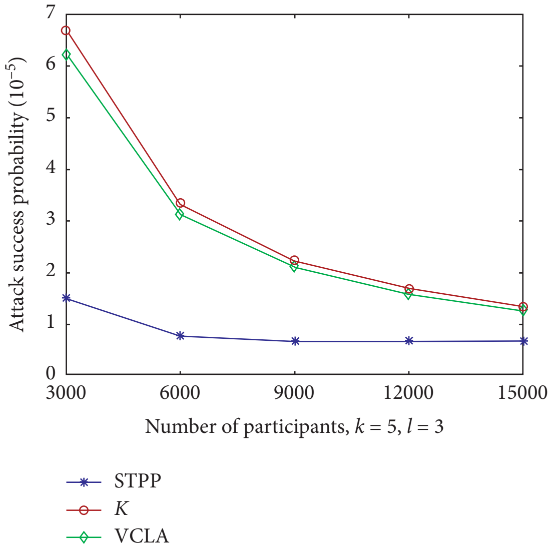


Figure 5: Experimental result on privacy protection.

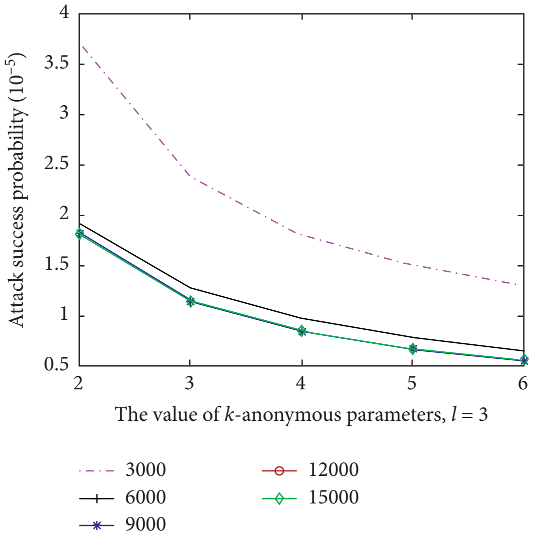


Figure 6: Privacy protection of the STPP algorithm.

background knowledge is very low (y-coordinate unit is 10^{-5}). STPP algorithm is slightly better than the comparison algorithms on privacy protection.

Figure 6 shows the relationship between the parameter k of k-anonymous and the privacy protection of the STPP algorithm. We conduct the experiments on New York City Taxi Trip dataset. From the experimental result, we can see that with the increase of k, the privacy protection increases gradually, which is because that increasing k leads to an increase of spatiotemporal sensitive data in each cluster, and the average probability is reduced that the real spatiotemporal data are inferred by attackers.

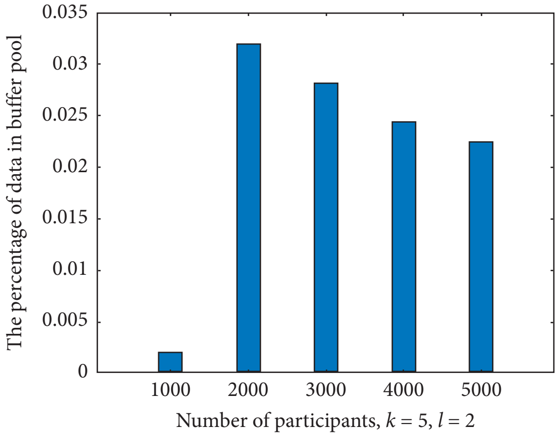


Figure 7: The proportion of sensing data in buffer pool.

When participants upload sensing data, TTPs will temporarily store sensing data that do not meet anonymity condition into buffer pool. Then, TTPs wait for the arrival of the next incremental data, which will generate the problem of delayed publish of sensing data. Figure 7 shows the ratio of buffer pool data to the number of sensing data for this publish. The x-coordinate is the number of participants on the Friendship Network dataset, and the y-coordinate is the ratio of sensing data in buffer pool. It can be seen that the proportion of data in buffer pool is very low, which proves that the sensing data in buffer pool have no great impact on delayed publish.

Through experiments on real-world datasets, we can see that the proposed STPP algorithm is superior to k-anonymous and VCLA algorithms in terms of running time, information loss, and privacy protection. STPP algorithm could solve the privacy protection problem of dynamic publishing for spatiotemporal crowdsourcing.

5. Conclusions

In the existing work, few researchers focus on privacy protection for dynamic publishing mechanism. There are few privacy protection methods for spatiotemporal sensitive data in dynamic publishing mechanism. In this paper, a dynamic publishing mechanism for spatiotemporal sensitive data privacy protection is proposed. Then, we design the dynamic k-anonymity algorithm and add the spatiotemporal data that met the adaptive threshold condition to the corresponding equivalence classes, making full use of the previous anonymous result to solve the problem of poor timeliness of static publishing. Thirdly, aiming at the shortcomings of k-anonymity, which is vulnerable to background knowledge attacks and homogeneous attacks, we anonymize participants' time attribute based on l-diversity, so as to improve privacy protection and reduce information loss. Finally, the performance of the proposed STPP algorithm is evaluated on two real-world datasets. Compared with the existing algorithms, experimental results show that STPP algorithm has lower time complexity, less information loss, and stronger privacy protection.

In the future, we will detect and process malicious participants (i.e., outliers) so as to better reduce information loss and protect participants' privacy data.

Data Availability

The experiment data used to support the findings of this study have been deposited in the GitHub repository (https://github.com/ltn21999/K_L-dynamic-privacy-protection).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant nos. 61822602, 61772207, 61802331, 61572418, 61602399, 61702439, and 61773331, the China Postdoctoral Science Foundation under Grant nos. 2019T120732 and 2017M622691, the National Science Foundation (NSF) under Grant nos. 1704287, 1252292, and 1741277, and the Graduate Innovation Foundation of Yantai University (GIFYTU) under Grant nos. YDYB2024 and YDZD1908.

References

[1] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.

[2] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, "Community-diversified influence maximization in social networks," *Information Systems*, vol. 92, pp. 1–12, 2020.

[3] Y. Wang, Z. Cai, Z.-H. Zhan, Y.-J. Gong, and X. Tong, "An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 3, pp. 414–429, 2019.

[4] N. A. H. Haldar, J. Li, M. Reynolds, T. Sellis, and J. X. Yu, "Location prediction in large-scale social networks: an in-depth benchmarking study," *VLDB Journal*, vol. 28, no. 5, pp. 623–648, 2019.

[5] J. Wang, Z. Cai, and J. Yu, "Achieving personalized k-Anonymity-Based content privacy for autonomous vehicles in CPS," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4242–4251, 2020.

[6] Z. Xiong, W. Li, Q. Han et al., "Privacy-preserving auto-driving: a GAN-based approach to protect vehicular camera data," in *Proceedings of 2019 IEEE International Conference on Data Mining (ICDM)*, Beijing, China, November 2019.

[7] M. Bi, Y. Wang, Y. Li, and X. Tong, *A Privacy-Preserving Mechanism Based on Local Differential Privacy in Edge Computing*, China Communications, Hong Kong, China, 2020.

[8] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[9] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[10] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," *Proceedings of the VLDB Endowment*, vol. 7, no. 10, pp. 919–930, 2014.

[11] K. Vu, R. Zheng, and J. Gao, "Efficient algorithms for k-anonymous location privacy in participatory sensing," in *Proceeding of the IEEE INFOCOM*, pp. 2399–2407, Orlando, FL, USA, March 2012.

[12] S. B. Avaghade and S. S. Patil, "Privacy preserving for spatio-temporal data publishing ensuring location diversity using K-anonymity technique," in *Proceedings of the 2015 International Conference on Computer, Communication and Control (IC4)*, September 2015.

[13] Z. Cai, Z. Duan, and W. Li, "Exploiting multi-dimensional task diversity in distributed auctions for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, no. 99, p. 1, 2020.

[14] Y. Wang, Y. Gao, Y. Li, and X. Tong, "A worker-selection incentive mechanism for optimizing platform-centric mobile crowdsourcing systems," *Computer Networks*, vol. 171, pp. 1–14, 2020.

[15] T. Liu, Y. Wang, Y. Li, X. Tong, L. Qi, and N. Jiang, "Privacy protection based on stream cipher for spatio-temporal data in IoT," *IEEE Internet of Things Journal*, 2020.

[16] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *Proceedings of the 39th IEEE International Conference on Distributed Computing Systems (ICDCS*, Dallas, TX, USA, July 2019.

[17] X. Wang, Z. Liu, X. Tian et al., "Incentivizing crowdsensing with location-privacy preserving," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6940–6952, 2017.

[18] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering (TNSE)*, vol. 7, no. 2, pp. 766–775, 2020.

[19] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "Energy efficient dynamic offloading in mobile edge computing for internet of things," *IEEE Transactions on Cloud Computing*, 2019.

[20] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, "Joint optimization of offloading utility and privacy for edge computing enabled IoT," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2622–2629, 2020.

[21] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2020.

[22] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[23] Y. Wang, Z. Cai, X. Tong, Y. Gao, and G. Yin, "Truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems," *Computer Networks*, vol. 135, pp. 32–43, 2018.

[24] P. Samarati, "Protecting respondents identities in microdata release," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[25] L. Fang and L. Tong, "A clustering K-anonymity privacy-preserving method for wearable iot devices," *Security and Communication Networks*, vol. 2018, Article ID 4945152, 8 pages, 2018.

[26] S. C. Lin, A. Y. Ye, and L. Xu, "K-anonymity location privacy protection method with coordinate transformation," *Journal of Chinese Computer Systems*, vol. 37, pp. 119–123, 2016.

[27] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart internet of things systems: a consideration from a privacy perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[28] H. Wang, H. Huang, Y. Qin, Y. Wang, and M. Wu, "Efficient location privacy-preserving K-anonymity method based on the credible chain," *ISPRS International Journal of Geo-Information*, vol. 6, no. 6, p. 163, 2017.

[29] A. Machanavajjhala, J. Gehrke, D. Kifer et al., "L-diversity: privacy beyond k-anonymity," in *Proceedings of the 22nd International Conference on Data Engineering*, April 2006.

[30] T. Dargahi, M. Ambrosin, M. Conti, and N. Asokan, "ABAKA: a novel attribute-based k-anonymous collaborative solution for LBSs," *Computer Communications*, vol. 85, pp. 1–13, 2016.

[31] A. Abdrashitov and A. Spivak, "Sensor data anonymization based on genetic algorithm clustering with L-Diversity," in *Proceedings of the 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT)*, April 2016.

[32] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong, and G. Wu, "An incentive mechanism with privacy protection in mobile crowdsourcing systems," *Computer Networks*, vol. 102, pp. 157–171, 2016.

[33] Z. Hu, J. Yang, and J. Zhang, "Personalized trajectory privacy protection method based on user-requirement," *International Journal of Cooperative Information Systems*, vol. 27, no. 3, 2018.

[34] F. Tian, S. Zhang, L. Lu et al., "A novel personalized differential privacy mechanism for trajectory data publication," in *2017 Proceedings of the International Conference on Networking & Network Applications (NaNA)*, October 2017.

[35] T. Peng, Q. Liu, D. Meng et al., "Collaborative trajectory privacy preserving scheme in location-based services," *Information Sciences*, vol. 387, pp. 165–179, 2017.

[36] D. Sun, Y. Luo, G. Fan et al., "Privacy protection algorithm based on trajectory shape diversity," *Journal of Computer Applications*, vol. 36, no. 6, pp. 1544–1551, 2016.

[37] X. Zheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Follow but No track: privacy preserved profile publishing in cyber-physical social systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1868–1878, 2017.

[38] X. Xu, B. Shen, X. Yin et al., "Edge server quantification and placement for offloading social media services in industrial cognitive IoV," *IEEE Transactions on Industrial Informatics*, no. 99, p. 1, 2020.

[39] K. Wang and B. C. M. Fung, "Anonymizing sequential releases," in *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Philadelphia, PA, USAACM, Philadelphia, PA, USA, August 2006.

[40] X. Xiao, "M-invariance: towards privacy preserving re-publication of dynamic datasets," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Beijing, China, June 2007.

[41] S. Cheng, C. Xu, and H. Dan, "Research on incremental privacy preserving data mining," *Application Research of Computers*, vol. 3, no. 8, 2018.

[42] F. Yan, X. Zhang, C. Li et al., "Differentially private histogram publishing through Fractal dimension for dynamic datasets," in *Proceedings of IEEE 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1542–1546, Wuhan, China, June 2018.

[43] Y. Wang, Z. Cai, Z. Zhan, B. Zhao, X. Tong, and L. Qi, "Walrasian equilibrium-based multiobjective optimization for task allocation in mobile crowdsourcing," *IEEE Transactions on Computational Social Systems*, 2020.

[44] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 665–673, 2018.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8860418, 12 pages
https://doi.org/10.1155/2020/8860418

Research Article

Packet-Based Intrusion Detection Using Bayesian Topic Models in Mobile Edge Computing

Xuefei Cao,[1] **Yulong Fu**,[1] **and Bo Chen**[2]

[1]*School of Cyber Engineering, Xidian University, No. 2 South Taibai Rd., Xi'an 710071, China*
[2]*National Lab of Radar Signal Process, Xidian University, No. 2 South Taibai Rd., Xi'an 710071, China*

Correspondence should be addressed to Xuefei Cao; xfcao@xidian.edu.cn

Received 6 May 2020; Revised 6 June 2020; Accepted 7 July 2020; Published 25 August 2020

Academic Editor: Xiaolong Xu

In this paper, a network intrusion detection system is proposed using Bayesian topic model latent Dirichlet allocation (LDA) for mobile edge computing (MEC). The method employs tcpdump packets and extracts multiple features from the packet headers. The tcpdump packets are transferred into documents based on the features. A topic model is trained using only attack-free traffic in order to learn the behavior patterns of normal traffic. Then, the test traffic is analyzed against the learned behavior patterns to measure the extent to which the test traffic resembles the normal traffic. A threshold is defined in the training phase as the minimum likelihood of a host. In the test phase, when a host's test traffic has a likelihood lower than the host's threshold, the traffic is labeled as an intrusion. The intrusion detection system is validated using DARPA 1999 dataset. Experiment shows that our method is suitable to protect the security of MEC.

1. Introduction

Mobile edge computing (MEC) has become the main feature of 5G communications [1]. During the development of MEC, researchers have always been keeping a focus on security issues. The security issues in MEC include application layer security, network layer security, data security, and node security. Intrusion detection systems (IDSs) protect the network layer security for MEC and have been an important component in it [2]. There are two methods to detect intrusions in general, i.e., signature-based method and anomaly-based method. The signature-based method predefines the patterns of intrusions and matches the network traffic against the patterns to raise detection alarms. While this method has low false alarm rate, it gives less than satisfactory results in detecting new types of attacks beyond the predefined patterns. The anomaly-based method establishes the normal behavior patterns for network traffic and if the pattern is accurate and extensive enough, any behavior different from the former would be regarded as an intrusion. The anomaly-based method has the ability to detect the "zero day exposure" attacks, and requires no prior knowledge of attacks. This makes the anomaly-based method superior to the signature-based method. Given the large amount of data and the diversity in services in MEC, the anomaly-based method proposes an attractive choice for MEC [3–5]. The main challenge of anomaly-based detection is how to establish an accurate and efficient behavior pattern using the normal network traffic.

There are two methods to realize the anomaly-based IDS, i.e., host-based method and network-based method [6]. In the host-based method, the network traffic to and from a single host is put together, and the host is analyzed according to the traffic. An independent behavior pattern would be established for the host's traffic. In the network-based method, however, the network traffic of all the hosts in the network is analyzed as a whole. Different hosts usually devote to different tasks, such as e-mail delivery and web page proxy, and they have different behavior patterns. Therefore, a host-based method will yield a more accurate behavior pattern compared with the network-based method [7].

LDA (latent Dirichlet allocation) is proposed by Blei et al. [8]. LDA views a document as a mixture of a series of probabilistic topics, and each topic is a collection of related words. A document is generated by first selecting several topics and then selecting words from each topic [9]. Given a collection of documents, one can deduce the topics covered by the corpus using LDA. For example, given 5000 documents which cover different topics, LDA is able to identify what these topics are from the documents. After running LDA on the 5000 documents, one can obtain a description of these topics by providing the words used with high frequency in each topic. For one topic, the LDA model could output the words used in it as film, show, music, movie, actor, play, musical, best, and so on; for another topic, LDA could output the frequently used words as million, tax, program, budget, billion, federal, year, spending, and so on. LDA does not generate the topic name, only the words used, but we know what the topic is about by looking at the words. In the above examples, the name of the first topic could be "arts," and the name of the second topic could be "budget." Because of LDA's ability to extract topics included in a large document corpus, it could be used for text categorization, document modeling, and collaborative filtering. Furthermore, we could apply it to analyze the network traffic which is also huge in volume. The resulted topics of network traffic could be viewed as a behavior pattern of network activities. If we only provide the normal traffic to LDA, then it could generate a behavior pattern of normal traffic. Given a new session of network traffic, if it deviates from the normal behavior pattern, it is likely to be an intrusion.

Based on the above idea, in this paper, we study the problem of intrusion detection in MEC using the LDA model. The challenge is how to analyze network traffic with LDA, i.e., how to turn the traffic into "documents," and how to define the "words" in network traffic so that the resulted "topics" could represent the behavior pattern of normal network activities. We propose a method to draw analogue between network traffic and documents. A comprehensive set of network features is abstracted from tcpdump packet headers, and the network traffic is turned into documents based on these features. We also propose a method to analyze network traffic using the LDA model for intrusion detection. Our method is testified on the widely used network traffic dataset DARPA 1999, and according to the experiment results, our method could detect the intrusions effectively in MEC.

We list the main contributions of our research:

(i) We explore the usage of LDA in the anomaly-based intrusion detection systems in MEC. As far as we know, this is the first intact work of applying LDA to the intrusion detections.

(ii) We propose the method of transforming network traffic into "documents" which are required by LDA. We propose to use packets in network traffic analysis. We select 16 feature fields and use the unique values in each feature fields as "words." We also propose a method to build vocabulary list. Based on this setting of words and vocabulary list, we are able to turn network traffic into documents and process the network traffic with LDA.

(iii) We present a way to detect intrusions using LDA. A host-based method is employed. LDA is used to analyze normal traffic of a host and extract the behavior pattern of the host. Then, the host's likelihood to the behavior pattern is computed. The lowest likelihood is used as a threshold. For a new traffic, if the likelihood is lower than the threshold, it is classified as an intrusion.

(iv) We validate our method in the widely tested dataset and compare the result to the result of existing scheme. According to the comparison results, our method could detect the network intrusion with a higher detection rate.

The remainder of this article is organized as follows: Section 2 discusses the state-of-the-art research results in the field, Section 3 introduces the LDA model, Section 4 proposes our method, and Section 5 describes the experiment using our method while Section 6 concludes the paper.

2. Related Works

Intrusion detection systems could be divided into three broad categories according to the types of network traffic in use.

One form of network traffic in use is system calls. Forrest et al. pioneered in proposing using the traces of system calls to detect the possible intrusions [10]. They trained an n-gram model ($n = 3$ to 6) over the normal system calls for a given host and looked in the test data for the trace differences. Liao and Vemuri introduced the text categorization techniques to Forrest's method [11]. They employed the k-nearest neighbor classifiers to count the system call frequency to describe the normal program behavior. Then, each process is converted into a vector and the similarity between processes is calculated using the text categorization technique. To determine whether a process is normal or not, they chose k neighbors with the nearest similarity and compared the process with the k neighbors. Ding et al. used sematic analysis of system calls to extract static behavior from executable programs [12]. The static behavior is defined as the sequences of system calls and is used to detect the malicious codes. A method of deriving system call sequences is presented, and an n-gram model is used to extract the features from the system calls. Creech et al. used system calls by applying a semantic analysis to kernel level system calls and derived a new feature to classify the network activities as normal or intrusion [13]. Maggi et al. proposed a host-based intrusion detection system using system call arguments and sequences [14]. They defined a set of anomaly detection models for the individual parameters of the call and added a stage of clustering in order to better fit models to arguments. The model is complemented with Markov models to capture the correlation between system calls.

Another form of audit data is TCP/IP connection descriptions which include the summarization of high-level interactions between hosts such as session duration, type of service, number of failed login attempts, status of guest log in, and so on. Many systems first reconstruct raw network data into connections and extract connection features before carrying out detection techniques. MADAMID [15], Bro [16], and EMERALD [17] are systems of this kind. These systems analyze the TCP/IP connections to abstract the behavior patterns of normal traffic and then detect the intrusions based on the behavior patterns. Stolfo et al. participated in the 1998 DARPA Lincoln Lab intrusion detection evaluation program [18]. Their project proposed an intrusion detection system and applied it to the DARPA 1998 dataset. They abstracted TCP/IP connections from DARPA 1998 tcpdump packets and then applied the data mining technique to the TCP/IP connections to obtain different features. They built specialized models using these features. The outputs of the models were the rules with which a classifier was trained to make final classification to a new connection. To remove the burden of transforming tcpdump packets into TCP/IP connections, the KDD99 dataset [19] was proposed. It is a revised version of the DARPA 1998 dataset [20] in which raw network traffic was summarized into TCP/IP connections where each connection is expressed by a set of network features. Various machine learning techniques have been applied to the dataset [19] and shown their effectiveness, for example, Naive Bayesian [21–24], nearest neighbor [25–29], neural networks [30–33], and fuzzy logic [34, 35].

The third form of network traffic in use is tcpdump packets. In some attacks, certain packet feature fields or payloads always employ less common values in order to launch successful attacks. Therefore, by analyzing the values of certain packet feature field, one can construct an effective intrusion detection system. One example is the firewall system: it secures the system by blocking the packets to certain ports or hosts. Recent research studies propose an improved method by building sophisticated models and combining more packet features to gain better detection results. The research in this line was first started by Mahoney, who proposed PHAD (packet header anomaly detector) by modeling more than 30 packet features and computed the abnormal score over the selected features. Attacks were detected based on the abnormal scores of a packet [36]. NETAD is an improvement of PHAD which is also proposed by Mahoney [37]. NETAD deleted the most notable packets including a connection's beginning and ending packets, and then it abstracted features from the first 48 bytes of a packet and modeled the protocol behavior accordingly. Scheme in [38] also used tcpdump packets and applied genetic algorithm to tcpdump packets. Reference [39] used similar packet feature fields as PHAD does [36], and it constructed a network behavior model for every protocol adopted in the traffic. Yassin et al. proposed a host-based PHAD [40]. They scored the packet features and performed the division of normal and abnormal using linear regression and Cohen's d measurement. Hareesh et al. [41] detected network attacks and worms by analyzing the packet header and payload. The research generated histogram for different IP header values, TCP flags, and payload. The histogram was used to represent the number of flows associated to a feature in a certain time. Then, data mining was employed to establish the normal behavior pattern given these histograms. Manandhar and Aung [42] analyzed the tcpdump packets but with a session-based method involving more packets to detect the high-level attacks.

There have been continuous efforts to apply the LDA model to the analysis of network traffic and cyber data. Cramer and Carin[43] studied the patterns of the network traffic in a corporation environment using LDA. They discovered the pattern differences in network usage between daytime and nighttime. Ferragut et al. [44] proposed several constructions of anomaly detectors in LDA's framework and noticed several abnormalities in a laboratory network. Huang et al. proposed an idea to analyze network traffic using LDA [45]. They suggested that network event can be regarded as "vocabulary," and a collection of events a user has done in a given time can be regarded as "document." They showed the possibility to detect network intrusions using the LDA model, but no detailed scheme was given. Steinhauer et al. proposed an anomaly detection system using LDA for telecommunication system [46]. They discussed the possibility of introducing LDA to the analysis of telecommunication network traffic. It turned out that the topics learned by LDA conformed to the telecommunication activities. But the proposal depended heavily on telecommunication experts to explain the result of the LDA model. Lee et al. proposed LARGen, an LDA-based automatic rule generation tool for signature-based intrusion detection systems [47]. They used LDA to analyze the network traffic and extracted the key content and signatures of malicious traffic. Then, IDS rules were built upon the signatures. They tested their method on some real network traffic.

3. Topic Model

Latent Dirichlet allocation is a statistical Bayesian topic model which could be used to infer the latent semantics of a set of documents. The LDA model is constructed under a basic assumption that the observed documents are yielded with a set of topics which are the probabilistic distributions over words. Each document is generated by first selecting the topics for the document and then selecting words from every topic [48].

The notations used in LDA are defined as follows:

(i) The vocabulary is a vector V, which is a collection of all the words used in the corpus. The length of the vocabulary is denoted as $|V|$. LDA treats a document using the concept of *bag-of-words*, i.e., a document is expressed using the predefined vocabulary and the times each word in the vocabulary appearing in the document; however, the order of words in the document is not considered.

(ii) There are D documents in all in the corpus. The d-th document in the corpus is expressed by $\mathbf{w}_d$ with $1 \leq d \leq D$. $\mathbf{w}_d$ is a vector of the size $1 \times |V|$, and each element in $\mathbf{w}_d$ is the times a word in the vocabulary appears in the document.

(iii) The corpus is $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_D\}$.

(iv) LDA assumes that the corpus $\mathbf{w}$ is generated by K latent topics. θ_d follows a Dirichlet distribution with prior parameter α, and it denotes the topic distribution for the d-th document. θ_d is a row vector with K columns (a $1 \times K$ vector) with all elements in θ_d adding up to 1 and with the i-th element in θ_d representing the portion of the i-th topic in $\mathbf{w}_d$.

(v) $\theta = \{\theta_1, \theta_2, \ldots, \theta_D\}$ is a $D \times K$ matrix and denotes the topic distribution of $\mathbf{w}$.

(vi) A topic is presented by ϕ_i, which is a $|V| \times 1$ vector denoting the word distribution over vocabulary of the i-th topic. ϕ_i also follows a Dirichlet distribution with prior parameter β. All elements of ϕ_i add up to 1.

(vii) K topics $\{\phi_1, \phi_2, \ldots, \phi_K\}$ are used for $\mathbf{w}$. We define Φ as the topic distribution of the corpus $\mathbf{w}$. $\Phi = \{\phi_1 \phi_2 \cdots \phi_K\}$ is a $|V| \times K$ matrix.

(viii) A word of a document is expressed by $w_{d,n}$ where $1 \le d \le D$, $1 \le n \le N_d$. N_d means that there are N_d words in the document $\mathbf{w}_d$. $z_{d,n}$ is the topic label for the n-th word in the d-th document $w_{d,n}$ and $z_{d,n} \in \{1, \ldots, K\}$. $z_{d,n}$ follows a multinomial distribution with θ_d as its prior parameter.

Table 1 provides a summarization of the notations.

Figure 1 shows the generative process of LDA.

In the figure, α is the hyperparameter (prior parameter) for θ and β is the hyperparameter for Φ. For each document in a corpus, a topic distribution θ_d is drawn based the hyperparameter α, and this process is repeated for D times. For each word in a document, a topic label is drawn based on the topic distribution, and the draw of a topic label $z_{d,n}$ is repeated N_d times for every word in the d-th document.

The distributions of variables in Figure 1 are as follows:

$$\begin{aligned} \theta_d &\sim \mathrm{Dir}(\alpha), \\ z_{d,n} &\sim \mathrm{Multinomial}(\theta_d), \\ \phi_i &\sim \mathrm{Dir}(\beta), \\ w_{d,n} &\sim \mathrm{Multinomial}\left(\phi_{z_{d,n}}\right). \end{aligned} \tag{1}$$

In the distributions above, " ~ " means "follows," $\mathrm{Dir}(x)$ means a Dirichlet distribution with hyperparameter x, and $\mathrm{Multinomial}(y)$ means a multinomial distribution with hyperparameter y.

Table 2 summarizes the parameters used in the LDA model.

$\mathbf{w}$ is an observable variable and is shown in gray in Figure 1; θ, Φ, and $\mathbf{z}$ are latent variables shown in white. LDA's goal is to infer θ and Φ. Variational Bayes and Gibbs sampling are two effective methods to do the inference [8, 9]. Gibbs sampling is a typical method to do the inference of hierarchical Dirichlet structures, and it is able to calculate the exact conditional posterior; in this paper, we use Gibbs sampling instead of variational Bayes. The performance of Gibbs sampling is slightly better than that of the variational Bayes but the speed is a little slower.

Gibbs sampling works on the idea that the topic label of a particular word is determined by the topic labels of all the other words in the corpus. Given the observed corpus $\mathbf{w}$, Gibbs sampling first calculates $\mathbf{z}$ according to the conditional posterior of $\mathbf{z}$ and then calculates θ and Φ according to the distribution of $\mathbf{z}$. We describe this process as follows:

(i) Draw θ_d for $d = \{1, \ldots, D\}$ using Dirichlet distribution with hyperparameter α. Draw ϕ_i for $i = \{1, \ldots, K\}$ using Dirichlet distribution with hyperparameter β. These are the initial values of θ and Φ.

(ii) Compute the conditional posterior of $z_{d,n}$ given the word $w_{d,n}$. The conditional posterior $p(z_{d,n} = j \mid w_{d,n}, \alpha, \beta)$ equals to $\phi_{w_{d,n},j} \times \theta_{d,j} \times c$ where c is a coefficient.

(iii) Draw $z_{d,n}$ according to the conditional posterior $p(z_{d,n} \mid w_{d,n}, \alpha, \beta)$.

(iv) Update the distribution of $z_{d,n}$ for every word in the d-th document accordingly.

(v) Calculate the conditional posterior of θ_d according to the Dirichlet distribution, but the hyperparameter should consider the distribution of $z_{d,n}$ in the d-th document.

(vi) After all documents have been processed, calculate the conditional posterior of ϕ_i according to the Dirichlet distribution, but the hyperparameter should consider the distribution of $z_{d,n} = i$ in $\mathbf{w}$.

The inferences of $z_{d,n}$, θ_d, and ϕ_i are as follows:

$$\begin{aligned} p\left(z_{d,n} = j \mid w_{d,n}, \alpha, \beta\right) &\propto \theta_{dj} \cdot \phi_{w_{d,n}j}, \\ p\left(\theta_d \mid \mathbf{w}_d, \alpha, \beta\right) &= \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \cdot \prod_{i=1}^{K} \theta_{di}^{n(d,i)+\alpha_i-1}, \\ p\left(\phi_i \mid \mathbf{w}, \alpha, \beta\right) &= \frac{\Gamma\left(\sum_{j=1}^{V} \beta_j\right)}{\prod_{j=1}^{V} \Gamma\left(\beta_j\right)} \cdot \prod_{j=1}^{V} \phi_{ij}^{n(\cdot,i,j)+\beta_j-1}, \end{aligned} \tag{2}$$

where $\Gamma(n) = (n-1)!$, $n(\cdot, i, j)$ is the number of word j assigned to topic i in the corpus, and $n(d, i)$ is the number of words in document d assigned to topic i.

4. Our Method

In this section, we propose our packet-based intrusion detection method using LDA for MEC. We first summarize how to generate documents from tcpdump packets and then describe our method of intrusion detection using LDA in detail.

4.1. Data Preprocessing. We consider the problem of how to transfer network traffic into documents that LDA model can handle. This procedure should be carried out before we can use the LDA model.

The network traffic we use is tcpdump packets because they help us to turn network traffic into documents easily. To turn the network traffic into documents, we should first construct the vocabulary list. We use a host-based intrusion

Table 1: Notations used.

Notation	Description
D	Number of documents
K	Number of topics
N_d	Number of words in the d-th document
V	Vocabulary list
$\mathbf{w}$	The corpus of documents
$\mathbf{w}_d$	The d-th document
$w_{d,n}$	The n-th word in the d-th document
θ	A $D \times K$ matrix, the topic distribution of the whole corpus
θ_i	A $1 \times K$ vector, the topic distribution of the i-th document
Φ	A $\lvert V\rvert \times K$ matrix; column i denotes the topic-word distribution of the i-th topic
ϕ_i	A $\lvert V\rvert \times 1$ vector, the word distribution of the i-th topic
$z_{d,n}$	The topic label of $w_{d,n}$, $z_{d,n} \in \{1, \ldots, K\}$

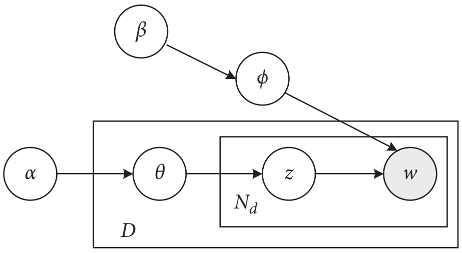


Figure 1: Illustration of the LDA model.

detection method, so we build an independent vocabulary list for each host. A tcpdump packet is composed of packet header and packet payload, and it could be grasped and analyzed by network sniff tools such as Wireshark [49]. In the packet header of a tcpdump packet, many feature fields are well defined such as MAC address, IP address, TCP service port, and so on. The format of packet header and feature fields is defined by corresponding IETF specifications. For example, RFC 791 [50] defines the format of IP header and RFC 793 [51] defines the TCP header. For most packets, the first packet header is the Ethernet header, followed by an IP header. According to the IETF specification RFC 791 [50], the IP header has a length of 20 bytes, and the 13th to the 16th byte is the source address sending out the packet. Since the tcpdump packets have such a well-defined format, we can make use of it. The values in the feature fields of a packet header can be treated as words, and the vocabulary list is the collection of all possible feature values. To define the vocabulary list suitable for intrusion detections, we select 16 feature fields from the packet header. According to available research result, these feature fields are used widely in IDS [36] and are shown in Table 3. In Table 3, the content in the bracket is the abbreviation name of the feature field. 16 feature fields are selected for a packet, and thus 16 words are generated from one packet. Each word is a combination of the feature field abbreviation name and the feature value.

Here, we give an example. A tcpdump packet is shown in Figure 2 using Wireshark. The packet is 79 B in length. The first 6 bytes indicate the Ethernet destination address, and the 7–12 bytes indicate the Ethernet source address. Thus, we can have two words EDST_00000c and ESRC_006097 according to the definitions in Table 3. The Ethernet header is followed by an IP header, the 16th byte is the type of service (0x10), and 17th-18th bytes are the IP packet length (0x0041 in hex and 65 in decimal), and thus we have two words TOS_10 and IL_65. Using the same method, we can abstract 16 words from the packet in all, and they are EDST_00000c, ESRC_006097, TOS_10, IL_65, FF_4000, TTL_40, SRC_192.168.1.30, DST_192.168.0.20, SP_21, TF_18, TU_0, TC_ffff, TO_Null, UC_Null, IC_Null, and PS_79.

However, to generate vocabulary list, we need further considerations. Since in our scheme, we use only normal traffic in the generation of vocabulary list, it may not cover all the feature values. Attacks usually employ feature values which are not covered in the normal traffic. To deal with these values, 16 extra words are added to vocabulary list to cover the features which do not appear in normal traffic, but could appear in attacks (or the test phase). For each feature field, we add an extra value, and this extra value is expressed by the combination of the field's abbreviation name and "others." For example, for the IP source field, the extra value is SRC_others. We use SRC_others to cover all the IP source addresses which do not appear in the normal traffic, but appear in the test phase. Therefore, the resulting vocabulary is all the unique feature values appearing in the traffic adding the 16 "_others" values. Let $F_i\, i \in \{1, 2, \ldots, 16\}$ denote the number of different features appearing in the i-th feature field; then, $|V| = F_1 + F_2 + \cdots + F_{16} + 16$ is the length of the vocabulary list.

Given the vocabulary list, we could transfer the traffic into documents. We view the tcpdump traffic in a given time length, for example, five minutes, as a document. We count which words are used in the document and count the times a word is used. A document is expressed as the times of words in vocabulary appearing in the document.

We also list the meaning of notations when we turn network traffic into documents in Table 4.

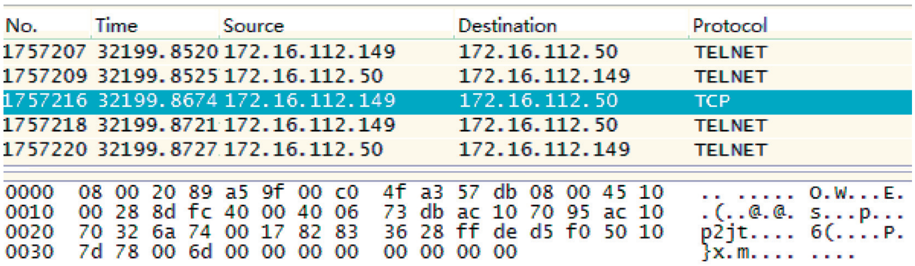
4.2. Intrusion Detection Using LDA. Given the documents transformed, we use an anomaly-based method to detect intrusions using LDA. Since LDA is able to extract the latent semantics of a corpus, we use it to abstract the latent behavior structure of network traffic. We train the LDA model

Table 2: Definitions of parameters.

Parameter	Definition
θ	Topic distribution of documents; follows a Dirichlet distribution with hyperparameter α
α	Hyperparameter of θ
Φ	Topic distribution over vocabulary; follows a Dirichlet distribution with hyperparameter β
β	Hyperparameter of Φ
z	Topic label of a word; follows a multinomial distribution with hyperparameter θ

Table 3: Feature list.

Packet layer	Features extracted
Ethernet layer	Higher 3 bytes of MAC source (ESRC), higher 3 bytes of MAC destination (EDST)
IP layer	IP length (IL), type of service (TOS), fragment flags (FF), time to live (TTL), IP source (SRC), IP destination (DST)
Transport/control layer	TCP flag (TF), TCP checksum (TC), TCP URG pointer (TU), TCP option (TO), UDP checksum (UC), ICMP checksum (IC), Service port (SP)
Others	Packet size (PS)



```
No.      Time       Source          Destination     Protocol
1757207  32199.8520 172.16.112.149  172.16.112.50   TELNET
1757209  32199.8525 172.16.112.50   172.16.112.149  TELNET
1757216  32199.8674 172.16.112.149  172.16.112.50   TCP
1757218  32199.8721 172.16.112.149  172.16.112.50   TELNET
1757220  32199.8727 172.16.112.50   172.16.112.149  TELNET

0000  08 00 20 89 a5 9f 00 c0  4f a3 57 db 08 00 45 10   .. ..... O.W...E.
0010  00 28 8d fc 40 00 40 06  73 db ac 10 70 95 ac 10   .(..@.@. s...p...
0020  70 32 6a 74 00 17 82 83  36 28 ff de d5 f0 50 10   p2jt.... 6(....P.
0030  7d 78 00 6d 00 00 00 00  00 00 00 00               }x.m.... ....
```

Figure 2: Example of a tcpdump packet.

Table 4: Meaning of notations in network traffic.

Notation	Description
D	The number of five-minute tcpdump sessions
K	The number of normal behavior patterns
$\mathbf{w}_d$	The d-th five-minute tcpdump session
V	Unique features values in the traffic
$w_{d,n}$	The n-th feature in the d-th session (abstracted from the $\lceil n/16 \rceil$-th packet in the session)

with only normal traffic. After running LDA on the training traffic which contains only normal traffic, we can automatically obtain the inference of latent variables θ and Φ. Since our method uses only normal traffic, Φ in fact describes what correct behaviors look like. It summarizes the features that should be included in a normal traffic behavior. For example, a topic-word distribution ϕ_i for a host 172.16.112.100 is TO_null, FF_0000, SP_53, and DST_172.16.112.20, and thus the normal traffic pattern for the host 172.16.112.100 related to ϕ_i could be the connection with the host 172.16.112.20 using UDP protocol (service port 53). The topic distribution of a document θ_i is the behavior structure distribution of a given session of network traffic. It describes what kind of behaviors are included in this network traffic.

To raise the intrusion alarm, we employ the document likelihood. It can be explained as the degree of how much a document looks like the normal behavior structure. We use the lowest likelihood of a host in the training phase as the threshold. A test document is labeled as abnormal and an alarm is raised if the likelihood of the test documents is lower than the threshold. In our method, every host has its own threshold, and the threshold is the minimum of the likelihood of all the host's training documents.

The likelihood of a document is computed using the following equation:

$$\begin{aligned}\text{lik}_d &= \frac{1}{N_d}\prod_{n=1}^{Nd}\sum_{j=1}^{K} p\left(w_{d,n} \mid z_{d,n} = j, \phi_j\right) \cdot p\left(z_{d,n} = j \mid \theta_d\right) \\ &= \frac{1}{N_d}\prod_{n=1}^{Nd}\sum_{j=1}^{K} \phi_{w_{d,n},j}\theta_{dj}.\end{aligned} \tag{3}$$

To sum up, our method comprises four modules.

(i) Vocabulary list of a host is built based on the host's attack-free tcpdump packets data during a long enough time. Each packet is denoted by 16 features, and the anomalies of each feature field are collected. The vocabulary is the collection of all the anomalies in the 16 feature fields plus 1 extra word for each feature field.

(ii) Traffic is separated by host. A host's traffic is divided into segments, and each segment contains five minutes of tcpdump packets. A segment is transformed into a document by calculating which features are used in the segment and how many times every feature is used.

(iii) Train a LDA model for every host using the host's attack-free network traffic (training traffic) to compute Φ of the host. Use equation (3) to compute the likelihood of every training document of the host with Φ and θ_d. Set the minimal likelihood as the host's threshold.

(iv) In the test phase, according to Φ computed in the training phase and θ_d computed in the test phase, we compute the likelihood of every test document. The test document will be labeled as an attack if its likelihood is lower than the threshold.

5. Experimental Results

We implement our packet-based intrusion detection method using LDA described in Section 4 in this section. We describe the dataset used, data preprocess procedure, the training phase, the test phase, and the results.

5.1. Dataset Description. The network traffic used in this session is DARPA 1999 dataset of MIT Lincoln Laboratory which was prepared for 1999 DARPA intrusion detection evaluation program [52]. It is one of the most popular experimental datasets for network intrusion detection systems. Although it has many limitations such as the simplicity of the attacks, inaccuracy in the information, and so on, it is still used as the benchmark of many IDSs and provides a baseline to compare the performance of different IDSs.

The DARPA 1999 dataset provides a standard set of extensively gathered audit data, which comprises rich types of intrusions simulated in a military network environment. In the dataset, there are three weeks of training data and two weeks of test data. In the three weeks of training dataset, different types of data are provided, including the tcpdump data and audit data. There is no attack in Week 1 and Week 3 training traffic, and in Week 2 training traffic, there are attacks whose information is provided by the dataset. There are two weeks of test traffic, in which 201 attacks are provided and the attacks cover all four attack categories of 56 different types. The four attack categories include DOS, denial-of-service, e.g., Neptune; R2L, remote-to-local, unauthorized access from a remote machine to local machine, e.g., guessing password; U2R, user-to-root, unauthorized access to local superuser (root) privileges, e.g., eject; and probing, illegal scanning of service port, e.g., ipsweep. The ground truth of all the attacks in the test datasets is provided in an individual file.

In our experiment, we use the third week's 8-day tcpdump traffic (Mar 15–Mar 19 with three extra days) in the training phase. We use the inside.tcpdump data which are the data collected in the internal network. In the test phase, we use two weeks of test data (Mar 29–Apr 2 and Apr 5–Apr 9). Also, the inside tcpdump data are employed.

5.2. Data Preprocess Procedure. DARPA 1999 dataset is the traffic of an military network including multiple hosts. Our intrusion detection system is host-based; thus, in the preprocess of the data, we first divide the traffic according to the host addresses. 18 hosts produced the training and test traffic; therefore, we divide the network traffic according to the hosts. Figure 3 illustrates how the network traffic is divided in the training traffic, and every column in the figure represents a host. The test traffic is also divided using the same way.

For every host, we generate its own vocabulary list. The host's vocabulary list is generated using the host's training dataset. There are 18 vocabulary lists in all. The vocabulary is generated using the method described in Section 4.1. Take the vocabulary generation for the host 172.16.112.50 (Pascal) as an example. Table 5 illustrates the unique features employed by Pascal in the training phase. Table 6 illustrates the resulted vocabulary of Pascal. The size of vocabulary list for Pascal is 2788.

To convert the network traffic into documents, we divide the traffic of each host into five-minute sessions. The first packet's arriving time in the session is at most 300 seconds earlier than that of the last packet in the session. Such a time slot is chosen because we want to make the time slot large enough to cover a whole attack. Then, we calculate how many times every word is used in the session, and the session is turned into a document.

To illustrate how a document is generated, we assume a simplified session of Pascal with 200 packets. The session's feature distributions are shown in Table 7. Based on the vocabulary list of Pascal, the resulted document is a vector with the size of 1 × 2788. In the vector, the {1, 2, 8, 9, 15, 20, 1348, 1350, 1359, 1363, 1365, 1394, 1396, 1425, 1442, 1452, 1455, 1458, 1461, 1465, 1466, 1467, 2788} digits are set as {100, 100, 100, 100, 200, 199, 1, 200, 200, 100, 100, 100, 100, 200, 200, 200, 200, 200, 200, 199, 1, 199, 1}, respectively. Note that this session should come from a test session because it contains IL_others, IC_others, and PS_others values.

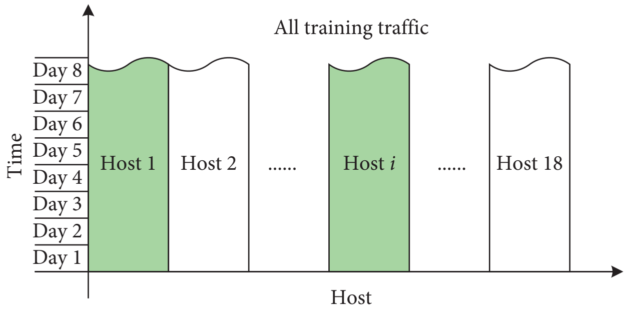


Figure 3: Example of data separation for training traffic.

Table 5: Feature anomalies of Pascal.

Field	Unique feature values	Anomaly #
EDST	0x0000c0, 0x00105a, 0x00107b, . . .	6
ESRC	0x0000c0, 0x00105a, 0x00107b, . . .	5
TOS	Null, 0x00, 0x08, 0x10	4
IL	Null, 38, 40–170, 172–217, . . ., 1489–1500	1329
FF	Null, 0x0000, 0x4000	3
TTL	Null, 60, 63, 64, 128, 254, 255	7
SRC	135.8.60.182, 135.13.216.191, 172.16.112.10 172.16.112.20, 172.16.112.50, . . .	29
DST	135.8.60.182, 135.13.216.191, 172.16.0.1 172.16.112.10, 172.16.112.20, 172.16.112.50, . . .	33
SP	Null, 20, 21, 22, 23, 25, 53, 79, 80, 113, 123, 514, 515, 6000, 6667, 8000	16
TF	Null, 0x02, 0x04, 0x10, . . .	9
TU	Null, 0	2
TC	Null, 0xffff	2
TO	Null, 0x020405b4	2
UC	Null, 0xffff	2
IC	Null, 0xffff	2
PS	60–184, 186–231, . . ., 1503–1514	1321

Table 6: Vocabulary list of Pascal.

Field	Words	Anomaly #
EDST	EDST_0000c0, EDST_00105a, EDST_00107b, . . ., EDST_others	7
ESRC	ESRC_0000c0, ESRC_00105a, ESRC_00107b, . . ., ESRC_others	6
TOS	TOS_Null, TOS_00, TOS_08, TOS_10, TOS_others	5
IL	IL_Null, IL_38, IL_40, IL_41, . . ., IL_1500, IL_others	1330
FF	FF_Null, FF_0000, FF_4000, FF_others	4
TTL	TTL_Null, TTL_60, TTL_63, TTL_64, . . ., TTL_255, TTL_others	8
SRC	SRC_135.8.60.182, SRC_135.13.216.191, SRC_172.16.112.10 SRC_172.16.112.20, . . ., SRC_others	30
DST	DST_135.8.60.182, DST_135.13.216.191, DST_172.16.0.1 DST_172.16.112.10, DST_172.16.112.20, . . ., DST_others	34
SP	SP_Null, SP_20, SP_21, SP_22, SP_23, SP_25, SP_53, . . ., SP_515, SP_6000, SP_6667, SP_8000, SP_others	17
TF	TF_Null, TF_02, TF_04, . . ., TF_others	10
TU	TU_Null, TU_0, TU_others	3
TC	TC_Null, TC_ffff, TC_others	3
TO	TO_Null, TO_020405b4, TO_others	3
UC	TC_Null, TC_ffff, TC_others	3
IC	IC_Null, IC_ffff, IC_others	3
PS	PS_60, PS_61, PS_62, . . ., PS_1513, PS_1514, PS_others	1322

Table 7: Example of a simplified session.

Field name	Feature value	Count number
EDST	0x0000c0	100
	0x00105a	100
ESRC	0x0000c0	100
	0x00105a	100
TOS	0x00	200
IL	38	199
	171	1
FF	0x0000	200
TTL	255	200
SRC	172.16.112.10	100
	172.16.112.50	100
DST	172.16.112.10	100
	172.16.112.50	100
SP	Null	200
TF	Null	200
TU	Null	200
TC	Null	200
TO	Null	200
UC	Null	200
IC	0xffff	199
	0xabcd	1
PS	60	199
	185	1

5.3. Training Phase. For each host, an independent LDA model is trained. This is because different hosts may be used for different purposes, for example, mail proxy and Internet server. As a result, the topic distributions could be totally different among hosts. The detection accuracy could be greatly improved if we train an individual LDA model for each host.

For all the hosts, the Dirichlet prior parameters α and β are set empirically to obtain good model quality. We have $\alpha = (10/K)$ and $\beta = 0.01$. According to [47], the number of topics used by LDA will have limited impact on the detection accuracy; thus, for most of hosts with a vocabulary size around 2000, we set $K = 150$ and set $K = 10$ for hosts 172.16.118.80 and 192.168.1.1 whose vocabulary sizes are around 100. Since too large a K will increase the running time, we do not choose K to be large. We use the training documents to train the LDA model. For different hosts, there are 1716 documents at most and 829 documents at least used in the training phase.

After the parameters of LDA have been set, we train the LDA model with the documents of a host and yield the topic-word distribution Φ and topic distribution θ of the host. Since only normal traffic is used in the training phase, Φ can be viewed as a normal behavior pattern of the host. The likelihood of each document is computed using equation (3), and the threshold of the host is set as the minimal likelihood.

5.4. Test Phase. We detect attacks in the test phase. In this phase, we run the LDA model still using the same parameter settings of α, β, and T. Φ is not inferred in this phase because the training phase has already computed Φ which is deemed as the normal behavior structure. The topic distribution θ of every test documents is inferred based on Φ computed in the training phase. The likelihood of each test document is computed using equation (3) according to Φ and the test document's topic distribution θ. It measures the extent to which the test document resembles the normal behavior structure. The document is identified as an attack if the likelihood is lower than the threshold. The host-based method is also used in the test phase.

5.5. Detection Results. All the 18 hosts generate 24037 documents using our method. Of all the documents, 1041 documents are labeled as intrusions. 490 are false positives and 730 are true positives. There are 94 attacks detected because several documents may correspond to one same attack instance.

We compare the performance of our scheme with the performance of PHAD [36] in terms of their ability in detecting intrusions. The comparison result is listed in Table 8. Column 1 of Table 8 is all the attack instances contained in the DARPA 1999 dataset, column 2 is the number of intrusions detected by PHAD, and column 3 is the number of intrusions detected by our method.

5.6. Result Analysis. From the comparison result of Table 8, we can see that our method is superior to PHAD because it detects more types and more instances of attacks. The reason is that our method employs LDA to learn the behavior rule of network traffic. By using LDA, every feature is treated as an independent variable, and all the features are used fully. The behavior rule for the normal traffic is generated automatically. In the LDA model, a topic is a representation of all the normal features with different probability. Φ is the description of normal traffic behavior. A five-minute session of traffic, or a document, should be generated by normal topics if it is to be labeled as normal. As a result, if a document's likelihood computed by the normal behavior rule, or Φ, is lower than the threshold, there may be an attack.

However, in PHAD, the behavior rule for normal traffic is generated by adding up all anomaly values of each feature field, and then the sum is used to separate attacks from normal traffic. The behavior rule generated in this way depends heavily on a single variable, and it is too strong. The information accuracy and extensiveness presented by features are lost by this method. As a result, PHAD cannot detect as many attacks as our method can detect. The limit of our method is that it detects fewer probe attacks such as portsweep, queso, and ipsweep. The reason is that our method is host-based but PHAD is network-based, and the latter has advantage to detect the probe attacks. To fix the problem, we can increase the weight of certain features including port number and TCP flag. We will look into this in our future work.

Table 8: Detection comparison.

Attack	#	Ours	PHAD
anypw	1	1	0
apache2	3	3	2
arppoison	4	1	0
back	4	3	0
casesen	3	2	1
crashiis	8	1	1
dict	1	1	0
dosnuke	4	4	4
Eject	2	2	0
fdformat	3	1	0
ffbconfig	2	1	0
ftpwrite	2	1	0
guessftp	2	1	1
guesspop	1	1	0
guesstelnet	4	2	2
httptunnel	3	1	0
imap	2	2	0
illegalsniffer	2	1	1
ipsweep	7	3	4
mailbomb	4	3	2
mscan	1	1	1
named	3	3	1
ncftp	5	1	0
neptune	4	3	1
netbus	3	2	3
Total		45	24
netcat	4	4	1
ntfsdos	3	1	0
ntinfoscan	3	2	1
perl	4	1	0
phf	4	2	0
pod	4	4	4
portsweep	15	3	13
ppmarcro	3	0	1
processtable	4	3	1
ps	4	2	0
queso	4	0	3
satan	2	2	2
sechole	3	0	1
selfping	3	1	0
sendmail	2	1	1
smurf	5	5	5
snmpget	4	1	0
tcpreset	3	1	0
teardrop	3	3	3
udpstorm	2	2	2
warez	4	2	1
xlock	3	3	1
xsnoop	3	2	0
xterm1	3	1	0
yaga	4	3	0
Total		49	40

6. Conclusion

Using the topic model, we propose a network intrusion detection scheme in this paper. Our scheme proposes a way to analyze network traffic using the LDA model. Packet features are employed to turn network traffic into documents, and the LDA model is used to learn the normal traffic behavior. Experiments on standard dataset are carried out using our method, and the experiment results show the efficiency of our method in detecting network intrusions. Our method can build normal behavior rules automatically for network in advance and then protect network traffic. It is suitable to be used in the networks where there are multiple data formats and data origins, and thus it provides a way of security protection to mobile edge computing.

Data Availability

The DARPA 1999 dataset used to support the findings of this study is included within the article.

Disclosure

Part of this study was finished during the first author's work with Duke University.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank National Defense Pre-Research Plan for the 13th Five-Year Project no. 90407180012, National Science Foundation of China (no. 61771361), Scientific Plan Project of Shaanxi Province (no. 2020JQ-319), and Fundamental Research Funds for the Central Universities (no. JB181503) for funding.

References

[1] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[2] T. Dbouk, A. Mourad, H. Otrok, H. Tout, and C. Talhi, "A novel ad-hoc mobile edge cloud offering security services through intelligent resource-aware offloading," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1665–1680, 2019.

[3] A. Sperotto, M. Mandjes, R. Sadre, P.-T. de Boer, and A. Pras, "Autonomic parameter tuning of anomaly-based IDSs: an SSH case study," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 128–141, 2012.

[4] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi et al., "Adaptive computation offloading with edge for 5G-envisioned Internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[5] X. Xu, B. Shen, X. Yin, M. R. Khosravi, H. Wu et al., "Edge server quantification and placement for offloading social media services in industrial cognitive IoV," *IEEE Transactions on Industrial Informatics*, 2020.

[6] P.-F. Marteau, "Sequence covering for efficient host-based intrusion detection," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 994–1006, 2019.

[7] C. J. Fung, J. Zhang, and R. Boutaba, "Effective acquaintance management based on bayesian learning for distributed

intrusion detection networks," *IEEE Transactions on Network and Service Management*, vol. 9, no. 3, pp. 320–332, 2012.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[9] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. 1, pp. 5228–5235, 2004.

[10] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for unix processes," in *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, IEEE, Los Alamitos, CA, USA, pp. 120–128, May 1996.

[11] Y. Liao and V. R. Vemuri, "Using text categorization techniques for intrusion detection," in *Proceedings of the 11th USENIX Security Symposium*, pp. 51–59, San Francisco, CA, USA, 2002.

[12] Y. Ding, X. Yuan, D. Zhou, L. Dong, and Z. An, "Feature representation and selection in malicious code detection methods based on static system calls," *Computers & Security*, vol. 30, no. 6-7, pp. 514–524, 2011.

[13] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns," *IEEE Transactions on Computers*, vol. 63, no. 4, pp. 807–819, 2014.

[14] F. Maggi, M. Matteucci, and S. Zanero, "Detecting intrusions through system call sequence and argument analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 4, pp. 381–395, 2010.

[15] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 4, pp. 227–261, 2000.

[16] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer Networks*, vol. 31, no. 23-24, pp. 2435–2463, 1999.

[17] P. A. Porras and P. G. Neumann, "Emerald: event monitoring enabling responses to anomalous live disturbances," in *Proceedings of the 20th National Information Systems Security Conference*, IEEE, Baltimore, MD, USA, pp. 353–365, 1997.

[18] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: results from the JAM project," in *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition*, IEEE, Hilton Head, SC, USA, pp. 130–144, January 2000.

[19] "KDD cup 1999 data," 1999, https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[20] "DARPA intrusion detection evaluation dataset," 1998, https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset.

[21] Y. Wang, "A multinomial logistic regression modeling approach for anomaly intrusion detection," *Computers & Security*, vol. 24, no. 8, pp. 662–674, 2005.

[22] Y. Wang, I. Kim, G. Mbateng, and S.-Y. Ho, "A latent class modeling approach to detect network intrusion," *Computer Communications*, vol. 30, no. 1, pp. 93–100, 2006.

[23] L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a hidden naïve Bayes multiclass classifier," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13492–13500, 2012.

[24] S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," *Procedia Technology*, vol. 4, pp. 119–128, 2012.

[25] S. Jiang, X. Song, H. Wang, J.-J. Han, and Q.-H. Li, "A clustering-based method for unsupervised intrusion detections," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 802–810, 2006.

[26] Y. Li, B. Fang, l. Guo, and Y. Chen, "Network anomaly detection based on TCM-KNN algorithm," in *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security*, ACM, Singapore, pp. 1–19, 2007.

[27] C.-F. Tsai and C.-Y. Lin, "A triangle area based nearest neighbors approach to intrusion detection," *Pattern Recognition*, vol. 43, no. 1, pp. 222–229, 2010.

[28] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: an intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Systems*, vol. 78, pp. 13–21, 2015.

[29] A. I. Saleh, F. M. Talaat, and L. M. Labib, "A hybrid intrusion detection system (HIDS) based on prioritized *k*-nearest neighbors and optimized SVM classifiers," *Artificial Intelligence Review*, vol. 51, no. 3, pp. 403–443, 2019.

[30] G. Liu, Z. Yi, and S. Yang, "A hierarchical intrusion detection model based on the PCA neural networks," *Neurocomputing*, vol. 70, no. 79, pp. 1561–1568, 2007.

[31] A. N. Toosi and M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers," *Computer Communications*, vol. 30, no. 10, pp. 2201–2212, 2007.

[32] S. Rezvy, M. Petridis, A. Lasebae, and T. Zebin, "Intrusion detection and classification with autoencoded deep neural network," *Innovative Security Solutions for Information Technology and Communications*, pp. 142–156, Springer, Berlin, Germany, 2019.

[33] J. Kim, J. Kim, H. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Proceedings of the 2016 International Conference on Platform Technology and Service*, IEEE, Jeju, Republic of Korea, January 2016.

[34] C.-H. Tsang, S. Kwong, and H. Wang, "Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection," *Pattern Recognition*, vol. 40, no. 9, pp. 2373–2391, 2007.

[35] N. N. P. Mkuzangwe and F. V. Nelwamondo, "A Fuzzy logic based network intrusion detection system for predicting the TCP SYN flooding attack," *Intelligent Information and Database Systems*, pp. 14–22, Springer, Berlin, Germany, 2017.

[36] M. V. Mahoney and P. K. Chan, "Learning nonstationary models of normal network traffic for detecting novel attacks," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Edmonton, Canada, pp. 376–385, July 2002.

[37] M. V. Mahoney, "Network traffic anomaly detection based on packet bytes," in *Proceedings of the 2003 ACM Symposium on Applied Computing*, ACM, Melbourne, FL, USA, pp. 346–350, 2003.

[38] T. Shon, X. Kovah, and J. Moon, "Applying genetic algorithm for classifying anomalous tcp/ip packets," *Neurocomputing*, vol. 69, no. 1618, pp. 2429–2433, 2006.

[39] S. B. Shamsuddin and M. E. Woodward, "Modeling protocol based packet header anomaly detector for network and host intrusion detection systems," in *Proceedings of the 6th International Conference on Cryptology and Network Security*, Springer, Singapore, pp. 209–227, 2007.

[40] W. Yassin, N. I. Udzir, A. Abdullah, M. T. Abdullah, Z. Muda, and H. Zulzalil, "Packet header anomaly detection using statistical analysis. International Joint Conference SOCO14-CISIS14-ICEUTE14," *Advances in Intelligent Systems and*

Computing, vol. 299, pp. 473–482, Springer, Berlin, Germany, 2014.

[41] I. Hareesh, S. Prasanna, M. Vijayalakshmi, and S. M. Shalinie, "Anomaly detection system based on analysis of packet header and payload histograms," in *Proceedings of the 2011 International Conference on Recent Trends in Information Technology (ICRTIT)*, IEEE, Chennai, India, pp. 412–416, June 2011.

[42] P. Manandhar and Z. Aung, "Towards practical anomaly-based intrusion detection by outlier mining on TCP Packets," *Database and Expert Systems Applications*, pp. 164–173, Springer, Berlin, Germany, 2014.

[43] C. Cramer and L. Carin, "Bayesian topic models for describing computer network behaviors," in *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Prague, Czech Republic, 2011.

[44] E. M. Ferragut, D. M. Darmon, C. A. Shue, and S. Kelley, "Automatic construction of anomaly detectors from graphical models," in *Proceedings of the 2011 IEEE Symposium on Computational Intelligence in Cyber Security*, IEEE, Paris, France, 2011.

[45] J. Huang, Z. Kalbarczyk, and D. M. Nicol, "Knowledge discovery from big data for intrusion detection using LDA," in *Proceedings of the 2014 IEEE International Congress on Big Data*, IEEE, Washington, DC, USA, pp. 760-761, 2010.

[46] H. J. Steinhauer, T. Helldin, G. Mathiason, and A. Karlsson, "Topic modeling for anomaly detection in telecommunication networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2019, 2019.

[47] S. Lee, S. Kim, S. Lee et al., "LARGen: automatic signature generation for malwares using latent Dirichlet allocation," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 771–783, 2018.

[48] D. Mimno and D. Blei, "Bayesian checking for topic models," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, IEEE, Edinburgh, Scotland, pp. 227–237, 2011.

[49] Wireshark, "Go deep," 2020, https://www.wireshark.org/.

[50] RFC 791, "Internet protocol," 1981, https://tools.ietf.org/html/rfc791#page-11.

[51] RFC 793, "Transmision control protocol," 1981, https://www.ietf.org/rfc/rfc793.txt.

[52] "Darpa intrusion detection evaluation dataset," 1999, https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8867094, 9 pages
https://doi.org/10.1155/2020/8867094

Research Article

A Task Offloading Method with Edge for 5G-Envisioned Cyber-Physical-Social Systems

Jielin Jiang,[1,2,3] Xing Zhang,[1] and Shengjun Li[4]

[1]*School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China*
[2]*Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing, China*
[3]*Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing, China*
[4]*School of Information Science and Engineering, Qufu Normal University, Rizhao, China*

Correspondence should be addressed to Shengjun Li; qfnulsj@163.com

Received 1 June 2020; Revised 5 July 2020; Accepted 21 July 2020; Published 7 August 2020

Academic Editor: Dou Wanchun

Recently, Cyber-Physical-Social Systems (CPSS) have been introduced as a new information physics system, which enables personnel organizations to control physical entities in a reliable, real-time, secure, and collaborative manner through cyberspace. Moreover, with the maturity of edge computing technology, the data generated by physical entities in CPSS are usually sent to edge computing nodes for effective processing. Nevertheless, it remains a challenge to ensure that edge nodes maintain load balance while minimizing the completion time in the event of the edge node outage. Given these problems, a Unique Task Offloading Method (UTOM) for CPSS is designed in this paper. Technically, the system model is constructed firstly and then a multi-objective problem is defined. Afterward, Improving the Strength Pareto Evolutionary Algorithm (SPEA2) is utilized to generate the feasible solutions of the above problem, whose aims are optimizing the propagation time and achieving load balance. Furthermore, the normalization method has been leveraged to produce standard data and select the global optimal solution. Finally, several necessary experiments of UTOM are introduced in detail.

1. Introduction

For the past few years, with the perpetual progress of Big Data, Cloud Computing, Internet of things (IoT), and other technologies, traditional physical systems and new information resources are further integrated, thus forming complex systems that incorporate machines, information, and human, namely, CPSS. They enable the physical system to have the functions of computation, communication, remote cooperation, etc., and make full use of social information and computing resources to carefully coordinate the physical system. CPSS make the machine more intelligent, as well as the personnel organization operating the physical entity in a more reliable, real-time, and security manner through the cyberspace. This also makes the development of the IoT more rapid and a variety of intelligent scene applications are broader [1, 2].

Nevertheless, with the increasing development of mobile devices, IoT devices, and multiple intelligent scenes (e.g., intelligent transportation, intelligent home, and intelligent cities), an increasing number of people have higher standards for applications in these scenarios [3]. When the physical system combines network information and social information, the mass data transmission characterized by multiple types and high speed also puts forward higher requirements for network communication (i.e., higher bandwidth and lower delay). This case conflicts with people's higher requirements for high-quality, low latency, and real-time network services. Thus, academia and industry urgently need to solve the problem of how to systematically and efficiently process the data, i.e., the historical data and the local real-time data, in CPSS. However, it makes little sense to consider the service outside the context of network performance. The 5G network with the purpose of accelerating the evolution of smart applications scene can not only enhance the data delivery rates and lessen the latency but also advance the amounts of infrastructures in intelligent applications.

Technically, the implementation of the 5G network needs the support of edge computing technique [4]. Edge computing is an inevitable development in the evolution of base stations combined with IT as well as mobile network [5]. The most intuitive benefit that edge computing brings is the ability to improve the quality of experience through high bandwidth and instant response [6]. At the same time, quality of experience is becoming more prominent among the booming new services, which have become an essential part of mobile social and entertainment [7, 8].

To offer immediate and efficient feedback for the users in CPSS, there is no doubt that edge computing, a significant paradigm, with abundant computing resources, needs to be adequately taken advantage of for users to experience a high quality of service applications in real time [9]. It makes the user close to the nodes geographically where the resource is being processed, thus significantly reducing the delay of offloading tasks. Specifically, in CPSS, the base stations are evolved into edge nodes to service the task requests and data from the users who are covered in these nodes. In addition to its advantages in offloading tasks, edge computing shortens the distance between people and processing nodes, making the traditional interception of information much less likely to cause harm to users, thus improving the user's security [10].

However, in the hybrid CPSS scenario, where multiple systems are involved, offloading tasks to reasonable nodes is a complex problem [11]. Thus, how to determine the offloading node of the computing tasks is a challenge in the CPSS scenario [12]. Also, since the number of nodes is limited and each node has restricted computing resources, resource utilization needs to be taken into consideration and to be promoted as much as possible [13, 14]. Under the premise of improving performance as much as possible, load balance, as a critical indicator, should be taken into account to ensure the stability of each node, because it reflects the overall efficiency and performance of the system [15, 16].

Based on the above discussion and questions, a unique task offload method in CPSS based on the technique of edge computing, namely, UTOM, is presented in this paper to optimize the offloading strategy to get the minimum delay and achieve load balance.

Specifically, the pivotal motivations and contributions of this paper are shown below:

(i) Few studies research on the offloading methods to pursue the minimum completion time and load balance variance with the consideration of the particular CPSS scenario. So a unique task offload method in CPSS based on the technique of edge computing is presented in this paper.

(ii) The evolutionary algorithm, Improving the Strength Pareto Evolutionary Algorithm (SPEA2), and normalization method, Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS), are deployed to obtain the feasible offloading strategies and select the optimal strategy.

(iii) Sufficient experimental comparisons and assessment analysis with traditional methods confirmed the effectiveness of UTOM.

The rest of our paper is presented as follows: Section 2 shows the related work of our paper. Section 3 presents the system model based on the CPSS combined with 5G-envisioned edge computing. The process of UTOM based on the MOEA with edge is elaborated in Section 4. Section 5 shows the evaluations of UTOM and demonstrates the effectiveness of this method. Conclusion and future work are presented in Section 6.

2. Related Work

The characteristics of edge computing, i.e., sufficient memory capacity and higher computing power, are prominent in CPSS [17]. The related researches about CPSS and its prominent advantages have been extensively studied in some previous pieces of literature.

CPSS has many applications due to its real time, diversity, high reliability, and other advantages. In [18], Wang systematically described how CPS is translated into CPSS, the definition and classification as well as applications of CPSS, the contribution and significance of CPSS, and how CPSS connects and functions with different entity worlds. Relevant work has been carried out in the Internet field combining scenes such as the IoT. Han et al. proposed to introduce dynamic and manifold human behaviour into the vehicle network to make it become a CPSS system and protocoled it as a parallel vehicle network, to achieve more stable and high-efficiency traffic state and ultra-low data communication delay between vehicles [19]. Wang et al. put forward a new unified method of CPSS framework based on cloud parallel driving, which aims at collaborative online automatic driving, and developed parallel testing, learning, and reinforcement learning for this framework [20].

Given the large amount of "4V" data in the CPSS scenario, it is difficult to effectively and timely solve the demand for these data traditionally [21]. Thus, we combine the edge computing with CPSS to address and solve the mentioned problems [22, 23]. Edge computing brings the advantages of cloud computing into various application scenarios of CPSS and provides efficient services similar to cloud services on the edge of the CPSS network [24].

The offloading strategies have become more efficient and adaptive by utilizing edge computing. Mach and Becvar presented a survey, in which they divide the research of computing load into three key areas: decision-making of computing load, allocation of computing resources in MEC, and mobile management, and this survey provides relevant research direction [25]. Advanced algorithms are proposed to solve offloading problems in some literatures. A low complexity online algorithm is developed by Mao et al., which only depends on the instantaneous side information and does not need to calculate the task request distribution information, and the algorithm determines the decision of unloading as well as the calculation of the transmission power of unloading [26]. Wang et al. proposed an innovative framework, which is developed to improve edge computing performance, and an optimal resource unloading scheme, which is designed to minimize the overall energy consumption of access nodes [27]. Wang et al. got the optimal solution based on transforming the energy consumption minimization problem into a convex problem and proposed a single variable search local optimal algorithm for the non-convex as well as non-smooth problem of delay minimization to obtain the optimal results [28]. Some unique solutions are also proposed to solve the problems in the offloading strategy. Chen and Hao reduced the task offloading problem to NP-hard problem and designed an efficient scheme to solve the task placement and resource allocation sub-problems [29].

However, most of existing studies only focus on one point in CPSS scenario and offloading method without considering them together. Indeed, compared with the previous works, this paper designed a CPSS-based offloading strategy named UTOM, whose purpose is offloading the tasks from the users in the condition of minimum time consumption and load balance variance.

3. System Model

Firstly, the framework of the task offloading model based on CPSS in 5G scene is presented in Figure 1. Secondly, the propagation delay model and load balance model for offloading strategies are designed. Thirdly, the task offloading problem has been defined as a multi-objective problem. Table 1 shows some key terms and their descriptions.

3.1. Task Offloading Resource Model. In a 5G scene based on CPSS, base stations are arranged to offer efficient services for task requesters. In general, the coverage of base stations includes some micro base stations with the aim of improving access speed and service efficiency. The micro base stations receive service or data request tasks from mobile handheld devices by using wireless signals.

As shown in Figure 1, the diagram briefly describes two scenarios: the online social scenario and the actual commuting scenario. In online social scenario, social contact data, app data, and IoT device data will be generated. Traffic data, motion data, trajectory data, and so on will be generated in the actual commuting scenario. In these scenarios, there exist a huge number of users around the base station to receive data. The purpose of this paper is to study how to offload the service data to the appropriate edge server with the purpose of minimizing the delay and keeping the load balance of the server.

It is assumed that there are quantifiable servers in the framework of task offloading in this section. Denote the task requester collections as $R = \{r_1, r_2, \ldots, r_M\}$, where M represents that there are M hypothetical task requesters in this scene. The scene assumes that each requester has only one computing task waiting to be processed. Denote the base station collection as $MB = \{mb_1, mb_2, \ldots, mb_N\}$, where N represents the number of base stations. Each base station only accepts task requests within its coverage. Then, the base station transfers its received tasks to the edge server to which it belongs. The edge server collection is denoted as $ES = \{es_1, es_2, \ldots, es_K\}$, where K represents that there are K edge servers in this framework.

3.2. Propagation Delay Model. This paper assumes that the overall propagation delay consists of four parts in all. VB_k^n is defined to judge whether the n-th $(n = 1, 2, \ldots, N)$ base station mb_n is combined with the k-th $(k = 1, 2, \ldots, K)$ edge server es_k.

$$VB_k^n = \begin{cases} *20c1, & \text{if } mb_n \text{ combine with } es_k, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The first part of overall propagation delay is propagation time for transferring task from the base station to the target edge server, which is calculated by

$$PT_n(X) = \left(1 - VB_n^k\right) \frac{ST_n \cdot \lambda_n}{TR}, \tag{2}$$

where ST_n represents the size of the task request coming from the base station mb_n in the coverage of edge servers es_k . Besides, λ_n signifies the number of the passing base stations in the process of task propagation, and TR denotes the propagation rate between base stations.

The second part is execution time for coping with the task requests coming from mb_n, which is defined as

$$CT_n(X) = \frac{ST_n}{AV_n \cdot PV}, \tag{3}$$

where AV_n represents the resource units VMs demanded by the task of mb_n and PV represents the processing power of each unit in VM.

The third part is average wait time of the task request in edge server, which is calculated by

$$AT_n(X) = \frac{WL}{AR}, \tag{4}$$

where AR represents the arrival rate of task and WL represents average wait length of the task request.

The fourth part is the return time of processing result coming from es_k, which is obtained by

$$RT_n(X) = \frac{ST_n'}{AV}, \tag{5}$$

where ST_n' signifies the task size of the results offloaded from mb_n.

The total propagation delay for responding to one certain task request is defined as

$$OT_n(X) = PT_n(X) + CT_n(X) + AT_n(X) + RT_n(X). \tag{6}$$

The average delay for responding to all the task requests is calculated by

$$\text{AVE} = \frac{1}{N} \sum_{n=1}^{N} OT_n(X). \tag{7}$$

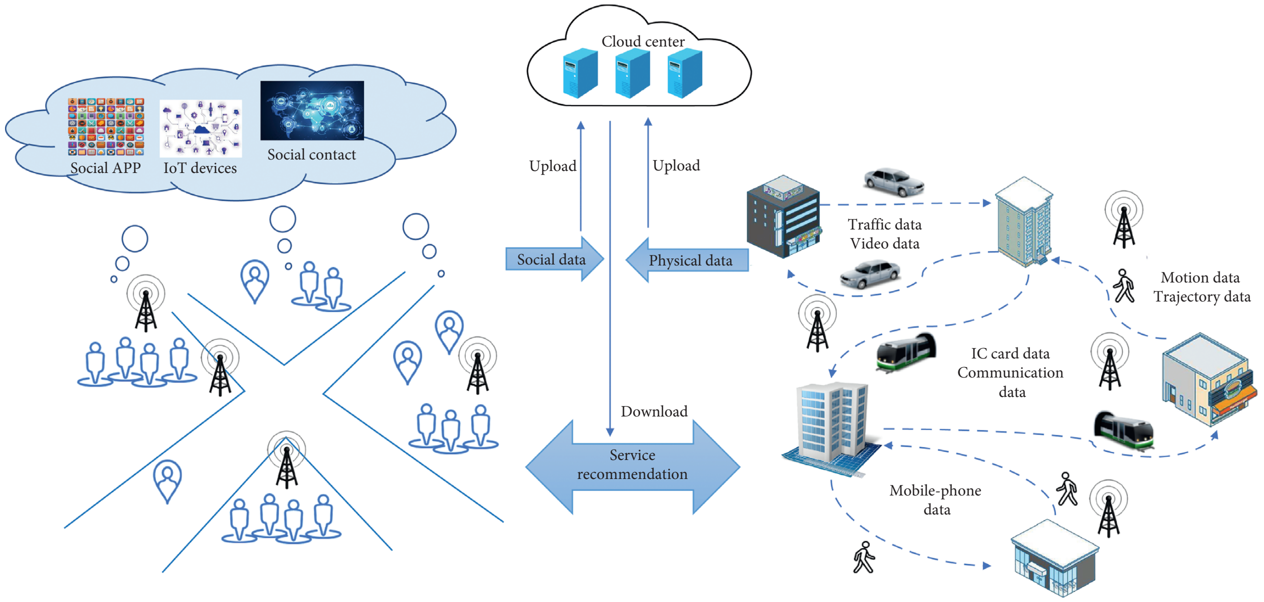


Figure 1: A task offloading framework based on CPSS with edge.

Table 1: Key terms and relevant descriptions.

Key terms	Relevant descriptions
ES	Edge server set, $ES = \{es_1, es_2, \ldots, es_K\}$
MB	Base station set, $MB = \{mb_1, mb_2, \ldots, mb_N\}$
R	A set of task requesters, $R = \{r_1, r_2, \ldots, r_M\}$
K	The size of the edge servers
N	The size of the base stations
M	The size of the task requesters
PV	The processing power of each unit in VM
C_{vm}	The number of the VMs in edge servers
TR	The propagation rate between base stations
AR	The arrival rate of task requests

3.3. Load Balance Model. This paper aims to investigate which edge server is the best node to offload. During the process of searching offloading strategies, the load balance is an important factor which assesses the reliability of the designed model. By utilizing of virtualized technique, the usage of virtual machine (VM) instances could be leveraged to obtain the load balance variance for all the edge servers in the offloading strategy.

SB_k is defined to estimate whether es_k has been occupied, which is acquired by

$$SB_k = \begin{cases} 1, & \text{if } es_k \text{ has been occupied,} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Besides, MB_n^k is defined to estimate whether the task in mb_n has been offloaded to es_k. It is acquired by

$$MB_k^n = \begin{cases} 1, & \text{if } mb_n \text{ offloads the task to } es_k, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Thus, the number of running edge servers is defined as

$$A_s = \sum_{n=1}^{N} MB_k^n. \tag{10}$$

The corresponding resource utilization of es_k is calculated by the utilization number of VM, which is defined as

$$R_k = \frac{1}{C_{vm}} \sum_{n=1}^{N} MB_k^n \cdot \varepsilon_n, \tag{11}$$

where ε_n represents the number of VMs required by the task requests in mb_n.

Then, the overall average resource utilization of the edge servers is obtained by

$$AU_R = \frac{1}{A_s} \sum_{k=1}^{K} R_k. \tag{12}$$

According to the different resource utilization in each edge server, the load balance variance of es_k is calculated by

$$BV_{\text{ave}} = \left(R_k - AU_R\right)^2. \tag{13}$$

At last, the overall load balance variance of the occupied edge servers in the offloading scene is calculated by

$$LB = \frac{1}{A_s} \sum_{k}^{K} SB_k \cdot BV_{\text{ave}}. \tag{14}$$

3.4. Problem Formulation. The objective functions of this system model have been presented in (7) and (14), which is expected to improve the overall effectiveness in the task offloading scenario. The objective problems are formulated as follows:

$$\min LB, \tag{15}$$

$$\min \text{AVE}, \tag{16}$$

$$\text{s.t.} \quad AV_n \leq C_{vm} \left(n \in \{1, 2, \ldots, N\}\right). \tag{17}$$

The constraint that the number of VMs requested in any task must be less than the number of VMs in the edge server has been defined in (17).

4. The Tasks Offloading Method

As shown in (7) and (14), in this segment, a unique task offloading means based on the MOEA with CPSS, called UTOM, is designed with the aim of minimizing the propagation delay and the load balance variance. The process of UTOM, i.e., searching the feasible solutions, normalizing the solutions, and selecting the optimal strategy, is presented in detail.

4.1. Offloading Strategy Option by SPEA2. The multi-objective offloading model based on SPEA2 is presented in problem formulation. SPEA2 utilizes an advanced fitness allocation strategy, which takes into account the number of individuals controlled by each individual [30]. Besides, it integrates the nearest neighbour density estimation technology, allowing more accurate guidance in the search process. Given these advantages, SPEA2 is utilized to be adopted in this method to figure out the double-objective optimization problems. The related fitness functions and constraints are encoded first in this process. Then, the selection process including environment selection and pairing selection is applied. Finally, the advanced evolutionary operators are performed to generate solutions.

(1) Encoding: First of all, we need to map the problem to be solved, i.e., minimizing the time consummation in (7) and the load balance variance in (14), into a mathematical problem. The solution of the problem is represented by a coded string of numbers, and the genetic operators operate on the string directly. There are many coding ways, and here floating-point coding is selected. Because we have high precision requirements for the results to be generated, the solution space will increase dramatically when utilizing integer coding. In addition, the floating-point coding method is easy to deal with the complex constraints of decision variables, which have been presented in the model part.

(2) Fitness functions and constraints: In the genetic algorithm, fitness function plays the role of selecting excellent individuals. According to the fitness value of the individual, it selects the individuals to be inherited to the next generation. In this method, the fitness function is transformed according to the objective functions (7) and (14). The practical constraint has been shown in (17), which represents that the number of VMs requested for any task must be less than the number of VMs in the edge servers.

(3) Selection operator: Selection refers to the operation of selecting excellent individuals from the group and eliminating the inferior ones. It is based on the evaluation of fitness. The larger the fitness, the greater the possibility of being selected, the number of his “offspring” in the next generation, and the selected individuals will be put into the matching database. This method selects a roulette operator, which ensures the individuals whose fitness function is better would be selected into the next generation as far as possible while ensuring that all individuals are likely to be selected.

(4) Crossover and mutation operators: The purpose of the crossover is to improve the searchability of the genetic algorithm of leap in the next generation of the new individual through the crossover operation. Crossing is an important method of genetic algorithm to obtain excellent individuals. The probability of crossover operation is in accordance with the random selection of two individuals in the library, and the cross-location is random. Indeed, single-point crossover is applied in UTOM.

The basic process of mutation operation is as follows: generate a random number $ran\ d$ between 0 and 1 and mutation probability pm. If $ran\ d > pm$, the mutation operation would be performed. Mutation operator itself is a kind of local random search, and it has the ability to avoid some of the permanent loss of information due to selection and crossover operators together with the selection and crossover operators. It makes the genetic algorithm maintain the population diversity, while avoiding premature convergence. In mutation operation, the probability of mutation should not be too large. If $pm > 0.5$, the genetic algorithm will degenerate into random search.

4.2. Optimal Strategy Selection by TOPSIS. Figure 2 shows how to utilize TOPSIS to derive the optimal strategy based on the strategies generated by SPEA2. TOPSIS generates results which accurately reflect the gap between the evaluation schemes [31]. Then, the best result could be gained by comparing these gaps. The symbols utilized in the flowchart are described as follows [32].

The initial strategies produced by SPEA2 are represented by P, and these strategies form two sets, i.e., propagation time strategies T and load balance variance strategies L, where $T = \{T_1, T_2, \ldots, T_P\}$ and $L = \{L_1, L_2, \ldots, L_P\}$.

The standardized values of propagation time and load balance variance are defined as $CT_p(X)$ and $CL_p(X)$. Two weights of the indicators are defined as μ_T and μ_L. Then, the standardized weight decision values are presented as WT_p and WL_p. Afterwards, the degree of closeness between alternatives and the best solution as well as the worst solution are measured as OS_p and WS_p. Next, the comprehensive evaluation value of the best solution and the worst solution is presented as C_r. At last, the best solution S_{idea} would be obtained from all the strategies.

4.3. The Overview of UTOM. The purpose of UTOM method is realizing the optimization of the objective functions presented in the system model. The overview of our UTOM is presented in Algorithm 1. In this algorithm, the scale of the population is I, the maximum amount of inheritance is J,

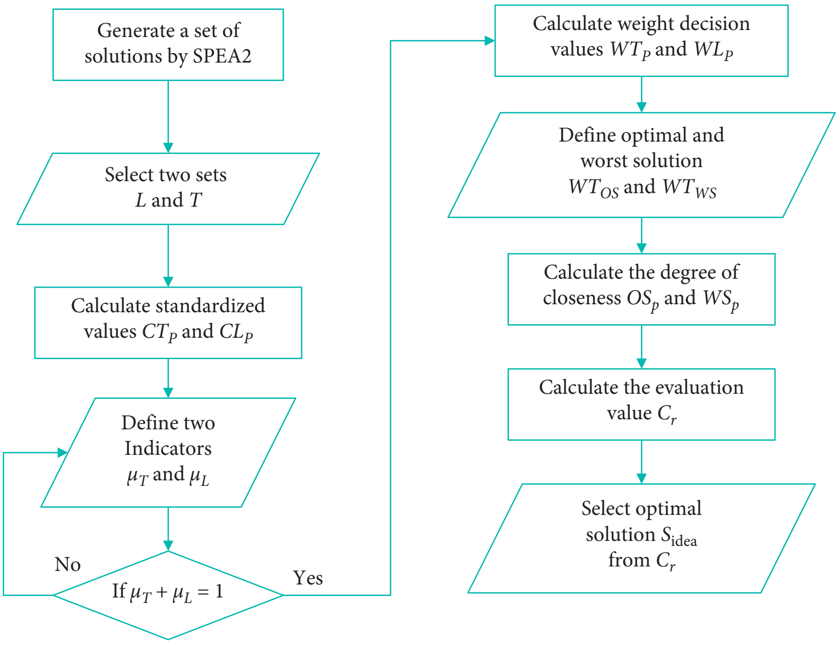

Generate a set of solutions by SPEA2
Select two sets L and T
Calculate standardized values CT_P and CL_P
Define two Indicators μ_T and μ_L
No
If $\mu_T + \mu_L = 1$
Yes
Calculate weight decision values WT_P and WL_P
Define optimal and worst solution WT_{OS} and WT_{WS}
Calculate the degree of closeness OS_p and WS_p
Calculate the evaluation value C_r
Select optimal solution S_{idea} from C_r

Figure 2: The flowchart of utilizing TOPSIS to derive the optimal strategy.

```
**Require** $J$, $I$ **Ensure** BS
Obtain tasks from task requesters
Initialize strategies $P$
**For** $(j = 1, 2, \ldots, J)$
  $i = 1$
    **While** $(i < \ = I\}$
    Execute the crossover, selection, and mutation operators to produce offspring
    **For** (individuals in population)
      Obtain total propagation time from (7)
      Obtain load balance variance from (14)
    **End For**
    Execute environmental selection operator
    $i = i + 1$
  **End While**
  Acquire the utility values by TOPSIS
  Obtain the best solution $BS$
**End For**
**Return** $BS$
```

Algorithm 1: Obtaining the best strategy by utilizing UTOM.

and the exportation of UTOM is the best strategy *BS*. Firstly, the initial strategies are produced randomly and presented as *P*. Then, feasible solutions are generated after *J* iterations by SPEA2 through fitness functions. Finally, TOPSIS is applied to calculate standardized values and select optimal value.

Table 2: Experimental variable setting.

Experimental variable	Value
The scale of tasks	{50, 100, 150, 200, 250}
The size of edge services	[0.5, 0.8]
The scale of running VMs	[1, 6]
TR	620 MB/s
AR	120 MB/s

5. Experimental Evaluation

Our paper utilizes the MECHREVO-Ti2 as the experimental station. The computer parameters are as follows: the CPU is Intel i7-6700H @ 2.6 GHz, the RAM is 8 GB, and the hard disk is 1T. Some experimental parameters and their values used in this section are shown in Table 2. To prove the effectiveness of this method, some traditional methods, i.e., Benchmark, First Fit Decreasing-based task offloading with time-saving and resource utilization optimization (FFD), and Best Fit Decreasing-based task offloading with time saving and resource utilization optimization (BFD), are utilized in this section. Benchmark supposes that the VM in the initial edge node falls short with regard to the requirement of task; the certain task would not be coped with the other node. FFD gives the required amount of the VM in the task, and the tasks will be ranked in an order. Then, the initial task will be offloaded to the initial node. BFD will sort all the tasks by descending order. Afterwards, the initial task will be offloaded to the initial node. Benchmark, FFD, and BFD will be uniformly referred to as the “three classical methods.”

5.1. Comparison of Experimental Results on the Employed Number of Edge Servers. The number of the employed edge servers by Benchmark, FFD, BFD, and UTOM is presented in Figure 3. It is obvious that the number of the employed edge servers is smaller than that of the other three methods. When the number of the evaluated tasks equals 100, the gap between the four methods is relatively little. The difference between the four means it begins to increase with the increasing of the task amount. It means that the UTOM has a better performance in higher number of tasks. These data show that the performance of our UTOM is the best in the comparison, while the performance of Benchmark is the worst.

5.2. Comparison of Experimental Results on the Average Propagation Time of Tasks. The average time equals the entire time divided by the number of tasks. Correspondingly, the average propagation time of tasks is calculated. The average propagation time keeps increasing with the enlarging of the task amount. The time calculated by UTOM is lower than the other three methods. The average propagation time of UTOM in different scale of tasks is 0.13, 0.22, 0.37, 0.48, and 0.59 (s) when the number of the tasks equals 50, 100, 150, 200, and 250. Besides, the difference of the three classical methods between UTOM is shown in Figure 4.

5.3. Comparison of Experimental Results on the Overall Propagation Time of Tasks. The entire time consists of four parts, which are the propagation time, the wait time, the execution time, and the return time. The total time represents the satisfaction of the users. It is conducted that the total time obtained by our UTOM is lower than the other methods by analysing Figure 5. The total time of UTOM is 6.49, 22.00, 56.10, 95.50, and 147.94 (s) when the number of the tasks equals 50, 100, 150, 200, and 250.

5.4. Comparison of Experimental Results on the Average Resource Utilization. The average resource utilization is not the objective function in our paper, while this index is another crucial evaluation parameter in the experimental comparison. This index represents the employed amount of the VM in the edge servers and is expected to get a high value in the experiment. Figure 6 presents the performance comparison of average resource utilization by utilizing of UTOM and three classical methods severally. The Benchmark performs worst in the comparison, and UTOM performs better than the other two means.

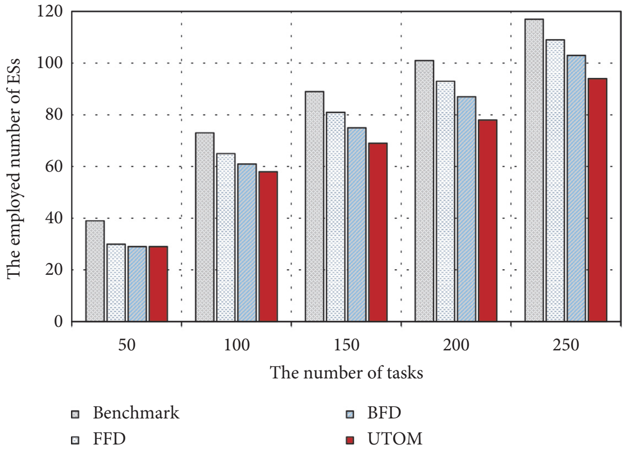


Figure 3: Experimental results on the employed scale of edge servers by UTOM and three classical methods.

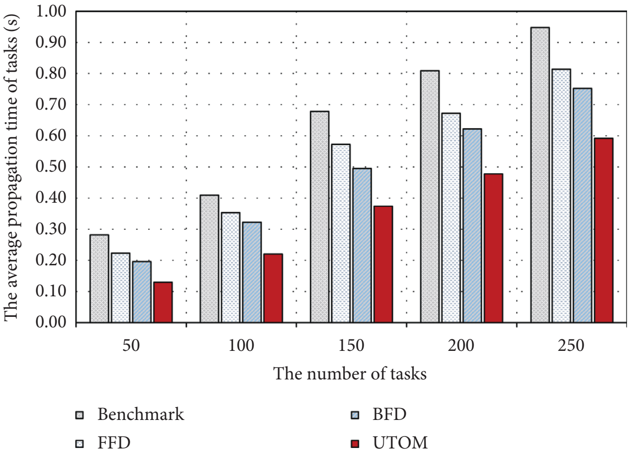


Figure 4: Experimental results on the average propagation time of tasks by UTOM and three classical methods.

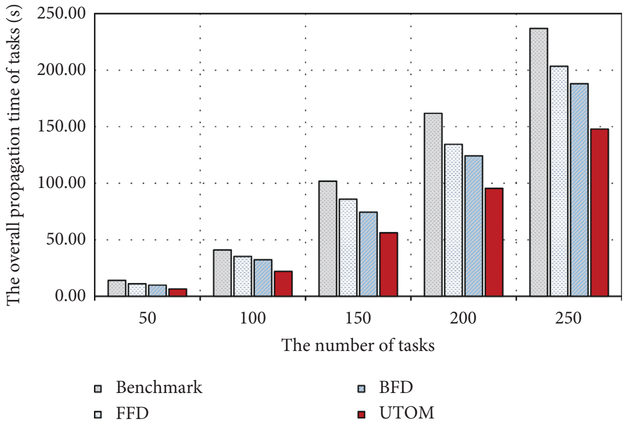


Figure 5: Experimental results on the entire propagation time of tasks by UTOM and three classical methods.

5.5. Comparison of Experimental Results on the Load Balance Variance. The load balance variance is the objective value in this experiment. It is summarized that the variance begins to

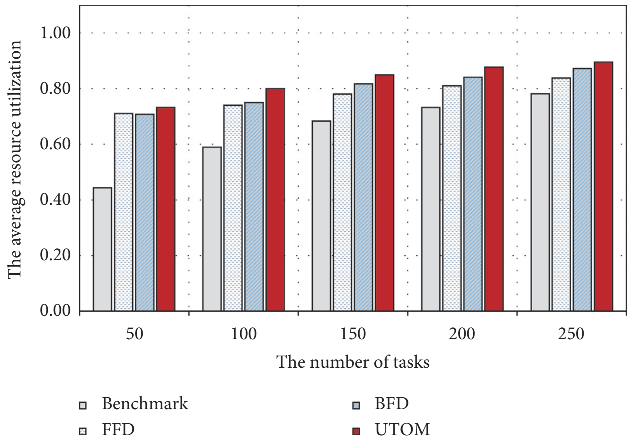


Figure 6: Experimental results on the average resource utilization by UTOM and three classical methods.

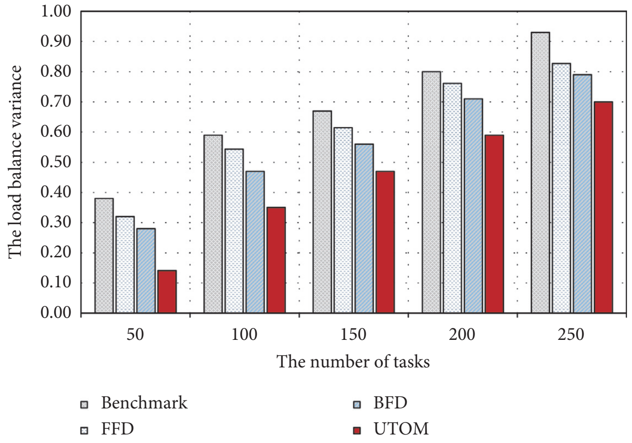


Figure 7: Experimental results on the load balance variance by UTOM and three classical methods.

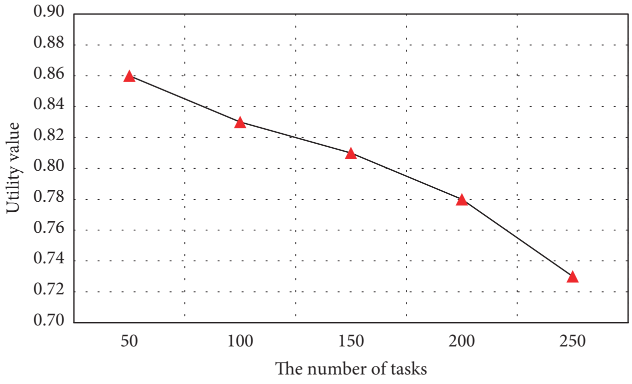


Figure 8: Experimental results on utility value by UTOM and three classical methods.

enlarge with the increasing of the amount of the tasks by analysing Figure 7. The lower value represents the better offloading strategy which the method obtained. The load balance variance of UTOM is 0.14, 0.35, 0.47, 0.59, and 0.70 when the number of the tasks equals 50, 100, 150, 200, and 250.

5.6. The Analysis of Experimental Results on the Utility Value. The different utility values of all the strategies are obtained by TOPSIS method. The certain strategy with the maximum value among the entire strategies is our best strategy. It is shown that the optimal utility value in different amount will reduce with the decreasing of the task scale by analysing Figure 8. From Figure 8, we intuitively deduce that UTOM obtains lower utility value with the improvement of task number. After the detailed statistics, the utility value of UTOM is 0.73, 0.78, 0.81, 0.83, and 0.86 when the number of the tasks equals 50, 100, 150, 200, and 250.

6. Conclusions

We devote ourselves to the problem of task offloading based on CPSS, in which edge computing technology is reasonably combined. The offloading problem is defined as an optimization problem of the minimizing of the propagation consumption and load balance variance. Furthermore, a method named UTOM is presented in this paper to optimize the offload strategy to get the minimum propagation delay and load balance variance. Besides, the normalization technique named TOPSIS is also utilized in combination to obtain standardized data. The experimental results show that the UTOM method has sufficient effectiveness and correctness. We intend to apply this method to real datasets based on CPSS to discuss the applicability in practice in future work.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Jielin Jiang conceived and designed the study. Shengjun Li performed the simulations. Xing Zhang wrote the paper. All authors reviewed and edited the manuscript. All authors read and approved the final manuscript.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61601235 and in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20160972.

References

[1] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6172–6181, 2019.

[2] W. Yu, W. Wang, P. Jiao, H. Wu, Y. Sun, and M. Tang, "Modeling the local and global evolution pattern of community structures for dynamic networks analysis," *IEEE Access*, vol. 7, pp. 71350–71360, 2019.

[3] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, 2019.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[5] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing-a key technology towards 5g," *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.

[6] H. Wu, Z. Han, K. Wolter, Y. Zhao, and H. Ko, "Deep learning driven wireless communications and mobile computing," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 4578685, 2 pages, 2019.

[7] L. Wang, L. Jiao, J. Li, and J. Gedeon, "Moera: mobility-agnostic online resource allocation for edge computing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1843–1856, 2019.

[8] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[9] X. Xu, Y. Xue, L. Qi et al., "An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles," *Future Generation Computer Systems*, vol. 96, pp. 89–100, 2019.

[10] M. Wen, K. Ota, H. Li, J. Lei, C. Gu, and Z. Su, "Secure data deduplication with reliable key management for dynamic updates in CPSS," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 4, pp. 137–147, 2015.

[11] X. Xu, S. Fu, L. Qi et al., "An iot-oriented data placement method with privacy preservation in cloud environment," *Journal of Network and Computer Applications*, vol. 124, pp. 148–157, 2018.

[12] X. Xu, Q. Liu, X. Zhang, J. Zhang, L. Qi, and W. Dou, "A blockchain-powered crowdsourcing method with privacy preservation in mobile environment," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1407–1419, 2019.

[13] H. Yu, H. Qi, and K. Li, "CPSS: A study of cyber physical system as a software-defined service," *Procedia Computer Science*, vol. 147, pp. 528–532, 2019.

[14] L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, and X. Xu, "A qos-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems," *World Wide Web*, vol. 23, no. 2, pp. 1275–1297, 2019.

[15] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, "Joint optimization of offloading utility and privacy for edge computing enabled iot," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2622–2629, 2020.

[16] L. Qi, Q. He, F. Chen et al., "Finding all you need: web apis recommendation in web of things through keywords search," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1063–1072, 2019.

[17] X. Xu, Y. Li, T. Huang et al., "An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks," *Journal of Network and Computer Applications*, vol. 133, pp. 75–85, 2019.

[18] F. Wang, "The emergence of intelligent enterprises: from CPS to CPSS," *IEEE Intelligent Systems*, vol. 25, no. 4, pp. 85–88, 2010.

[19] S. Han, X. Wang, J. J. Zhang, D. Cao, and F.-Y. Wang, "Parallel vehicular networks: a CPSS-based approach via multimodal big data in iov," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1079–1089, 2018.

[20] F.-Y. Wang, N.-N. Zheng, D. Cao, C. M. Martinez, L. Li, and T. Liu, "Parallel driving in CPSS: a unified approach for transport automation and vehicle intelligence," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 577–587, 2017.

[21] X. Xu, Q. Liu, Y. Luo et al., "A computation offloading method over big data for iot-enabled cloud-edge computing," *Future Generation Computer Systems*, vol. 95, pp. 522–533, 2019.

[22] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: a survey, use cases, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2359–2391, 2017.

[23] N. Moustafa, K.-K. R. Choo, I. Radwan, and S. Camtepe, "Outlier dirichlet mixture mechanism: adversarial statistical learning for anomaly detection in the fog," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 1975–1987, 2019.

[24] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[25] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[26] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[27] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1784–1797, 2018.

[28] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: partial computation offloading using dynamic voltage scaling," *IEEE Transactions on Communications*, vol. 64, no. 10, p. 1, 2016.

[29] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.

[30] K. Giannakoglou, D. Tsahalis, J. Periaux et al., "Spea2: improving the strength pareto evolutionary algorithm for multiobjective optimization," 2001.

[31] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, and L. Qi, "Trust-oriented iot service placement for smart cities in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4084–4091, 2019.

[32] V. T. Lokare and P. M. Jadhav, "Using the AHP and topsis methods for decision making in best course selection after HSC," in *Proceedings of the 2016 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–6, Coimbatore, India, January 2016.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8854236, 13 pages
https://doi.org/10.1155/2020/8854236

Research Article

A Multifault Diagnosis Method of Gear Box Running on Edge Equipment

Xiao-ping Zhao,[1,2] Yong-hong Zhang ,[3] and Fan Shao[3]

[1]*School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China*
[2]*Network Monitoring Centre of Jiangsu Province, Nanjing University of Information Science and Technology, Nanjing, China*
[3]*School of Automation, Nanjing University of Information Science and Technology, Nanjing, China*

Correspondence should be addressed to Yong-hong Zhang; zyh@nuist.edu.cn

Received 4 June 2020; Revised 5 July 2020; Accepted 13 July 2020; Published 3 August 2020

Academic Editor: Dou Wanchun

In recent years, a large number of edge computing devices have been used to monitor the operating state of industrial equipment and perform fault diagnosis analysis. Therefore, the fault diagnosis algorithm in the edge computing device is particularly important. With the increase in the number of device detection points and the sampling frequency, mechanical health monitoring has entered the era of big data. Edge computing can process and analyze data in real time or faster, making data processing closer to the source, rather than the external data center or cloud, which can shorten the delay time. After using 8 bits and 16 bits to quantify the deep measurement learning model, there is no obvious loss of accuracy compared with the original floating-point model, which shows that the model can be deployed and reasoned on the edge device, while ensuring real time. Compared with using servers for deployment, using edge devices not only reduces costs but also makes deployment more flexible.

1. Introduction

Gearboxes play an important role in modern machinery and equipment, which are gradually developing toward complexity, precision, and intelligence. A gearbox is composed of gears, bearings, a shaft and box body, and other parts. It has the characteristics of a compact structure, high transmission efficiency, long service life, and reliable operation. It is an indispensable general component in modern industry, including aviation, power systems, automobiles, and industrial machine tools. But because of its complex structure and high running speed in a harsh environment, it can easily break down, so the gearbox is an important factor in machine failure. Gears and bearings are two important parts of gearboxes, and they are prone to local faults due to fatigue, wear, and tear, leading to abnormal operation of gearboxes, which may cause economic losses, including damage to machines. But the performance and life of some bearings and gears are higher than expected. To repair or replace them regularly will waste manpower, material, and production resources. Using edge computing devices for diagnosis can generate faster network service response, meeting the industry's basic needs in real-time business, application intelligence, security, and privacy protection. So, using edge computing equipment [1] to monitor and diagnose mechanical equipment can effectively avoid the above situation. Consequently, the research of efficient gearbox condition monitoring and fault identification technology is of great significance for ensuring production safety, preventing and avoiding major accidents.

2. Materials and Methods

2.1. Model Compression on Edge Computing. Fault diagnosis has three main steps: feature extraction, feature dimension reduction, and classification. Traditional feature extraction generally adopts artificial methods such as wavelet transforms, statistical features, and empirical mode decomposition. PCA, ICA, and self-encoders are used to reduce the dimension of features; Bayesian and nearest neighbor classifiers are most commonly used for classification. The process is shown in Figure 1.

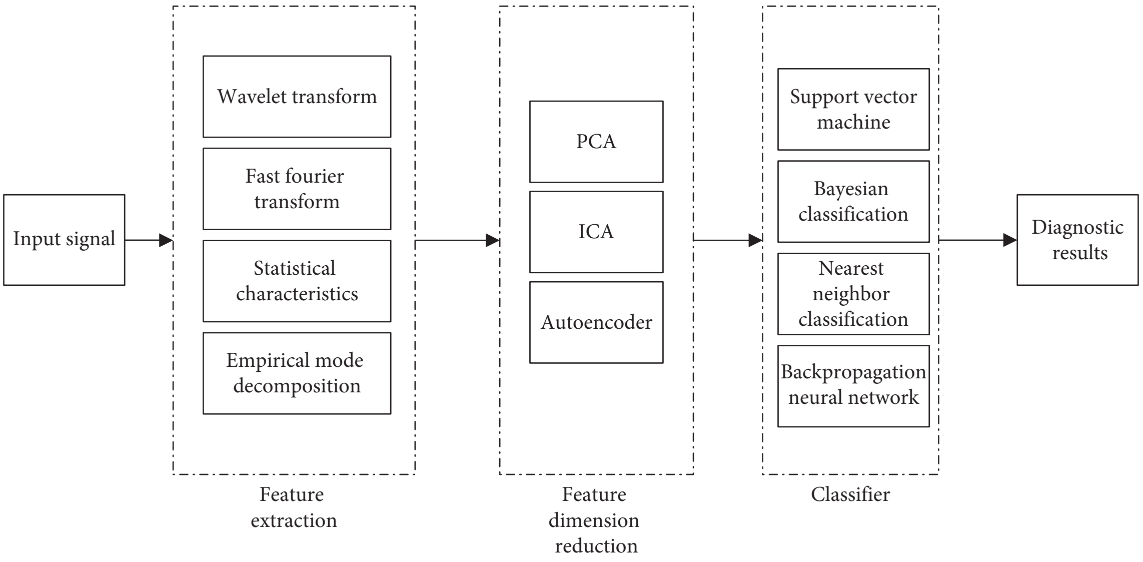


Figure 1: Fault diagnosis algorithm flow.

An increasing number of researchers are using neural networks for automatic extraction of fault features and feature dimensionality reduction and softmax for fault classification. Softmax is a generalization of a logistic classifier that mainly solves multiclassification problems. Assuming that the input sample in training data are x and the corresponding label is y, the probability of determining the sample as a class j is $p(y = j \,|\, x)$. The output of a K-class classifier will be a K-dimensional vector $h_\theta(x^{(i)})$. The elements of a vector sum to 1, and the category with the largest median of its elements is the prediction class, as shown in

$$h_\theta\left(x^{(i)}\right) = \begin{bmatrix} p\left(y^{(i)}=1\,|\,x^{(i)};\theta\right) \\ p\left(y^{(i)}=2\,|\,x^{(i)};\theta\right) \\ \vdots \\ p\left(y^{(i)}=k\,|\,x^{(i)};\theta\right) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}, \tag{1}$$

where $\theta_1; \theta_2; \ldots; \theta_k \in \mathfrak{R}^{n+1}$ is the model parameter and $1/\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}$ is the normalization function. The probability distribution is normalized so that the sum of all probabilities is 1.

Deep learning technology is developing rapidly, especially in the fields of image classification, target recognition, scene semantic analysis, and natural language processing. The deep neural network is demonstrably superior at processing complex data and predicting complex systems. Many experts and scholars in the field of mechanical failure have achieved good results when applying deep learning techniques to mechanical fault diagnosis.

Shao et al. [2], Wang et al. [3], and Chen et al. [4] used deep belief networks (DBNs) to diagnose the faults of rolling bearings and gearboxes, and the robustness and accuracy of DBNs compared to some mainstream fault diagnosis methods was verified. Using DBN, Li et al. [5] studied information extraction and fusion under high background noise and achieved superior results compared to traditional methods. Convolutional neural network (CNN) has been applied to fault diagnosis to reduce the number of model parameters and improve calculation speed. CNN can be considered a neural network model for image processing. In the field of fault diagnosis, CNN can extract features and can be used to predict the classification of artificial features [6, 7]. Lu et al. [8] processed bearing running-health data based on a shallow CNN and extracted characteristic parameters and classifications of fault states. Zhang et al. [9] constructed a multilayer one-dimensional CNN and used the time-domain signals of bearing data to carry out fault diagnosis research, with good results. Wang et al. [10] and others used short-time Fourier transforms to convert collected motor-vibration signals to spectra and constructed a two-dimensional CNN for fault diagnosis, which achieved high diagnostic accuracy. Verstraete et al. [11] transformed the time domain signals of rolling bearings to time-frequency spectrograms by short-time Fourier transform, wavelet packet transform, and Hilbert–Huang transform, trained them by CNN, and studied the performance of the network by changing the size of the input time-frequency spectrograms and the denoising method. A one-dimensional signal can be transformed to a two-dimensional time-frequency diagram by time-frequency conversion, and then fault diagnosis by CNN can achieve good results. Zhang et al. [12] took the time-frequency spectrum of a rolling bearing vibration signal after Fourier transform as input. Using deep fully convolutional neural network (DFCNN), the vibration signal data of the rolling bearing rolling 2-3 turns is modeled by a large number of convolution layers. All effects reached 100%.

The above research shows that in-depth learning has strong adaptive feature extraction and classification ability in the face of large mechanical data tasks. These studies have played a good role in the diagnosis of single target faults. However, in practice, gearboxes often have many kinds of faults simultaneously, and there are hundreds of combinations of complex faults. To solve this problem, a deep measurement learning model based on triplet loss is proposed in this paper. The multifault signals of gearbox bearings and gears are processed, and a variety of complex faults are simulated by using different bearings, gears, loads, and rotational speeds. Triplet loss is used as a loss function to optimize the model, thus efficiently completing the task of classifying complex faults [13–15].

Each triple [16, 17] is constructed by randomly selecting a sample from the training dataset as an anchor (x^a) and then randomly selecting a sample of the same type as the anchor called positive (x^p) and different classes of samples called negative (x^n). The anchor, positive, and negative constitute a complete triple. A neural network is trained for each sample in the triple, and the feature expressions of the three samples are denoted as $f(x_i^a), f(x_i^p), f(x_i^n)$.

The purpose of triplet loss is to make the distance between the characteristic expressions of x^a and x^p as small as possible, while making the distance between the characteristic expressions of x^a and x^n as large as possible. There is a minimum interval α between the distances (α is a hyperparameter, which can be set manually). As shown in Figure 2, the triplet learns to calculate the triplet loss multiple times to reduce the distance between similar samples and increase the distance between heterogeneous samples. In Euclidean space, a closer distance between two-fault data indicates greater similarity. The formula is

$$f(x_i^a) - f\left(x_i^p\right)_2^2 + \alpha < f(x_i^a) - f(x_i^n)_2^2, \tag{2}$$

where the subscript 2 represents the L2 paradigm and normalizes the data. The corresponding objective function is

$$\Sigma_i^n\left[f(x_i^a) - f\left(x_i^p\right)_2^2 - f(x_i^a) - f(x_i^n)_2^2 + \alpha \right]_+. \tag{3}$$

In the previous equation, the subscript + indicates that when the value in brackets is greater than zero, the loss is the value, and when it is less than zero, the loss is zero. It can be seen from the objective function that when the distance between the characteristic expressions of x^a and x^p is greater than that between the expressions of x^a and x^n minus α. If the value in brackets is greater than zero, the loss will occur. And, conversely, the loss will be zero. When the loss is not zero, all network parameters are adjusted by a backpropagation algorithm to optimize the features.

The choice of triplets is essentially a resampling process. There are many eligible triplets in the entire dataset. Assume a total of B fault data points of P types and K data points of each type. Then $B = P \times K$; that is, there are B original sample points, $K - 1$ similar sample points, and $B - K$ heterogeneous sample points. Therefore, the number of qualified triplets is $B \times (K - 1) \times (B - K)$. Some triplets satisfy the optimization goal; that is, the distance between x^a and x^p is much smaller than between x^a and x^n. Calculating these triplets is not helpful for optimizing the target. On the contrary, it will reduce the training efficiency. Schroff et al. [18] mentioned that we should choose the combination which violates the optimization goal most seriously, where the distance between x^a and x^p is much larger than that between x^a and x^n. Whether the optimization goal is violated is found by calculating the Euclidean distance of the new features of embedding between the fault data, but the embedding updates the transformation each time; that is, the triplets that violate the optimization target may be different each time. If the triplet is reselected for each update, the training efficiency of the algorithm will be greatly reduced. There are currently two solutions to this issue.

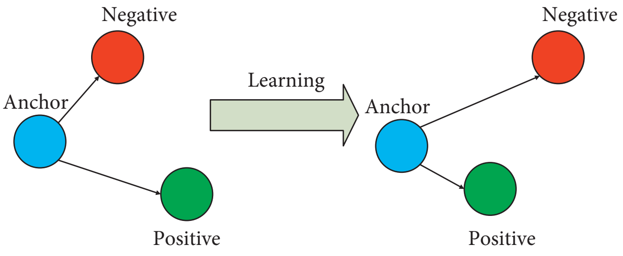


Figure 2: Triplet loss optimization goal.

(1) For each n iterations, traverse the triplets to calculate the triplet loss on the latest training result until the network converges or iterates to a stop, instead of updating the triplets after each iteration.

(2) Update the triplet loss online. A small number of triplets are selected to form a minibatch, in which all positive fault pairs are selected according to embedding at that time. Then, a negative fault that satisfies the condition that the distance between x^a and x^n is less than that between x^a and x^p is selected in the negative fault group. After calculating the triplet loss based on the above triplets, the embedding is updated and repeated until the network converges or the iteration stops.

The composite fault of the gear and bearing faults in the gearbox is taken as the research object of this paper, and the deep metric learning model is established. The model uses triplet loss as a loss function to construct a network. It maps features to Euclidean space and calculates the feature distance of similar samples and heterogeneous samples on Euclidean space. The closer the distance, the higher the similarity. By continually optimizing the triplet loss, the neural network continues to learn new features and bring the distances of similar samples ever closer, while the distance of heterogeneous samples increases.

Figure 3 shows the deep metric learning network model designed in this paper. The model consists of four layers (input layer, deep network, triple layer, and loss function calculation layer).

The task of the deep network layer is to extract the characteristics of the composite fault signal. There are many network structures to choose from, such as the convolutional neural network (CNN), long short-term memory neural network (LSTM), and fully connected neural networks.

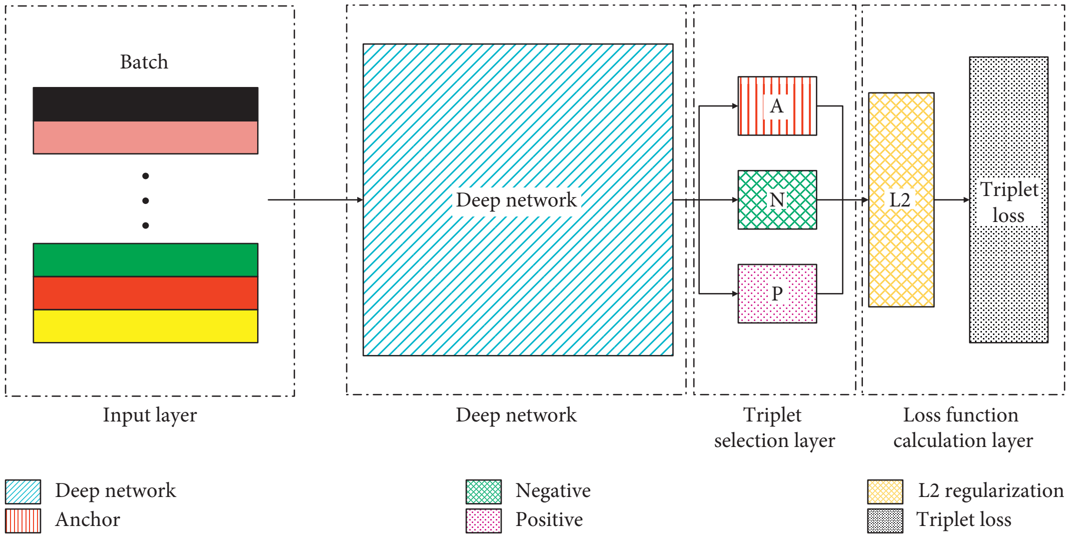


FIGURE 3: Structure of the deep metric learning model.

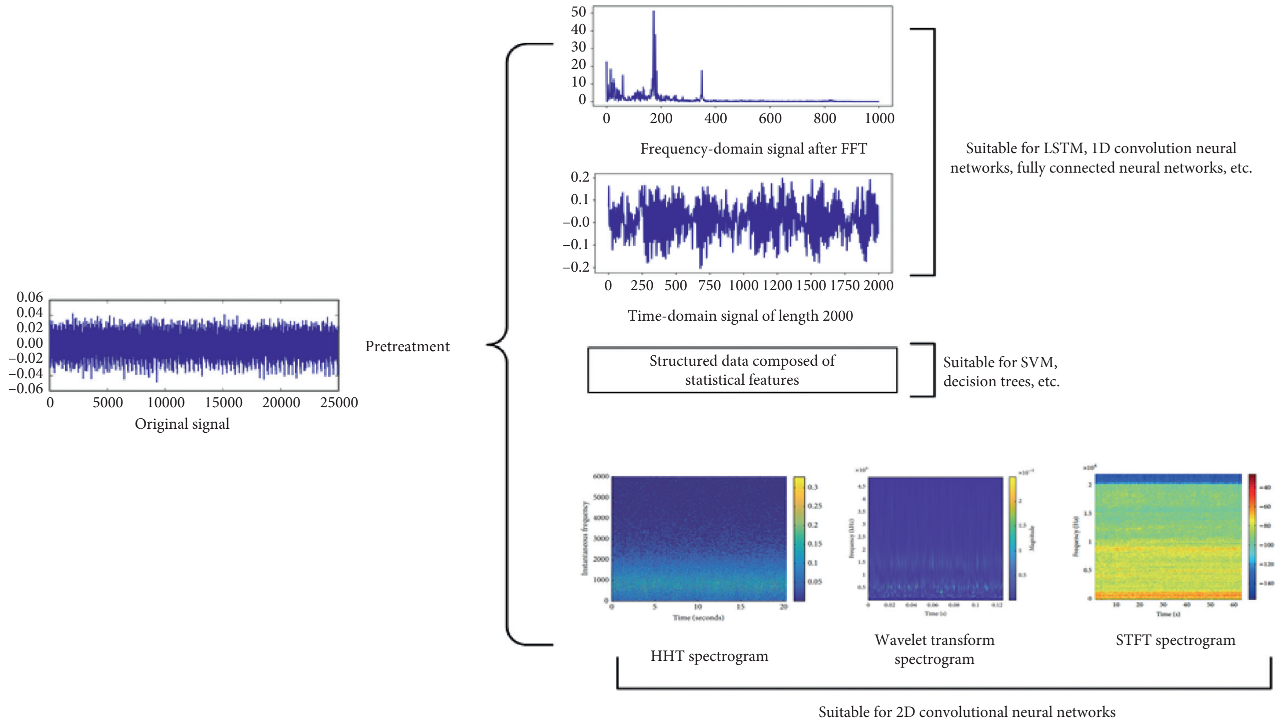


FIGURE 4: Pretreatment method.

The input forms corresponding to various networks are shown in Figure 4.

The literature [10, 19] shows that the effect of using the time domain signal directly as input for network training is not good. At the same time, the loss of the network cannot converge when the time domain signal is used as input, and the accuracy is only 30%. When diagnosing single faults of a gearbox [20], network training using the frequency domain signal as input data after fast Fourier transform achieved good results.

The composite fault signal is a nonstationary signal whose frequency varies with time. It is more complex than a single fault signal. It is difficult to accurately diagnose the composite fault information in a gearbox using only the frequency domain signal, which only extracts the components of each frequency in the signal, and losses the time information of each frequency. Therefore, two signals with very different time domains may be the same as the spectrum.

Consider a nonstationary signal as a superposition of a series of short-term signals. In this paper, STFT [21] is used to divide the signal into several time intervals. On the basis of a traditional Fourier transform, the frequency spectrum is calculated by a sliding time window, and the frequency in a certain time interval is determined. The time-frequency description of the signal is carried out so that the time information will not be lost. Assuming a nonstationary signal S_t, the short-time Fourier transform of S_t is defined as

$$\mathrm{STFT}_s(t, w) = \int_{-\infty}^{+\infty} S(\tau) h(\tau - t) e^{-iw\tau} \mathrm{d}\tau, \tag{4}$$

where t is the time translation parameter and $h(t)$ represents a window function centered on t, truncates the signal through the window function, and divides the signal into multiple segments.

The intercepted signal can be expressed as

$$S_t = S(\tau) h(\tau - t), \tag{5}$$

where S_t is a signal corresponding to the original signal for a fixed time t, and $S(T)$ is a signal whose execution time corresponds to T. A Fourier transform of S_t is used to obtain the spectrum of S_t:

$$S_t(w) = \frac{1}{\sqrt{2\pi}} \int \mathrm{e}^{-jw\tau} S_t(\tau) \mathrm{d}\tau = \frac{1}{\sqrt{2\pi}} \int \mathrm{e}^{-jw\tau} S_t(\tau) h(\tau - t) \mathrm{d}\tau. \tag{6}$$

By changing the size of the translation parameter t, the center position of the window function can be changed to obtain Fourier transforms at different times.

A different spectrum is obtained at each time interval, and the total of these spectra constitutes a time-frequency distribution, that is, a spectrogram.

After the short-time Fourier transform of the signal, the spectral energy relation of time t is

$$P_{\mathrm{sp}}(t, w) = \left|S_t(w)\right|^2 = \left|\frac{1}{\sqrt{2\pi}} \int \mathrm{e}^{-jw\tau} S_t(\tau) h(\tau - t) \mathrm{d}\tau\right|^2. \tag{7}$$

As shown in Figure 5, the composite fault signal is converted into a time-frequency diagram through STFT and finally compressed to generate an 80 × 80 image for input to the network.

The structure of a convolutional network layer directly affects the effect of the network model, so the selection of network structure parameters is particularly important.

Figure 6 shows the structure of the convolutional network layer.

Table 1 shows the structural parameters adopted by the convolutional neural network layer. The network uses ReLU as the activation function and a uniform distribution when initializing network parameters. The range is [−0.1, 0.1]. The network uses the Adam optimizer with the learning rate set to 0.06. The dropout is set to 0.5 to avoid overfitting on the network. This structure ensures that the network can learn as many features as possible, and it prevents overfitting.

The convolutional neural network layer is followed by a triplet selection layer that shares the 32-dimensional features of the convolutional neural network output and generates a triplet for optimization. This article takes the online update of triplet loss mentioned in Section 3 to solve the triplet selection problem. The last layer is the loss function calculation layer, which normalizes the characteristics of the output through the L2 paradigm and finally calculates the triplet loss.

Triplet loss is used as the loss function of the network (hyperparameter margin = 1). With the minimized triplet loss as the optimization goal of the network, the back-propagation (BP) algorithm is used to continuously update the weight of the neural network to train the optimal features.

In the new trained feature space, the distance between the data of different fault types is great, and the distance between data of the same fault type is small [5, 22, 23].

3. Operation Process on Edge Equipment

Figure 7 shows a flowchart of training and fault diagnosis of the deep measurement learning model based on triplet loss.

The steps are as follows.

The first step is sample collection. Through the short-time Fourier transform, the comprehensive fault data of gearbox is converted into time-frequency diagram.

The next is network training. The frequency domain signal is input to the network, and the deep network extracts a feature of each point of fault data. With the Euclidean distance between these features and the label of the fault data, the model can select the triplet according to the second scheme in Section 3 to calculate the triplet loss function. The network weights are updated by backpropagation and the above steps are repeated until the network converges or the iteration ends, saving the model parameters.

The third step is diagnosis. At the end of model training, forward propagation is used to obtain a feature of each fault category, which is the template needed to diagnose the unknown fault data. After deep network processing, the unknown fault data can also get a feature. The Euclidean distance between the feature of unknown data and the feature of the template can be calculated, and the minimum value of the Euclidean distance can be selected. The diagnosis result can be obtained by comparing the minimum value with the preset threshold.

The fourth step is to obtain the diagnosis results. The threshold is set because the unknown fault data probably indicate a completely new type of fault. Its characteristics are far from the Euclidean distance of each feature in the template, but the model selects a recent fault type as the output. If the minimum value is greater than the threshold, then the feature vector of the fault is stored in the template library, the label is recorded as unknown fault 1, and the diagnosis result outputs "unknown fault 1." When the unknown data of the fault type are encountered again, the deep metric learning model can accurately diagnose it. If the minimum value is less than the threshold, then the diagnosis result is output as the fault category of the template closest to the fault data (the smaller the Euclidean distance between fault data features is, the more similar they are).

4. Experiment and Analysis

4.1. Data Preprocessing Based on Edge Devices. Training deep learning networks requires a large amount of data support,

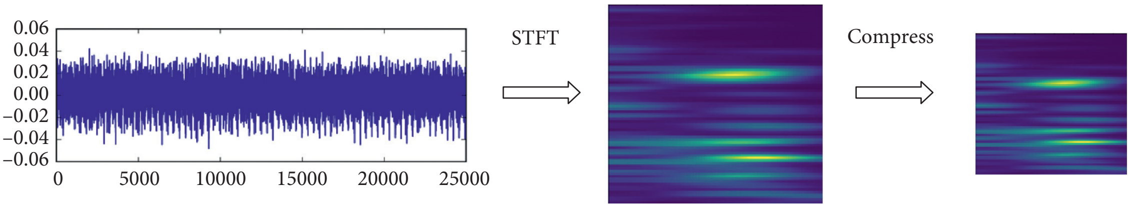


Figure 5: Pretreatment results.

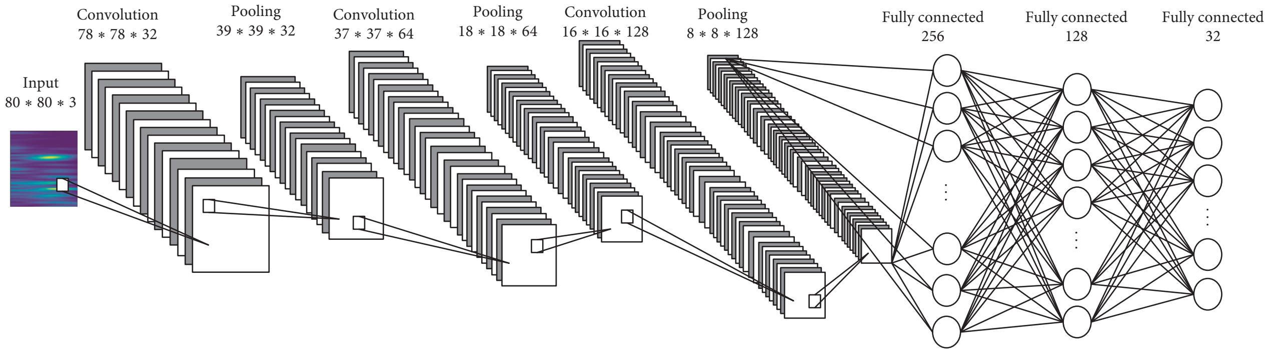


Figure 6: Structure of convolutional neural network.

Table 1: Network structure parameters.

Layers	Output size	Parameter
Conv2d	(None, 78, 78, 32)	320
Activation (ReLU)	(None, 78, 78, 32)	0
max_pooling2d	(None, 39, 39, 32)	0
Conv2d_1	(None, 37, 37, 64)	18, 496
Activation_1 (ReLU)	(None, 37, 37, 64)	0
max_pooling2d_1	(None, 18, 18, 64)	0
Conv2d_2	(None, 16, 16, 128)	73, 856
Activation_2 (ReLU)	(None, 16, 16, 128)	0
max_pooling2d_2	(None, 8, 8, 128)	0
Flatten	(None, 8192)	0
Dense	(None, 256)	2, 097, 408
Activation_3 (ReLU)	(None, 256)	0
Dropout	(None, 256) dropout = 0.5	0
Dense_1	(None, 32)	8, 224

and the quality of training data directly influences the output of the model. This paper used the power transmission fault diagnosis test bench (DDS) produced by Spectra Quest as the research object (as shown in Figure 8). When installing the acceleration sensor (SQI608A11-3F), this study refers to the acquisition method of the bearing data of Case Western Reserve University (http://csegroups.case.edu/bearingdatacenter/home). The acceleration sensor was mounted on the left and right sides of the fixed shaft of the gearbox by bolts (as shown in Figure 8 for sensors 1 and 2). The sampling frequency was 20 kHz, and the sampling time was 20 s.

The research object of this paper is gearbox compound fault. At present, there is no open dataset of compound fault. Therefore, we refer to the collection process of bearing data of Case Western Reserve University and carry out data collection on the Drivetrain Dynamics Simulator (DDS) by ourselves. If there is an appropriate open dataset, we will conduct further research.

By replacing gears (e.g., those with missing teeth, broken teeth, eccentricity, excessive wear, and cracks) and bearings (inner ring fault, outer ring fault, rolling element fault, and compound fault) in the gearbox, 30 kinds of faults that may occur in the gearbox were simulated, as shown in Figure 9. To simulate a more realistic production environment, artificial noise pollution was carried out by tapping the gearbox or table with metal at random time, and the pollution signal accounted for about 5% of the total signal.

The picture of specific fault location and damage degree is shown in Figure 10.

At the same time, to increase the diversity of the sample, the speed was changed by controlling the driving motor of the front end when collecting data; the load was changed by controlling the load regulator, so as to simulate the type of working conditions that could occur in actual production. Each fault sample was collected at four motor speeds (1700, 1800, 3400, and 3800 rpm) and four loads (A, B, C, and D; see Table 2 for the load voltage and current of each load).

The time domain signals of the left and right channels were collected under each working condition to obtain 960 vibration signal files (30 multifault combination types × 4 speeds × 4 loads × 2 channels). Each signal file contained 409,600 signal points.

The vibration signal file was divided randomly, and the 409,600 points in each signal file were evenly divided into 200 segments of 2,048 points. In order to fully study the recognition ability of the model, nine different data division methods are developed in this paper; the division is shown in Table 3.

In order to compare the diagnostic effect of convolution neural network + softmax classifier and the depth measurement learning model based on triplet loss, two kinds of

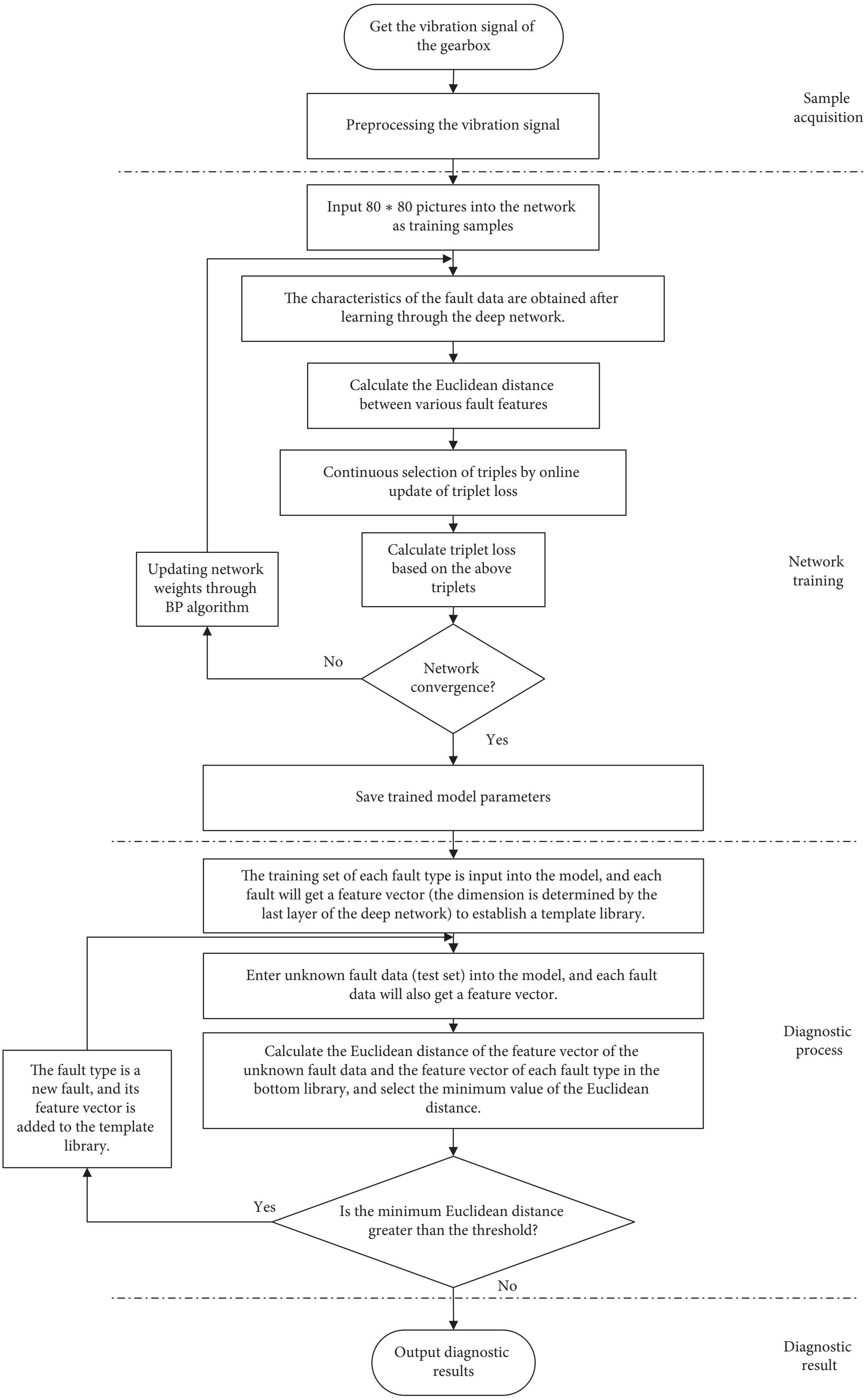

Get the vibration signal of the gearbox
Preprocessing the vibration signal
Sample acquisition
Input 80 * 80 pictures into the network as training samples
The characteristics of the fault data are obtained after learning through the deep network.
Calculate the Euclidean distance between various fault features
Continuous selection of triples by online update of triplet loss
Calculate triplet loss based on the above triplets
Updating network weights through BP algorithm
Network training
No
Network convergence?
Yes
Save trained model parameters
The training set of each fault type is input into the model, and each fault will get a feature vector (the dimension is determined by the last layer of the deep network) to establish a template library.
Enter unknown fault data (test set) into the model, and each fault data will also get a feature vector.
Diagnostic process
The fault type is a new fault, and its feature vector is added to the template library.
Calculate the Euclidean distance of the feature vector of the unknown fault data and the feature vector of each fault type in the bottom library, and select the minimum value of the Euclidean distance.
Yes
Is the minimum Euclidean distance greater than the threshold?
No
Output diagnostic results
Diagnostic result

Figure 7: Model training and diagnostic process.

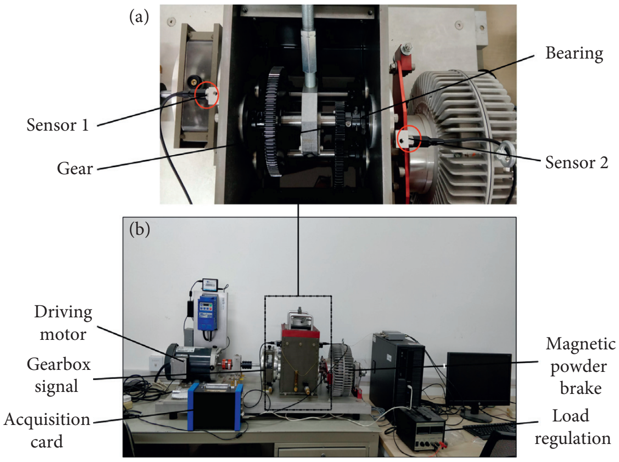


Figure 8: Power transmission fault diagnosis test bench.

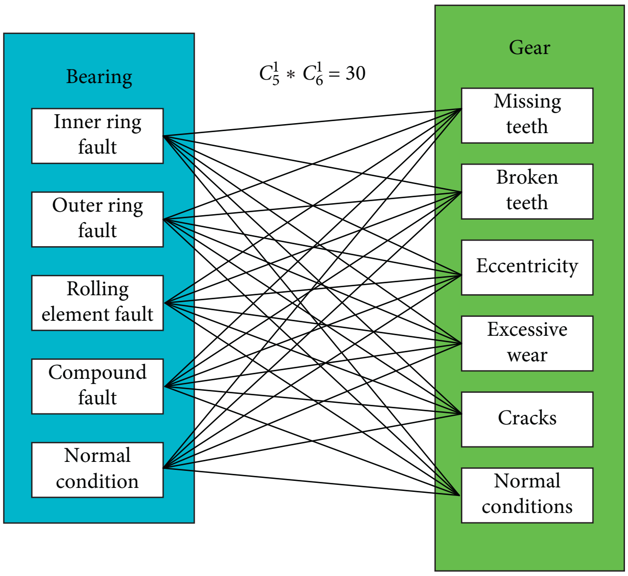


Figure 9: Combination types of bearing and gear faults.

labels are made (the structure and parameters of convolution network in the two models are identical).

(1) The fault types were divided into 30 categories (five bearing faults and six gear faults). We created labels for training and test sets separately. The label corresponding to softmax must encode the label of the data. We used one hot encoding.

(2) The fault types were divided into 30 categories (five bearing faults, six gear faults, four loads, and four speeds). We created separate labels for training and test sets. The label for entering the triplet loss was the ones matrix.

5. Results and Discussion

5.1. Experimental Verification. The data in Table 3 are input into the diagnostic model of convolution neural network + softmax classifier and the deep measurement learning model based on triplet loss, respectively, for model training (the structure and parameters of convolution network are identical in the two models).

From the experimental results in Table 4, we can see that the accuracy of the diagnostic model of convolutional neural network + softmax classifier can reach 94.66% when the percentage cut data (i.e., sufficient training data, including various rotational speeds and loads) are normalized (i.e., Table 4, experiment 10), while the accuracy of the depth measurement learning model based on triplet loss can reach 97.73%.

When using the data missing from a certain load to train the network and using this load data for network testing (experiments 2–5, 11–14), the two models can obtain higher accuracy on the test set, but the deep measurement learning model is still better [24–26].

When using the data missing from a certain speed to train the network and using this speed data for network testing (experiments 6–9, 15–18), we can see from Figure 11

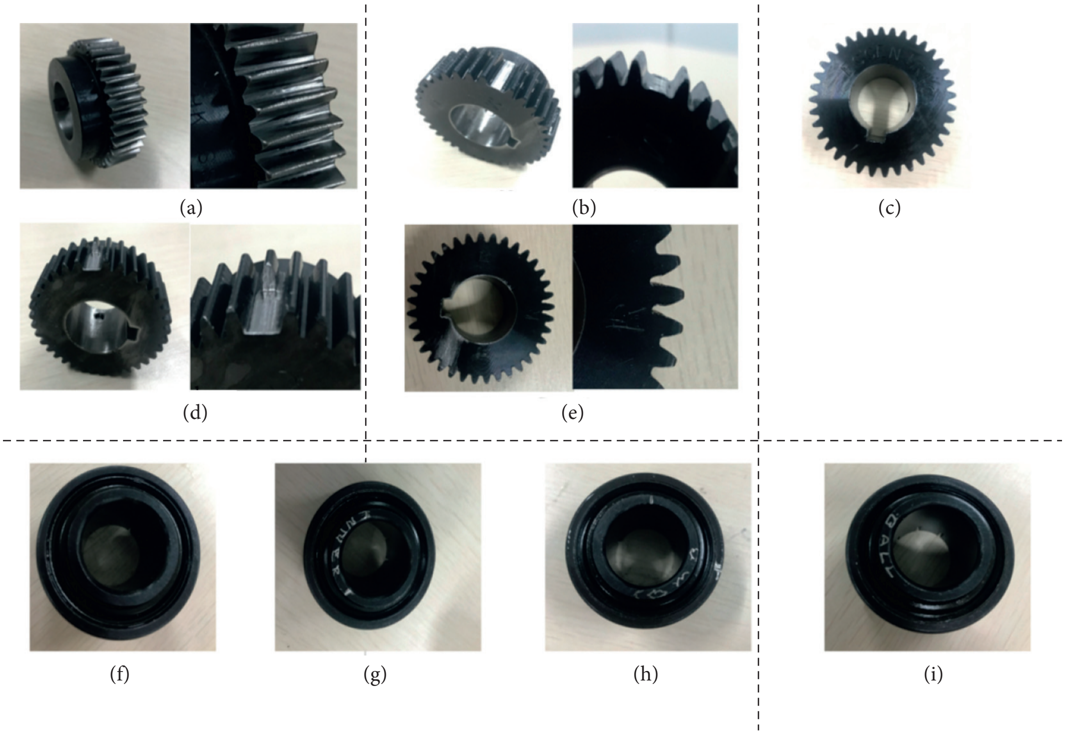


Figure 10: Bearing faulty bearings and gears. (a) Excessive wear. (b) Missing teeth. (c) Eccentricity. (d) Broken teeth. (e) Cracks. (f) Outer ring fault. (g) Inner ring fault. (h) Compound fault. (i) Rolling element fault.

Table 2: Type of load.

Load	Current (A)	Voltage (V)
A	0	0
B	0.37	4
C	0.56	6
D	0.75	8

Table 3: Segmentation of experimental data.

Dataset	Segmentation method	Training set		Test set		Total number of samples
		Types of signals	Number of samples	Types of signals	Number of samples	
A	Segmentation by percentage	Random 75%	144,000	Remaining 25%	48,000	192,000
B	Segmentation by load	B, C, D	144,000	A	48,000	192,000
C	Segmentation by load	A, C, D	144,000	B	48,000	192,000
D	Segmentation by load	A, B, D	144,000	C	48,000	192,000
E	Segmentation by load	A, B, C	144,000	D	48,000	192,000
F	Segmentation by speed	1800, 3400, 3800	144,000	1700	48,000	192,000
G	Segmentation by speed	1700, 3400, 3800	144,000	1800	48,000	192,000
H	Segmentation by speed	1700, 1800, 3800	144,000	3400	48,000	192,000
I	Segmentation by speed	1700, 1800, 3400	144,000	3800	48,000	192,000

Table 4: List of the highest test set accuracy achieved by each experiment.

Experiment	Data	Normalization	Accuracy of test set	
			Using softmax	Using triplet loss
1	A	No	0.93942	0.9611
2	B	No	0.89214	0.9437
3	C	No	0.90307	0.9489
4	D	No	0.91742	0.9509
5	E	No	0.92573	0.9593
6	F	No	0.68717	0.8334
7	G	No	0.67384	0.8376
8	H	No	0.48658	0.8197
9	I	No	0.38966	0.8168
10	A	Yes	***0.94668***	0.9773
11	B	Yes	0.91268	0.9615
12	C	Yes	0.92335	0.9703
13	D	Yes	0.93611	0.9756
14	E	Yes	0.93282	0.9632
15	F	Yes	***0.68371***	0.9124
16	G	Yes	***0.64178***	0.9133
17	H	Yes	***0.50438***	0.9098
18	I	Yes	***0.49335***	0.9045

Underlined italic values indicate extreme cases in the experimental results, and they are explained in detail below.

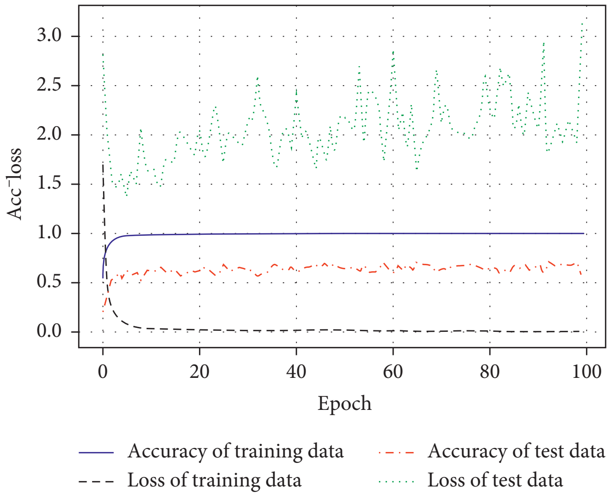


Figure 11: Accuracy of traditional network in experiment 10.

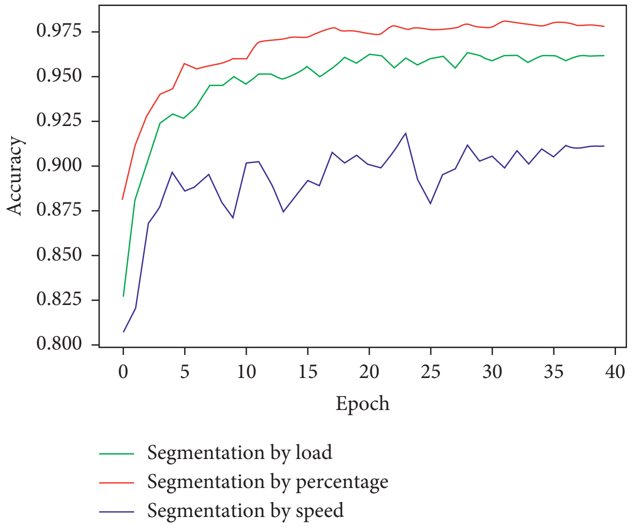


Figure 12: Accuracy of deep metric learning model with different data.

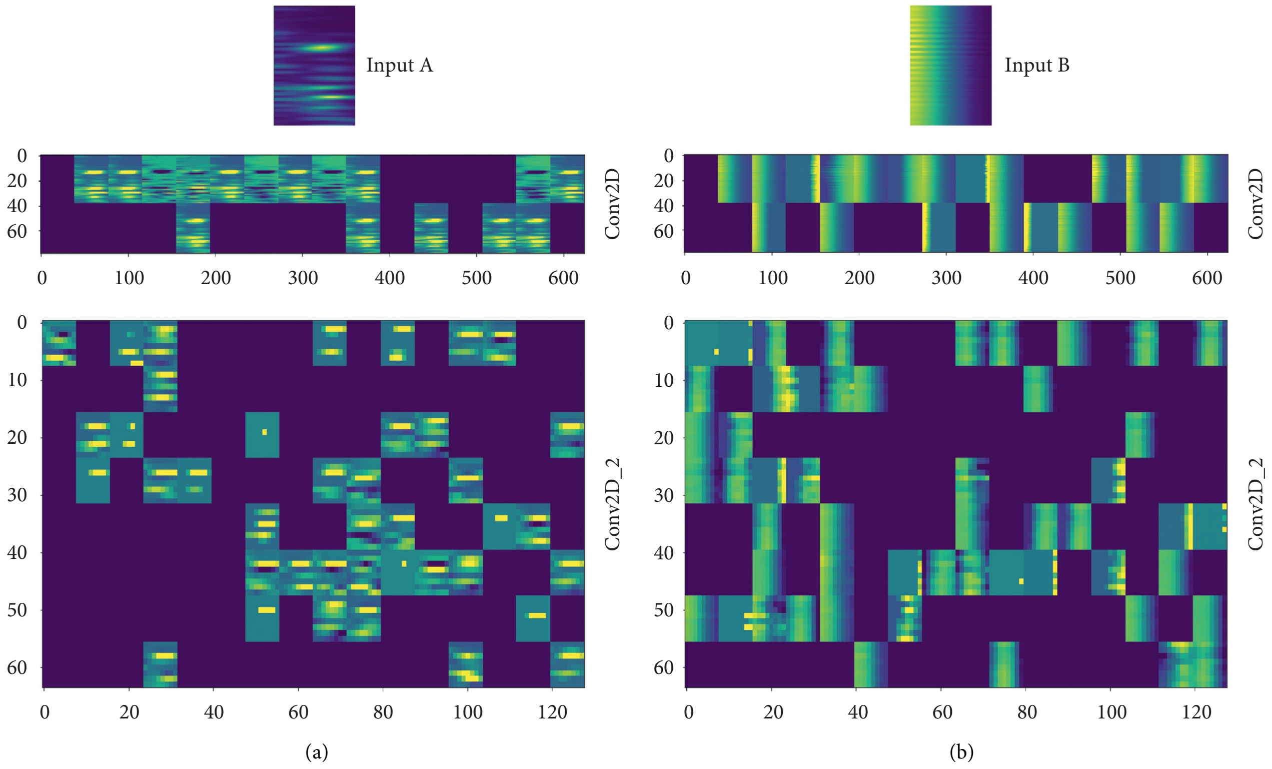


Figure 13: Feature visualization of faults A and B. (a) Input A. (b) Input B.

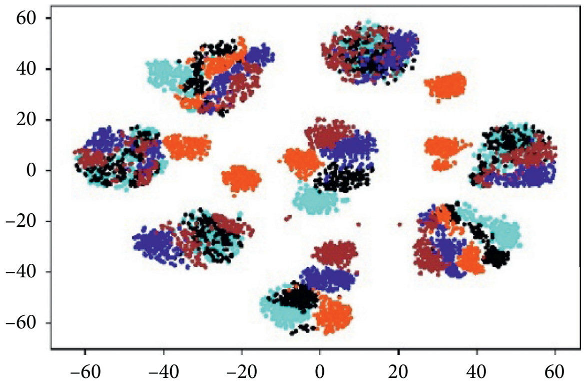


Figure 14: T-SNE visualization before processing.

that the diagnostic model of convolutional network + softmax classifier produces serious overfitting phenomenon, and the accuracy of the test set is very low.

But the deep metric learning model has strong generalization ability. As shown in Figure 12, the model can still obtain a high diagnostic accuracy even when some working condition data may be missing in practical application. Even in the experiment of missing motor speed data (experiments 15–18), it can achieve more than 90% accuracy.

The basic reasons why the method proposed in this paper can achieve higher diagnostic accuracy are as follows:

(1) Vibration signals are transformed into time-frequency diagrams by STFT, and the features of time-frequency diagrams are extracted by convolutional neural network; thus the frequency and time information of fault signals are effectively utilized. In order to verify the capability of the proposed method in feature extraction more intuitively, two different kinds of fault signals (A and B) are randomly selected and input into the network model with the highest accuracy obtained in Table 4, experiment 10. The output features of convolution layer Conv2d and convolution layer Conv2d_2 in Figure 6 are visualized [27]. The visualization results of the convolution kernel of signals A and B are shown in Figure 13.

(2) Using triplet loss to measure the distance between different kinds of faults makes the distance between

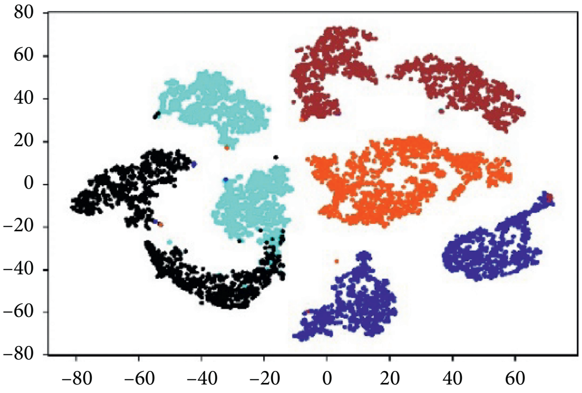


Figure 15: T-SNE visualization after processing.

similar fault features very close and that between different fault features very far, which makes the diagnosis more accurate and easier. However, the traditional convolutional neural network + softmax classifier model does not measure the distance between fault features. In order to prove that the model can make the fault data feature meet the distance between similar samples more and more close, at the same time, the distance between different samples is farther and farther [28]. Five fault types (A, B, C, D, and E) are selected randomly in this paper. 1600 data points ($5 * 1600 = 8000$ data) are selected randomly for each fault type. Figure 14 shows the original distribution of 8000 data points visualized by T-SNE.

Figure 15 shows the distribution of 8000 data points in t-SNE visualization after processing by deep measurement learning model.

6. Conclusions

In this paper, for the first time, the depth metric learning model is used to diagnose the faults of inner bearings and gears in gearbox at the same time, and the collected data are segmented in different ways to simulate the situation of missing some working condition data in practical application, so as to verify the performance of the network model. At the same time, the model of convolution neural network + softmax classifier is constructed to compare and classify. The conclusions are as follows:

(1) When there is no missing data type, the depth metric learning model can extract features adaptively when dealing with complex faults of gearbox. The diagnostic accuracy of complex faults can reach 97.73%, which is higher than that of using convolution network + softmax classifier model.

(2) When using the missing load data to train the network and use the missing load data to test the network, the accuracy of the convolution network + softmax classifier model is still 97 + 0.6%, while the accuracy of the convolution network + softmax classifier model is only 92 + 1%.

(3) When using the data training network with missing speed and using the missing speed data for network testing, the convolution network + softmax classifier model produces serious overfitting phenomenon, and the accuracy of the test set is very low, only about 60%. However, the depth measurement learning model has not been fitted, and the accuracy of the test set is still higher than 90%.

(4) The multifault dataset of gearbox collected in this paper has certain research value and can be used to evaluate the model for this kind of problem.

Data Availability

This article refers to the collection method of CWRU bearing data, sensor placement, etc., using Spectra Quest's powertrain fault diagnosis comprehensive test bench (Drivetrain Dynamics Simulator, DDS) as the test object to collect gearbox compound fault data. The storage address of the compound fault data in the gearbox is https://pan.baidu.com/s/1zBJLV-O5v6nS9rfjI6a5Kg and the extraction code is 286w. These data include vibration signal data without any processing.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors thank LetPub (http://www.letpub.com) for its linguistic assistance during the preparation of this manuscript. The authors acknowledge the financial support of the National Science Foundation of China (Grant nos. 51505234, 51575283, and 51405241).

References

[1] S. Li, S. Zhao, Y. Yuan, Q. Sun, and K. Zhang, "Dynamic security risk evaluation via hybrid bayesian risk graph in cyber-physical social systems," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 4, pp. 1133–1141, 2018.

[2] H.-d. Shao, H.-k. Jiang, X. Zhang et al., "Rolling bearing fault diagnosis using an optimization deep belief network," *Measurement Science & Technology*, vol. 26, no. 11, 2015.

[3] X.-q. Wang, L. Yan-feng, T. Rui et al., "Bearing fault diagnosis method based on Hilbert envelope spectrum and deep belief network," *Journal of Vibroengineering*, vol. 17, no. 3, pp. 1295–1308, 2015.

[4] Z.-q. Chen, C. Li, and R. V. Sánchez, "Multi-layer neural network with deep belief network for gearbox fault diagnosis," *Journal of Vibroengineering*, vol. 17, no. 5, pp. 2379–2392, 2015.

[5] C. Li, R.-V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. Vásquez, "Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis," *Neurocomputing*, vol. 168, no. C, pp. 119–127, 2015.

[6] M. Bhadane and K. I. Ramachandran, "Bearing fault identification and classification with convolutional neural network," in *Proceedings of the International Conference on Circuit ,Power*

and Computing Technologies (ICCPCT), pp. 1–5, IEEE, Kollam, India, April 2017.

[7] H. Jeong, S. Park, S. Woo, and S. Lee, "Rotating machinery diagnostics using deep learning on orbit plot images," *Procedia Manufacturing*, vol. 5, pp. 1107–1118, 2016.

[8] C. Lu, Z. Wang, and B. Zhou, "Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification," *Advanced Engineering Informatics*, vol. 32, pp. 139–151, 2017.

[9] W. Zhang, G.-l. Peng, and C.-h. Li, *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, pp. 77–84, Springer, Taiwan, 2016.

[10] L.-H. Wang, X.-P. Zhao, J.-X. Wu, Y.-Y. Xie, and Y.-H. Zhang, "Motor fault diagnosis based on short-time fourier transform and convolutional neural network," *Chinese Journal of Mechanical Engineering*, vol. 30, no. 6, pp. 1357–1368, 2017.

[11] D. Verstraete, A. ferrada, E. L. Droguett, V. Meruane, and M. Modarres, "Deep learning enabled fault diagnosis using time-frequency image analysis of rolling element bearings," *Shock and Vibration*, vol. 2017, Article ID 5067651, 17 pages, 2017.

[12] W. Zhang, F. Zhang, W. Chen et al., "Fault state recognition of rolling bearing based fully convolutional network," *Computing in Science & Engineering*, vol. 21, no. 5, pp. 55–63, 2018.

[13] X. Xu, B. Shen, X. Yin et al., "Edge server quantification and placement for offloading social media services in industrial cognitive IoV," *IEEE Transactions on Industrial Informatics*, 2020.

[14] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and B. Zakirul Alam, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[15] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, and L. Qi, "Trust-oriented IoT service placement for smart cities in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4084–4091, 2020.

[16] R. Salakhutdinov and G. E. Hinton, "Replicated softmax: an undirected topic model," in *Proceedings of the International Conference on Neural Information Processing Systems*, Curran Associates Inc., Vancouver, Canada, pp. 1607–1614, 2010.

[17] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," *Similarity-Based Pattern Recognition*, Springer International Publishing, Cham, Switzerland, 2015.

[18] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: a unified embedding for face recognition and clustering," pp. 815–823, 2015.

[19] L. I. U. Hong-mei, L. I. Lian-feng, and M. A. Jian, "Rolling bearing fault diagnosis based on STFT-deep learning and sound signals," *Shock and Vibration*, vol. 2016, Article ID 6127479, 12 pages, 2016.

[20] X. Zhao, J. Wu, Y. Zhang, Y. Shi, and L. Wang, "Fault diagnosis of motor in frequency domain signal by stacked denoising auto-encoder," *Computers, Materials & Continua*, vol. 57, no. 2, pp. 223–242, 2018.

[21] Y. Wang, J. Wang, and H. Huang, "Fault diagnosis of variable speed planetary gearbox based on nonlinear short time fourier transform order tracking," *China Mechanical Engineering*, vol. 29, no. 14, pp. 54–61, 2018.

[22] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2010.

[23] P. Tamilselvan and P. Wang, "Failure diagnosis using deep belief learning based health state classification," *Reliability Engineering & System Safety*, vol. 115, no. 7, pp. 124–135, 2013.

[24] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2020.

[25] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Transactions on Network Science and Engineering*, 2020.

[26] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, 2019.

[27] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou, and Q. Ni, "Data-driven web APIs recommendation for building web applications," *IEEE Transactions on Big Data*, 2020.

[28] C. P. Mbo'O and K. Hameyer, "Bearing damage diagnosis by means of the linear discriminant analysis of stator current feature," in *Proceedings of the 2015 IEEE 10th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED), IEEE, Guarda, Portugal*, September 2015.

Hindawi
Security and Communication Networks
Volume 2020, Article ID 8833416, 14 pages
https://doi.org/10.1155/2020/8833416

Review Article

A Survey on Services Provision and Distribution of Official and Commercial Intellectual Property Platforms

Yang Wang[1], **Haijin Gui**[2] **and Lei Ma**[3]

[1] *School of Intellectual Property, Nanjing University of Science and Technology, Nanjing 210094, China*
[2] *Wuxi Vocational Institute of Commerce, Wuxi 214153, China*
[3] *School of Public Affairs, Nanjing University of Science and Technology, Nanjing 210094, China*

Correspondence should be addressed to Lei Ma; maryma208@sina.com

Received 1 May 2020; Revised 1 June 2020; Accepted 18 June 2020; Published 1 August 2020

Academic Editor: Xiaolong Xu

Next generation of Information Technologies (IT), such as edging/cloud computing, cybersecurity, and artificial intelligence, has been in a rapid development and therefore concerned wide areas. Management of Intellectual Property Rights (IPR) plays an increasingly important role on knowledge design and engineering, innovation and patent management, intangible assets audition, R&D, and so forth; however, it also meets the challenges from proper platform and service provision, especially when large-scale mobile and distributed requirements become popular. In this paper, Intellectual Property Platforms and corresponding commercial tools have been collected, investigated, and reviewed, involving official platforms in China, USA, EU, and another 6 countries, as well as 12 intellectual property analysis tools commonly used online. Detailed comparison and discussion have been undertaken in order to find potential challenges and opportunities for improved service provision, for example, searching privacy preservation, cloud/edge-based service offloading and efficient distribution, and ontology-based intelligent IPR engineering, which can possibly be commercialised in the near future.

1. Introduction

The powerful advantages of cloud computing technology bring great convenience to the public. With the deepening of technology demand, edge computing also plays a huge role. Edge computing is an open framework, in which data source close to the edge of the network and integrates core functions, such as grid, storage, computing, application, and provides the nearest edge intelligent service according to the main needs of the current digital industry in data optimization, fast connection, privacy security protection, and so forth [1, 2]. However, it cannot be ignored that there are still considerable opportunities and challenges in wireless communication security of edge-cloud computing [3]. In the same network, including the data information, its security and usability are threatened, thus the extension of a series of network security management technology, such as data encryption technology, antivirus technology, firewall technology. The Internet of things further extends the physical and logical boundaries of the Internet, gradually forming a three-tier service delivery architecture based on the matter-edge-cloud. With the emerging era of 5G network, edge computing can be applied to computing unloading of Internet of things [4–6].

Intellectual Property (IP) refers to invention; literary works are created by the mind, including symbols, names, and images used in business. In law, IP rights are protected by patents, copyrights, trademarks, and so forth, to ensure that people can obtain recognition and their own interests from their own inventions or creations. In order to create and innovate a thriving environment, an IP system should be built to achieve the balance between innovators and wider public interests [7, 8].

There are two types of IP rights:

(i) Industrial property is composed of industrial design, invention patent, trademark, and geographical indication.

(ii) Copyright includes literary works, film and television works, art works, and architectural design. For example, the rights of performing artists and producers of works are related to copyright.

Nowadays, industry 4.0 and industry 5.0 have been envisaged, which is likely to bring significant changes to IP and IP information [9–11]. With the rapid development of Internet innovation technology, information resources have become one of the most strategic resources in the current society [12, 13]. The key to global competition in the future is IP competition. It is the general trend to provide the privacy protection methods of blockchain in the network mobile environment, improve the society's awareness of IP protection in the network environment [14–16], and guide the whole society to raise the issue of IP protection to the global development strategy. With the increasing number of published IP rights, what searches the target information quickly, accurately, and securely from the massive data has become one of the most concerned hot issues of the public. At the same time, the retrieval and analysis of this huge knowledge source becomes a complex, detailed, highly interactive, and repetitive task, which requires a lot of professional knowledge with different retrieval strategies. For this reason, the number of analysis platforms issued by relevant domestic and foreign institutions has increased correspondingly to assist in reviewing a great quantity of files. This paper describes the commonly used Intellectual Property (IP) offices in various countries and the main patent search and analysis platforms in China and abroad and summarizes and compares their pros and cons.

The rest of the paper is divided into the following sections: Intellectual Property Platforms have been introduced in Section 2, followed by intellectual property services provision in Section 3. Detailed discussions and perspectives have been proposed in Section 4 with conclusion in Section 5.

2. Intellectual Property Platforms

2.1. Intellectual Property Platforms in China. The China National Intellectual Property Administration (CNIPA) [17] formerly was called the State Intellectual Property Office of China (SIPO), which is supervised by the newly established state administration for market regulation now. It was established in 1980 with the approval of the State Council. CNIPA mainly manages patent, utility model, trademark, and other IP related fields. The patent search and analysis system launched by CNIPA provides rich and comprehensive data resources and professional and high-quality patent search and analysis services. At present, the system has included patent data of 103 countries, regions, and organizations, including abstract data, full-text data, citations, and legal status data.

2.2. Intellectual Property Platforms in USA. The United States Patent and Trademark Office (USPTO) is an agency of the US Department of Commerce that issues patents to inventors and businesses for their inventions [18] and registers trademarks for the certification of invention products and IP property rights. Through trademark registration, the interests of commercial investment are protected, products and services are promoted, and the rights and interests of consumers are protected. Its main function is to process and disseminate patent and trademark information. The USPTO promotes the development of knowledge technology by issuing patents and provides rewards and compensation for inventions, investors of invention patents, and global public dissemination of new technologies.

2.3. Intellectual Property Platforms in European Union. The examiner of the European Patent Office (EPO) is responsible for studying, examining, and deciding whether to pass the European patent application submitted by the applicant and granting patents to the Contracting States of the European Patent Convention [19]. It is known as the European patent. Although EPO provides a single grant process, the granted patents are a series of national patents. In addition to granting European patents, it is also responsible for the national patent application search report generated on behalf of 13 national patent offices in France, Italy, the Netherlands, Belgium, and so forth. Retrieval and review procedures shall be carried out separately, and application information can be processed in a timely manner. In addition, domestic patent application can be applied for protection by multiple countries at the same time and can enjoy the same effect in any member country, using English, French, and German languages, with freedom of language.

2.4. Intellectual Property Platforms in Other Developed Countries. UK-Intellectual Property Office (UK-IPO) is set up by the British government [20]. Its function is to review and approve relevant legal IP rights (such as patents, designs, trademarks, and copyrights). Besides, coordinating the relevant work of government policy makers, law-enforcing departments and enterprises, jointly deal with criminal acts in the field of IP rights, and ensure the legitimacy of IP rights.

The Japanese Patent Office (JPO) is a government agency responsible for industrial property affairs, which is subordinate to the Ministry of economy and industry of Japan [21]. The Japanese patent office is one of the largest in the world. The main function of Japan's patent office is to manage laws related to patents, utility models, designs and trademarks, so as to promote the growth of Japan's economy and industry. The specific copyright affairs shall be managed by the Cultural Affairs Bureau.

Korea Intellectual Property Office (KIPO) established Korea Intellectual Property Rights Information Service (KIPRIS) centre and provided Internet Patent information retrieval service [22, 23]. Korean patent provided English version of Korean Patent Abstract (KPA) foreign utility model retrieval service. In addition, KIPRIS can provide patent (including patent and utility model) search, design search, trademark search, KPA search, Korean English machine translation and other services.

The German Patent and Trade Mark Office (DPMA) belongs to the jurisdiction of the Federal Ministry of justice and is the management centre of German industrial property [24]. DPMA is the largest intellectual property office in Europe and the fifth in the World Intellectual Property Office. DPMA's main responsibility is to confer, register, manage, and issue IP related information.

The Canadian Intellectual Property Office (CIPO) is an unusual running institution established by the Ministry of Innovation, Science and Economic Development of Canada [25]. CIPO is mainly responsible for the management of patent, trademark, copyright, industrial design, layout, and other implementation affairs of integrated circuit. Among them, CIPO patent department is responsible for patent application and authorization, and industrial design department is responsible for engineering registration of industrial design.

3. Intellectual Property Services Provision

3.1. Intellectual Property Searching and Retrieval Services. Intellectual property search and acquisition services are mainly for the search and acquisition of patents, trademarks, copyrights, and so forth. In the era of science and technology, information and data are growing explosively, and increasing Intellectual Property Rights are included in major patent offices around the world. In order to provide rapid and accurate screening for the public and facilitate institutions and groups, relevant institutions or enterprises in each country are trying to develop search tools for IP information and link to IP databases in different countries through the platform. And the search methods and functions applied to the platform are more and more mature [26, 27]. From the perspective of patent search, this paper introduces the commonly used IP search websites at home and abroad. The details are as follows.

The main page of the website of the SIPO has the function of patent search, as shown in Figure 1, which contains all patent information since the first patent application, including description items, abstracts, specifications and design graphics. There are three retrieval methods: simple retrieval, advanced retrieval, and PC classified retrieval. In addition, some Chinese search websites provide search services, such as China Patent Database, Wanfang Data Knowledge Service Platform, SooPAT Patent Search Engine, Patent Star Search Platform, Patent Information Service Platform, National Key Industry Service Platform, Hong Kong Intellectual Property Office, and Macao Economic Services.

The USPTO allows the user to retrieve all US authorized patents from 1790 to 1975 from two search portals, the patent number and the US patent classification number. At this stage, however, only full-text image pages are available. Since 1976, in addition to full-text image, all kinds of authorized patent documents in the USA can be retrieved from 31 search portals, including searchable basic description items, abstracts, and full-text patents, as shown in Figure 2.

The European Patent Office provides patent searchers with raw data from 70 countries and 40 patent offices, namely, bibliographic data in Figure 3. Its patent information service provides the translated version of the patent and provides legal consultation and prompt services throughout the patent application process. Patent retrieval is mainly completed by EPO Espacenet database. Other patent retrieval data can also be retrieved through Espacenet database, including patent family information, legal status information, citations, and other Asian patent retrieval data, which provides also a relatively unique function compared with other patent retrieval systems, that is, the search function option of Cooperative Patent Classification (CPC); Espacenet provides a more comprehensive tool and function for a better patent search structure.

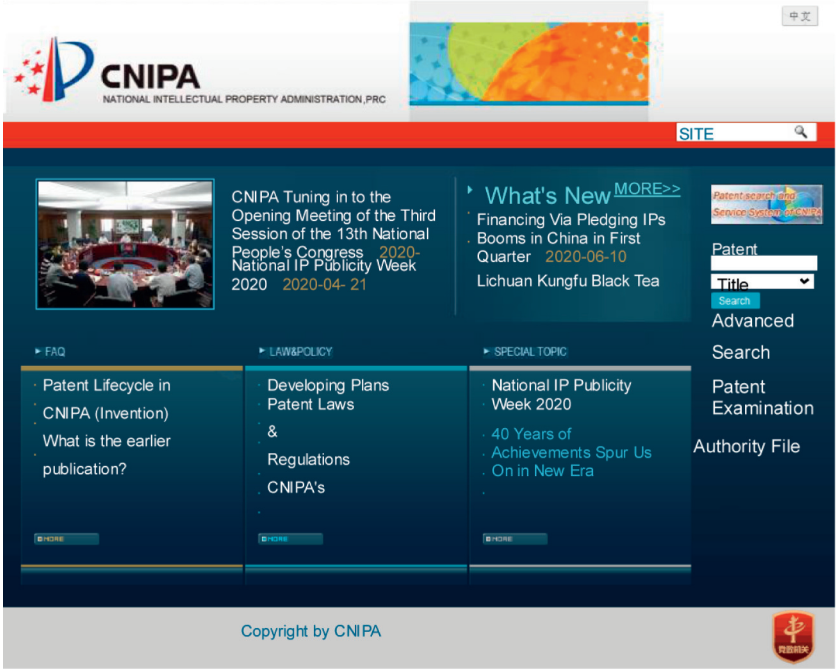
The UK Patent Office has a variety of collection channels, mainly providing six patent retrieval channels, namely, online patent document information service system (Ipsum), patent document publication retrieval, European Patent Office retrieval, Supplementary Protection Certificate (SPC) retrieval, patent classification retrieval, and patent review report retrieval, as shown in Figure 4.

Patent application data submitted by Japan is included in JPO database. The INPIT platform collects and stores global industrial property information and provides data consultation. In addition, Japan Patent Office (in Figure 5), and the patent related reference materials are retrieved from Japan Patent Information Platform (J-PlatPat), which is managed by INPIT. J-PlatPat allows access to patent map guidance (PMGs) and other tools and searches information from Patent Abstracts of Japan (PAJ) by keywords. It is worth mentioning that Japan Industrial Property Information Training Centre provides industrial property information particularly.

KIPRIS Retrieval System collects the patent and new use data published in Korea since 1948 and provides public applications and patents registering more than 2 million. It provides English search page and registered and unregistered users. Page navigation provides IP information (patent, design, trademark, and kpa) retrieval database connection, as shown in Figure 6, with the translation of Korean patent summary tools. The retrieval method has the functions of basic retrieval, intelligent retrieval, advanced retrieval, and the display of results.

DPMA, by publishing and search services DPMA register and DEPATISnet, enables the public to quickly learn about effective IP rights seen in Figure 7. DEPATISnet database is an online service offered by the GPTO. The underlying database applied is the DEPATIS information system database of the GPTO, which is also a retrieval database used internally by the GPTO. In this database, patent documents published by many countries and organizations can be retrieved and the full text of patent specification in PDF format can be browsed. In addition to the primary search, the German database also provides expert search and monitoring search.

The CIPO is in charge of the introduction of five kinds of IP rights, among which retrieval function includes patent, patent database, Patent Reexamination Board, industrial design, and industrial design database, to help users understand various IP-related laws, rules, application process, related costs, and other information, as shown in Figure 8.



CNIPA
NATIONAL INTELLECTUAL PROPERTY ADMINISTRATION,PRC

SITE

CNIPA Tuning in to the Opening Meeting of the Third Session of the 13th National People's Congress 2020-
National IP Publicity Week 2020 2020-04- 21

What's New MORE>>
Financing Via Pledging IPs Booms in China in First Quarter 2020-06-10
Lichuan Kungfu Black Tea

Patent search and Service System of CNIPA
Patent
Title
Search
Advanced Search
Patent Examination
Authority File

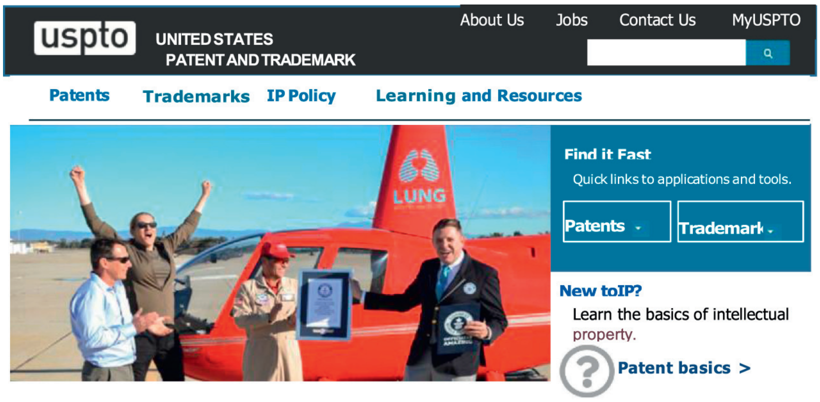
FAQ
Patent Lifecycle in CNIPA (Invention)
What is the earlier publication?

LAW&POLICY
Developing Plans
Patent Laws
&
Regulations
CNIPA's

SPECIAL TOPIC
National IP Publicity Week 2020
40 Years of Achievements Spur Us On in New Era

Copyright by CNIPA

FIGURE 1: Intellectual property information search page of CNIPA [17].



uspto UNITED STATES PATENT AND TRADEMARK

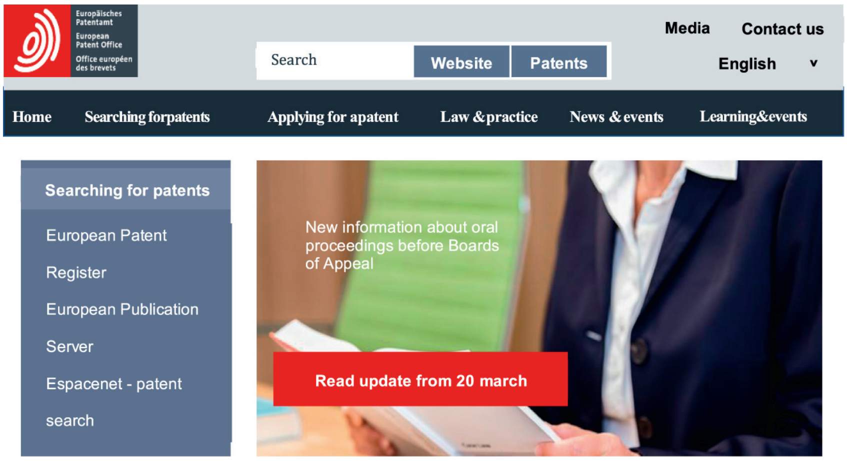
About Us Jobs Contact Us MyUSPTO

Patents Trademarks IP Policy Learning and Resources

Find it Fast
Quick links to applications and tools.
Patents
Trademark

New toIP?
Learn the basics of intellectual property.
Patent basics >

FIGURE 2: Intellectual property information search page of USPTO [18].



Europäisches Patentamt
European Patent Office
Office européen des brevets

Search Website Patents
Media Contact us
English

Home Searching forpatents Applying for apatent Law &practice News &events Learning&events

Searching for patents
European Patent
Register
European Publication
Server
Espacenet - patent
search

New information about oral proceedings before Boards of Appeal
Read update from 20 march

FIGURE 3: Intellectual property information search page of EPO [19].

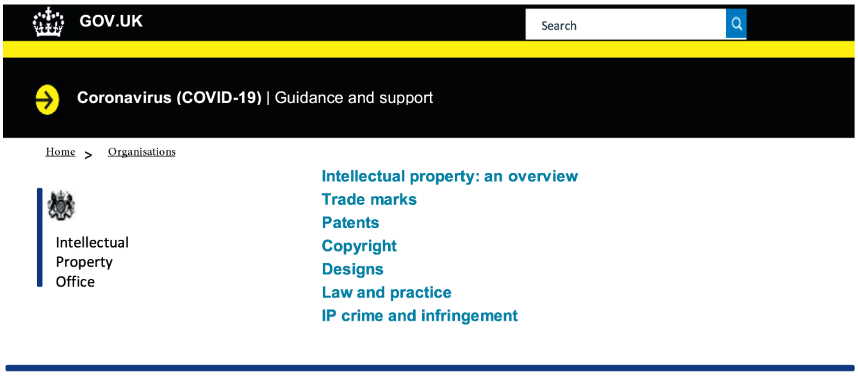
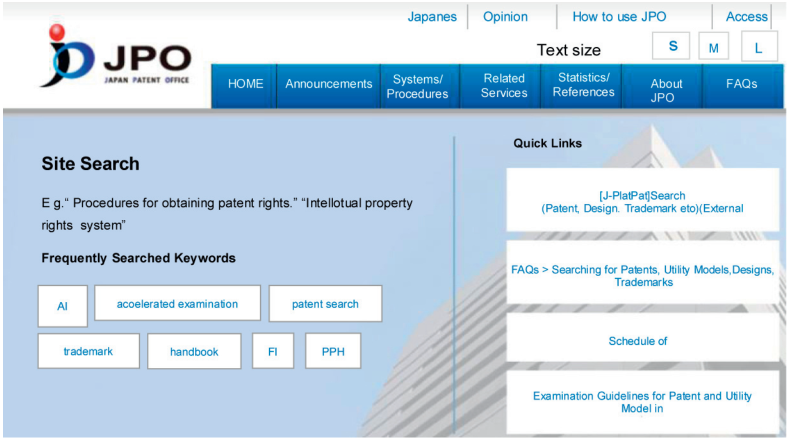

GOV.UK
Search
Coronavirus (COVID-19) | Guidance and support
Home > Organisations
Intellectual Property Office
Intellectual property: an overview
Trade marks
Patents
Copyright
Designs
Law and practice
IP crime and infringement

Figure 4: Intellectual property information search page of UK-IPO [20].



E g.“ Procedures for obtaining patent rights.” “Intellotual property rights system”

Frequently Searched Keywords
AI
acoelerated examination
patent search
trademark
handbook
FI
PPH
Quick Links
[J-PlatPat]Search (Patent, Design, Trademark eto)(External
FAQs > Searching for Patents, Utility Models,Designs, Trademarks
Schedule of
Examination Guidelines for Patent and Utility Model in

Figure 5: Intellectual property information search page of JPO [21].

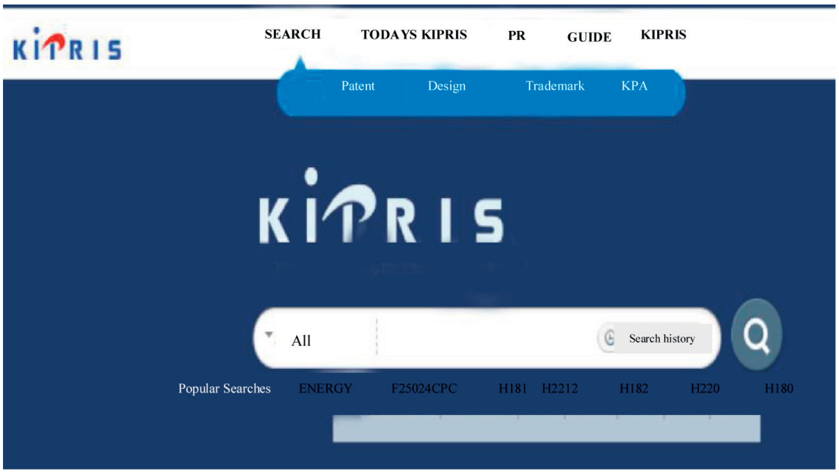


Figure 6: Search page of KIPRIS [23].

3.2. Intellectual Property Data Retrieval and Analysis Services. Nowadays, software development of various industries has been integrated into social life and developed rapidly, there are more and more advanced patent websites on the Internet [28], and more and more patent knowledge intelligent services. Some members of the public can use different platforms to obtain patent full text for free, skillfully use patent websites and conduct objective, in-depth, and comprehensive patent analysis. Patent analysis is to use data mining and information metrology to make statistics, comparison, and analysis of patent information and transform patent information into competitive intelligence with prediction function [29]. Based on a great quantity of patent information collected from intellectual property offices of various countries, the following summarizes the characteristics of commonly used patent analysis tools at

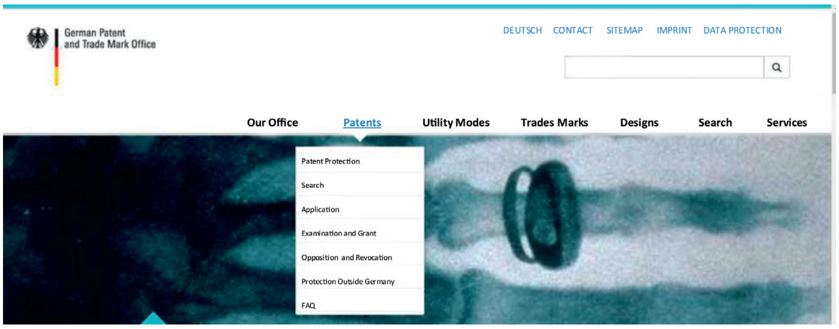
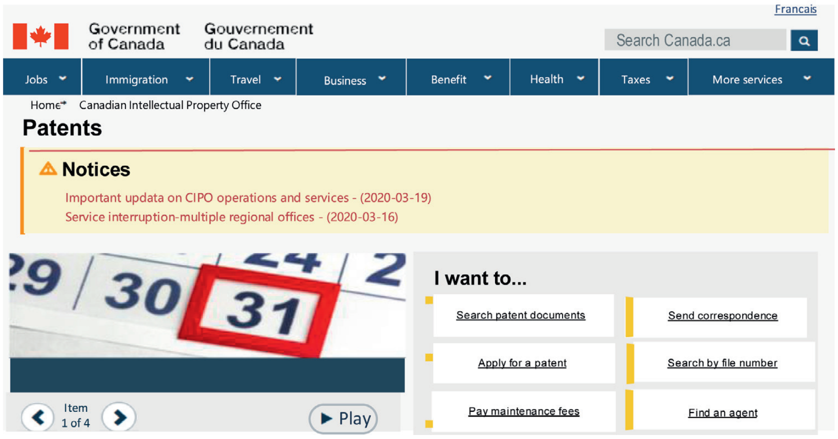


Figure 7: Intellectual property information search page of DPMA [24].



Patents

Notices

Important updata on CIPO operations and services - (2020-03-19)

Service interruption-multiple regional offices - (2020-03-16)

I want to...

Figure 8: Intellectual property information search page of CIPO [25].

home and abroad and compares different platforms to provide reference for researchers.

3.2.1. *Patent Analysis Tools in China*

(1) Patent Search and Analysis System. Patent search and analysis system is an independent system established by the CNIPA [30]. With the continuous upgrading and improvement in recent years, the system has become a widely used search and analysis platform for the public, covering patent data of 103 countries, regions, and organizations. It includes two subsystems: retrieval and analysis. The retrieval subsystem provides the general functions of general retrieval and table retrieval and also independently opens the drug subject retrieval function. The analysis subsystem mainly includes document database analysis, rapid analysis, customized analysis, advanced analysis, and other functions. Through the patent data analysis model provided by the system, users can effectively and comprehensively analyze the potential information relationship and complete patent information chain in patent data and improve the efficiency of patent information utilization. Customized analysis and advanced analysis provide more specialized functions such as technology evolution trend, enterprise positioning development analysis, list analysis, and matrix analysis, which are only used by advanced users at present.

(2) Patentics Retrieval and Analysis Platform. Patentics is the most advanced dynamic intelligent patent data platform system in the world, which integrates patent information retrieval, download, analysis, and management [31]. The foreign patent database retrieval is integrated into the platform, which is divided into three parts: Web version, client version, big data analysis module, patent operation analysis platform, and big patent analysis system. The patent retrieval system has the function of intelligent semantic retrieval. It can search related patents in the global patent database according to the semantics contained in the text content and sort them according to the relevant degree, which greatly improves the quality and efficiency of the retrieval. In addition, it has automatic translation search function in Chinese and English. At the same time, based on the research of intelligent classification of American patent classification and international patent classification, online intelligent analysis of patent information can realize real-time monitoring and intelligent patent early warning of competitors.

(3) Dawei Innojoy Patent Search Platform. Dawei Innojoy patent search engine is a patent information integrated application platform integrating global patent search, analysis, management, transformation, independent database building, and other functions [32]. At present, it is one of the professional patent search and analysis tools in China,

including more than 100 million business patent data in more than 100 countries. The patent search engine adopts advanced data mining and natural language processing technology, highly integrates patent literature resources of various countries, such as patent abstracts, specifications, legal status, family patents, and other information, supports more than one million level patent information for online analysis, and provides users with major special intellectual property review and technical development decision support. Figure 9 depicts the data search and analysis function diagram of this system.

(4) Hengheton Hit_Hengku. Hit_ Hengku is a patent analysis system independently developed by Hengheton, which integrates patent information retrieval, management, and analysis [33], as depicted in Figure 10. The analysis functions of this system include authorization information statistics, current technology ownership of competitors, technology information statistics, patent citation analysis, and patent value analysis. The original file can be converted into various general format files, which is easy to use and manage. In addition, the system can make a variety of statistical charts for patent description items and operate various command controls in images, such as color, data, 2D or 3D, title annotation, and legend controls. According to the needs of users, various forms of statistical reports are automatically generated, in which all kinds of important statistical information and charts are clearly presented to provide valuable information for users.

(5) Incopat. Incopat global technology analysis and operation platform is the first patent information platform integrating the world's top innovation intelligence and translating it into Chinese [34], which includes a large number of patent information from all over the world and integrates patent search and analysis, data and user management, and other functional modules. Incopat has collected more than one hundred million patent data in the world. It has collected and processed patent data of 22 major countries, with complete data field and high data quality. Incopat has comprehensive data integration and processing functions, with more than 230 search fields and multidimensional patent laws, references, and operational information. Incopat uses the methods of data mining and iterative optimization and uses more than 20 patent parameters to establish an objective value evaluation system. It enables users to pay attention to key information in time and rank the value of patents. Incopat's semantic retrieval adopts the advanced international deep learning algorithm and supports the input of voice, and the system automatically matches related patents, conducts clustering analysis according to the technical theme, displays the technical layout among competitors through various visualization methods such as competitive molecular map and patent topographic map, and forms a variety of analysis maps combining with a variety of analysis templates.

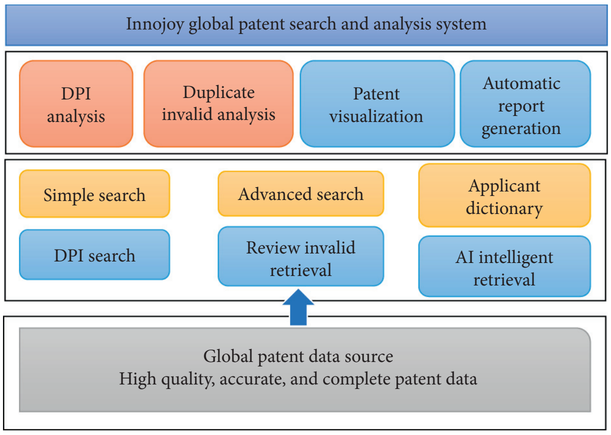

Innojoy global patent search and analysis system
DPI analysis
Duplicate invalid analysis
Patent visualization
Automatic report generation
Simple search
Advanced search
Applicant dictionary
DPI search
Review invalid retrieval
AI intelligent retrieval
Global patent data source
High quality, accurate, and complete patent data

Figure 9: Innojoy global patent search and analysis system diagram.

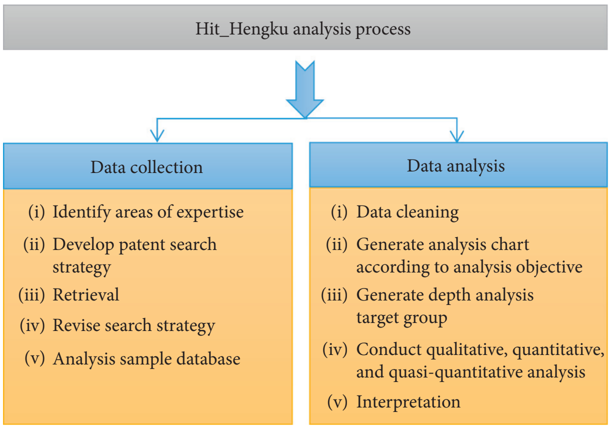

Hit_Hengku analysis process
Data collection
(i) Identify areas of expertise
(ii) Develop patent search strategy
(iii) Retrieval
(iv) Revise search strategy
(v) Analysis sample database
Data analysis
(i) Data cleaning
(ii) Generate analysis chart according to analysis objective
(iii) Generate depth analysis target group
(iv) Conduct qualitative, quantitative, and quasi-quantitative analysis
(v) Interpretation

Figure 10: Hit_Hengku analysis process diagram.

(6) PatSnap Smart Bud Global Patent Analysis and Retrieval System. PatSnap smart bud is a global patent retrieval database, which integrates patent information retrieval, management, and analysis and deeply integrates 140 million patent data of 116 countries and regions from 1790 to now. The update speed is timely. The specific functions include global patent data, full-text translation, advanced search, multidimensional browsing, insights patent analysis report, citation analysis, and patent value evaluation. Among them, insights patent analysis report means one click to generate patent analysis report to understand industry development and peer technology layout [35], shown in Figure 11.

3.2.2. Commercial Patent Analysis Tools in Foreign Countries

(1) The Derwent Analysis Software (DA). Derwent Analytics (DA) is a kind of software developed by former Thomson Reuters Company, which can deeply mine data and conduct visual analysis. It classifies information and data analyzes and summarizes them and has the characteristics of friendly and intuitive interface. It can analyze the industry trends according to the original data provided by Thomson Derwent patent database and provide the basis for comprehensively mastering the industry information [36]. The specific functions of DA include automatic summary, data

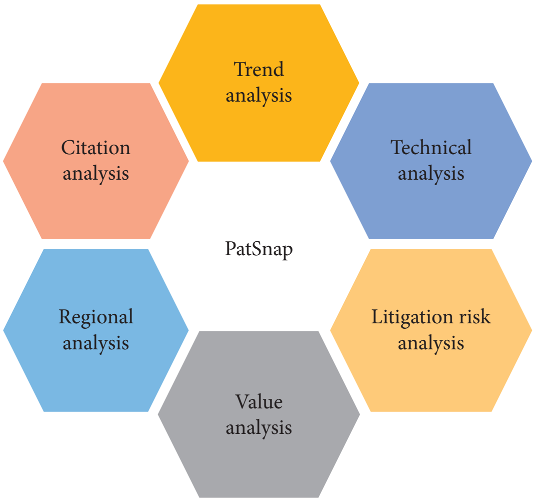


Figure 11: PatSnap search analysis function diagram.

sorting, comparison matrix, data mapping, preset analysis module (macro), and toolbox.

(2) Thomson Data Analyzer. Thomson Data Analyzer (TDA) is patent analysis software developed by Thomson Company [37]. It is the second generation of Derwent Analytics. It is text mining software with powerful analysis function, through which patent data can be deeply mined and visualized, and from a lot of patents to find competition and technical information, find new technology in the industry, and determine research strategy and development direction. In addition, TDA has the features of high degree of automation, friendly interface, and intuitionistic and has the functions of data import, management, cleaning, and analysis, as illustrated in Figure 12.

(3) VantagePoint. VantagePoint is a product of search analysis technology, which is a data mining tool developed by our search technology company to deeply mine patent information [38]. The data used in the system is purchased by the user directly from the data supplier. It mainly makes statistical analysis on various items in the data domain and provides data cleaning function. If the data field contains written text, the software uses natural language algorithm for subject analysis. This system uses many algorithms such as model matching, basic rules, and natural language processing technology to mine text. It allows the users to create user management dictionaries that refine specific data. Based on one-dimensional and two-dimensional analysis, VantagePoint provides multidimensional analysis function to help establish clustering or interrelationship in various relationships, depicted in Figure 13.

(4) STN AnaVist. STN AnaVist is software with a strong interactive analysis and visualization, which is developed by CAS itself. It provides a variety of types of scientific literature and tools for analyzing patent search results. From the analysis results, it can present the research mode and trend of the current industry. STN AnaVist can analyze search results from patent databases, such as multidisciplinary CAplus, USPATFULL, PCTFULL, and DWPI. And it can analyze the data of the CAplus and DWPI at the same time. From patent analysis, we can track competitive information, understand the latest application of existing technology, determine research trends, and provide strategic decision-making basis [39–41].

(5) Delphion. Delphion is a patent information service platform developed by Thomson Reuters in the United States [42]. It integrates five tools, snapshot, enterprise tree, Patentab II, text clustering, and citation link, and provides online analysis, list and histogram generation, document clustering, and citation analysis. Delphion has a wide range of patents and a variety of comprehensive analysis tools [43]. It has collected patent documents of several countries and also linked the world patent library and INPADOC (International Patent Document Centre) of Derwent Company. Patent retrieval methods include fast retrieval, patent number retrieval, Boolean retrieval, and advanced retrieval.

(6) Innography. Innography is a high-quality IP data used for patent analysis in the US Innography is patent search and analysis software with core patent mining function launched by ProQuest dialog company [44]. It is the latest patent retrieval and analysis platform based on the Internet, which contains patent literature information of major international intellectual property institutions. Its data sources include patents of more than 90 countries and regions, more than 80 million global patent data, business data, US patent litigation, and US business standard data, which provide support and help for patent analysis. Its characteristic functions are core patent mining based on patent intensity, similar patent retrieval, semantic retrieval, powerful analysis function, and visualization technology.

4. Discussions

4.1. Advantages of Present Intellectual Property Platforms. Each State Intellectual Property Office has different search characteristics and advantages [45]. Function comparison is conducted in Table 1 in order to provide detailed differences between these platforms. It takes patent search platforms in major intellectual property offices as an example to introduce the advantages of each platform in terms of information acquisition, search function, and characteristics.

It can be seen from Table 1 above that the patent collection information of each intellectual property office is relatively rich and the information acquisition channels are relatively comprehensive. Patent retrieval methods include basic retrieval, advanced retrieval, and patent number retrieval in different degrees and have different advantages and characteristics. In terms of information acquisition methods, different national patent offices can link databases of different types and institutions with a wide range of collection, so as to improve the quality of IP information. However, the language retrieval pages of each organization are different. For example, WIPO provides nine language retrieval interfaces, including China, Russia, Germany, and Japan.

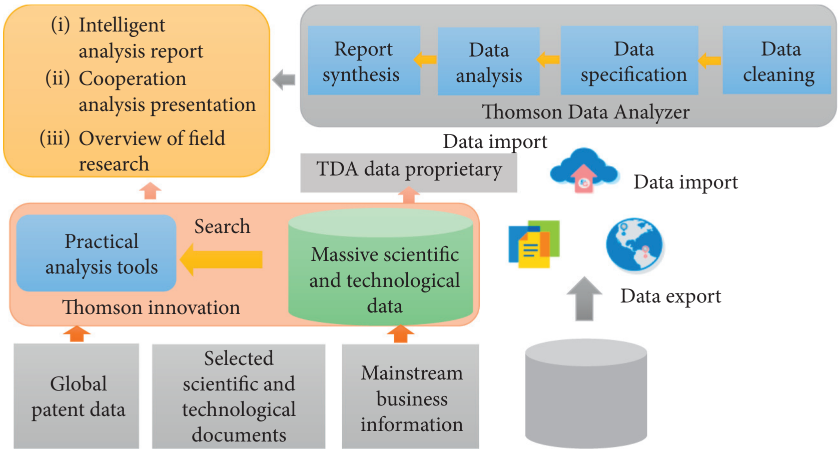

(i) Intelligent analysis report
(ii) Cooperation analysis presentation
(iii) Overview of field research
Report synthesis
Data analysis
Data specification
Data cleaning
Thomson Data Analyzer
Data import
TDA data proprietary
Data import
Practical analysis tools
Search
Massive scientific and technological data
Thomson innovation
Data export
Global patent data
Selected scientific and technological documents
Mainstream business information

Figure 12: TDA structure diagram.

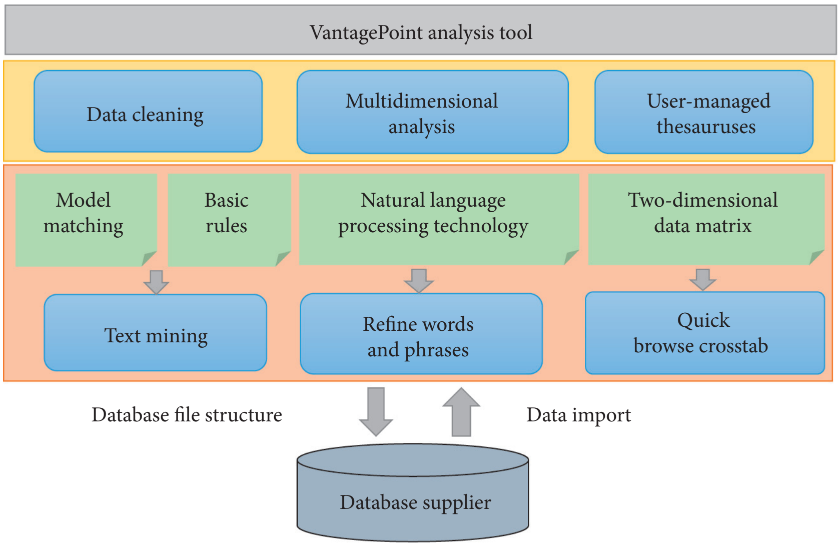


Figure 13: VatagePoint analysis tool diagram.

Espacenet provides Chinese, English, German, French, and Japanese retrieval interfaces. USPTO only includes US patents. In terms of retrieval methods, each platform supports basic and advanced retrieval. It is worth mentioning that USPTO provides retrieval of specific themes and images, bringing features and convenience to the public. CNIPA provides patent search, bilingual dictionary, and other functions. The Boolean logic retrieval provided by CIPO is also one of the highlights. In addition, some countries also provide automatic machine translation of patent documents, such as CNIPA, EPO, and WIPO. In the sorting of search results, CNIPA is sorted according to the rise and fall of application date and announcement date, while USPTO does not support sorting of search results.

4.2. Provision of Intellectual Property Search and Analysis Services. In addition to the intellectual property information collected by the Intellectual Property Office of the official governments of various countries, with numerous patent search and analysis methods or special tools, it is very convenient for the public to screen and analyze them [46, 47]. The patent analysis method is based on bibliometrics, using quantitative or qualitative analysis methods to process, count, and combine patent information to obtain competitive intelligence. It is mainly divided into four ways: basic statistical analysis, cluster analysis, cooccurrence analysis, and citation analysis.

(i) Statistical analysis: the method of quantitative analysis is used to make combination statistics of patent information to obtain information analysis method of technology dynamic development trend. The results are mainly presented in statistical reports, charts, etc.

(ii) Cluster analysis: gathering patents in different technical fields into different subcategories to understand the technical distribution in each technical field. The display methods include clustering map, structured data clustering, and unstructured data clustering.

(iii) Cooccurrence analysis: including cocitation, coword, and cocategory analysis, which is a quantitative analysis method of the information in the literature.

Table 1: Function comparison of patent search platform of different State Intellectual Property Offices.

No.	Patent databases	Information available	Search function	Advantages
1	CNIPA	The CNIPA owns all kinds of patent documents of more than 90 patent agencies and nearly half of patent specifications issued by patent agencies. China has announced more than 50000 patents. China has announced over 50000 patents	Search function field search, IPC category navigation search	It provides Chinese patent machine translation service with comprehensive information and fast and accurate retrieval information.
2	EPO	Espacenet contains more than 90 million files of various countries; the global patent index (GPI), as a complementary tool of Espacenet, is used to search global patents	Quick query, advanced search, number search, category search	It provides browsing options for the search function of CPC; compared with other patent search systems, it is a unique function. It provides patent legal status information directly linked to patent public reference documents and provides complete machine translation to translate patent documents.
3	WIPO	PATENTSCOPE contents include PCT electronic bulletin, Madrid application trademark database, US, Canada, European patent database, and Jopal technology journal database	Simple search, advanced search, by file combination, advanced cross language query	The search results are sorted by relevance for easy reference; support machine translation; provide legal information query; can be linked to the main databases of patents in the world. It provides full-paper search capabilities. There is no limit to the number of patent search results.
4	USPTO	The database is updated every Tuesday, providing a full-text image description of the film since 90 years and a full-text text description since 1976	Quick query, advanced search, official account query, fixed title query, number retrieval, picture retrieval	Complete database resources; access to patent information is very convenient, providing a rich information service platform, all-round service for innovation of small and medium-sized enterprises.
5	JPO	J-PlatPat publishes detailed information about the status of the whole process of patent application (e.g., examination and grant) and information publicity; PAJ (patent abstract of Japan) can also search for information	Quick search, joint search, multiple fields for category number or keyword search	Early Japanese patent documents are included completely, with huge data advantages and convenient retrieval methods. Patent texts included on the platform have basically been converted into text format and full-text application search fields; browsing interface can provide highlighting function, etc.
6	KIPO	KIPRIS provides patent (including patent and utility model) search, design search, trademark search, kPa search, Korean English machine translation, and other services. It updates data every day	Basic search, intelligent search, advanced search	Viewing the full text, legal status, priority, race, and other information, providing Korean patent summary translation, linking to domestic and foreign online patent databases, etc.
7	UK-IPO	Check the patent status to get the latest information about the patent application in the UK. Search for published patent applications and registered UK patents	Ipsum search methods include publication number, Espacenet, patent classification, and review date	Providing a variety of search paths and search systems and searching a comprehensive range. In addition, three kinds of patent information retrieval systems are provided, which can be retrieved through the entry of application number, name, classification number, keyword, etc.
8	DPMA	DEPATIS searches for global patent publications. It is mainly to archive the electronic files of patent retrieval platform and then show the reclassified IPC in the search results	"IPC" tag search or browse international patent classification number system, keyword or symbol search IPC through rating page, or IPC through IPC index search IPC through IPC number	It covers nearly 90 million patent documents. You can use the patent number to search for successful retrieval information.

Table 1: Continued.

No.	Patent databases	Information available	Search function	Advantages
9	CIPO	The database is provided by the Canadian Intellectual Property Office and contains patent documents since 1920. The classification of patent documents can be divided into two stages: The Canadian Patent Classification System (CPC) and then IPC (after October 1, 1989)	Basic query, patent number query, Boolean query, advanced query	Canada's patent search results are very rich, including almost all patent information. When retrieving, the number of Canadian patents hit out of all patents will be displayed so that users know total number of native patents and list of patents retrieved.

(iv) Citation analysis: analyzing the citation and citation of patents to reveal the interdependence between technology competitors, including both qualitative and quantitative analysis. The quality of citation analysis tools lies in the source of citation data and the presentation of citation results.

In addition, the main functions of patent search and analysis tool data include data monitoring, collection, cleaning, and processing. There are different intellectual property analysis tools at home and abroad to link countries' intellectual property information for retrieval and analysis services [48–50]. The services provided by patent analysis tools commonly used at home and abroad for patent information of official patent offices in various countries are compared, as discussed in Table 2, where eight indicators are provided for these patent analysis tools, including data monitoring, data collection, data cleaning, data processing, statistical analysis, cluster analysis, cooccurrence analysis, and citation analysis.

According to Table 2, almost all of the 12 patent analysis tools have data cleaning and statistical analysis functions. It is not clear whether Incopat and PatSnap have data cleaning function. More than half of the platforms provide data processing and citation analysis services. It is worth mentioning that, on the whole, patent retrieval and analysis services provided by China Patent Office are more comprehensive, but whether there are data monitoring and symbiosis analysis is uncertain. In terms of data monitoring, only PatSnap, Delphion, and innovation have it. In data collection and data processing, the foreign platform is obviously weaker than the domestic platform, especially in data collection, and the foreign analysis platform cited in this paper has obvious defects. But in the function of cluster analysis and symbiosis analysis, the foreign analysis platform is better than the domestic one. From the eight indicators, innovation has the least. The results show that each platform has its own main functions, and there are some defects and imperfections in its functions.

4.3. Limitation of Present Intellectual Property Platforms and Patent Analysis Tool

4.3.1. Limitation of Present Intellectual Property Platforms. There are some limitations in the above intellectual property offices. First, in CNIPA, the search speed and full-text download speed are relatively slow, the search results do not highlight keywords, and there is no choice to translate patent publications. Second, in EPO, patent search is only applicable to EP and WO files, and there are problems in using patent number to search information: compared with other databases, the maximum number of search words in each field is limited. Third, in WIPO, the data coverage is low for Espacenet; it is impossible to filter results based on publication date/applicant in the information retrieval of patent number retrieval. Fourth, in USPTO, full-paper search and other retrieve tools are only applicable to US patents. Fifth, in Japan Patent Information Platform, full-text search and other search tools are only applicable to JP and PCT files. Sixth, KIPO and KIPRIS are limited to special search and analysis tools, and activation of machine translation tools requires payment/subscription. Seventh, in UK, the Ipsum System is limited to specialized search and analysis tools. Finally, in DPMA, the full-text search of DEPATIS is only for patents, the search results are not highlighted with keywords, and there is no choice to translate patent publications.

4.3.2. Limitation of Present Patent Analysis Tool. In addition to the existing intellectual property information retrieval technology problems and challenges [51], data explosion in the network era, in particular patent in-depth analysis, has become a key research field for researchers or enterprises, combining with emerging science and technology and bringing new opportunities and challenges [52, 53]. The following mainly introduces the limitations from three aspects: cloud platform support, data encryption and privacy protection, and ontology model building knowledge system.

(1) Support of Cloud/Edge Platform. At present, there are few researches based on cloud platform and intellectual property protection at home and abroad, most of which stay in theoretical and experimental research. There are not many patent analysis products based on cloud platform in the market, which are not perfect, and there are technical loopholes [54–57]. There are few researches on patent analysis of cloud platform in China; besides, most of them are based on domestic cloud computing patents. However, there is no empirical research on the development, layout, and application of cloud platform technology at home and abroad.

(2) Data Encryption and Privacy Protection. Data protection is also one of the most important issues in the era of information explosion. More and more data protection

Table 2: Functional comparison and analysis methods of patent analysis tools.

No.	Names of analysis tools	Data monitoring	Data collection	Data cleaning	Data processing	Statistical analysis	Cluster analysis	Cooccurrence analysis	Citation analysis
1	Patent search and analysis system	—	✓	✓	✓	✓	✓	—	✓
2	Innojoy	—	✓	✓	✓	✓	✓	×	✓
3	Patentics	×	✓	✓	×	✓	✓	✓	✓
4	HIT_Hengku	×	✓	✓	✓	✓	×	×	✓
5	Incopat	×	✓	—	✓	✓	✓	—	✓
6	PatSnap	✓	✓	—	—	✓	✓	—	✓
7	Derwent Analytics	×	×	✓	✓	✓	×	✓	×
8	Thomson Data Analyzer	×	×	✓	✓	✓	✓	✓	×
9	Vantage point	×	×	✓	✓	✓	✓	✓	×
10	STN AnaVist	×	×	✓	×	×	✓	✓	×
11	Delphion	✓	×	✓	×	✓	✓	×	✓
12	Innography	✓	×	✓	×	×	×	×	×

methods are proposed by scholars [58, 59]. In terms of confidential storage of intellectual property data, homomorphic encryption technology has the best performance in security and functionality [60]. However, because its encryption algorithm needs a lot of complex exponential operation, which largely descends the data processing ability, the data processing technology based on encryption technology still needs to be improved. Besides, the vision of the exploration and application of big data has far exceeded the willingness to protect personal privacy. The high attention to the protection of IP information also makes privacy issues covered and ignored. The analysis of the above major database intellectual property protection platforms focuses more on the protection of data producers' rights and market interests but lacks the protection of data providers' privacy rights.

(3) Establishment of Ontology-Based Knowledge System. In the modern society with explosive data volume, the concepts contained in the massive intellectual property related data and the relationship between them need to be described by a standard semantic model so that computers can better understand them and then realize the sharing and reuse of existing knowledge. As a tool for modeling conceptual models, ontology has gradually become the basis of knowledge management and semantic web since it was proposed and plays an important role in Digital Library and other applications [61–63]. This is because domain ontology library can make existing systems better understand the information expressed in natural language at the semantic level. With the progress of ontology learning research and the development of data mining and other related technologies, the application of ontology in intellectual property information retrieval has gradually become a trend in the future, but the current research on the application of ontology technology in intellectual property information retrieval and analysis platform is not mature and perfect. Therefore, using ontology to build knowledge system in intellectual property data management will become the future development trend; as a result, this concept and method are not widely used in various intellectual property management platforms.

5. Conclusions

Data covered by global IP rights is growing explosively. It is of great significance for enterprises, governments, institutions, and scholars to effectively retrieve and analyze the information collected by IP offices of various countries. This paper summarizes the current situation of IP offices in various countries, as well as the retrieval, analysis, and visualization function integration platform of data and information collected by IP offices in various countries. Developers have developed a variety of patent analysis tools, showing different characteristics. In addition, this paper introduces the official IP platform and patent analysis tools at home and abroad in detail, compares different dimensions and corresponding indicators, discusses the advantages and disadvantages, and brings reference value to researchers. In addition, combined with the emergence of new technologies, in view of the current emerging technologies, namely, cloud computing, data encryption and privacy protection, ontology knowledge system research, and other aspects, this paper expounds the shortcomings of the existing analysis platform, considering the limitations of the existing technology and framework, IP retrieval, and analysis and other technology related fields have very broad research and development prospects.

Data Availability

This paper reviewed 8 Intellectual Property Platforms and 12 intellectual property analysis tools. All the data can be accessed from their official websites: http://english.sipo.gov.cn, https://www.gov.uk/government/organisations/intellectual-property-office, https://www.epo.org/index.html, https://www.gov.uk/government/organisations/intellectual-property-office, https://www.jpo.go.jp/e/index.html, https://www.kipo.go.kr/en/MainApp?c=1000, http://eng.kipris.or.kr/enghome/main.jsp, https://www.dpma.de/english/index.html, http://www.ic.gc.ca/eic/site/cipointernet-internetopic.nsf/eng/h_wr00001.html, https://www.thevantagepoint.com/, http://www.stn-international.com/stn_anavist.html, and http://www.delphion.com.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by NSSFC, China (Grant no. 19ZDA078) and the Fund of Research for Decision Making and Consultation of Jiangsu Service-Oriented Government Construction, China.

References

[1] W. Yu, W. Wang, P. Jiao, H. Wu, Y. Sun, and M. Tang, "Modeling the local and global evolution pattern of community structures for dynamic networks analysis," *IEEE Access*, vol. 7, pp. 71350–71360, 2019.

[2] J. Jin, L. Ma, and X. Ye, "Digital transformation strategies for existed firms: from the perspectives of data ownership and key value propositions," *Asian Journal of Technology Innovation*, vol. 28, no. 1, pp. 77–93, 2020.

[3] H. M. Wu, X. Y. Li, and Y. J. Deng, "Deep learning-driven wireless communication for edge-cloud computing: opportunities and challenges," *Journal of Cloud Computing*, vol. 9, no. 21, pp. 1–14, 2020.

[4] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[5] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2020.

[6] X. Xu, B. Shen, X. Yin et al., "Edge server quantification and placement for offloading social media services in industrial cognitive IoV," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.

[7] A. Brem, P. A. Nylund, and E. L. Hitchen, "Open innovation and intellectual property rights: how do SMEs benefit from patents, industrial designs, trademarks and copyrights?" *Management Decision*, vol. 55, no. 6, pp. 1285–1306, 2017.

[8] L. Ma, Z. Liu, X. J. Huang et al., "The impact of local government policy on innovation ecosystem in knowledge resource scarce region: case study of Changzhou, China," *Science, Technology & Society*, vol. 24, no. 1, pp. 1–24, 2019.

[9] P. R. Merges, "What kind of rights are intellectual property rights?" in *Oxford Handbook of IP Law*, R. Dreyfuss and J. Pila, Eds., pp. 1–50, Oxford University Press, Oxford, UK, 2017.

[10] J. Yin, Z. Ge, and W. Song, "Research on the construction of intellectual property operation platform under the background of "internet +"," *Technology and Investment*, vol. 8, no. 4, pp. 179–194, 2017.

[11] J. List, "Current trends and future directions for IP Information research," *World Patent Information*, vol. 52, pp. A1–A2, 2018.

[12] L. Ma, J. Yu, and H. M. Song, "Research on innovation superhighway based on driving strategies," *Science & Technology and Economy*, vol. 24, no. 6, pp. 1–6, 2011.

[13] Y. Chen, Z. Y. Liu, and L. X. Su, "A theoretical framework of internet enabled country/region "discovery-innovation" system," *Studies in Science of Science*, vol. 32, no. 2, pp. 170–177, 2014.

[14] M. Lupu, "Information retrieval, machine learning, and natural language processing for intellectual property information," *World Patent Information*, vol. 49, pp. A1–A3, 2017.

[15] X. Xu, Q. Liu, X. Zhang, J. Zhang, L. Qi, and W. Dou, "A blockchain-powered crowdsourcing method with privacy preservation in mobile environment," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1407–1419, 2019.

[16] H. Wu, W. J. Knottenbelt, and K. Wolter, "An efficient application partitioning algorithm in mobile environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 7, pp. 1464–1480, 2019.

[17] China National Intellectual Property Administration, 2020, http://english.sipo.gov.cn/.

[18] United States Patent and Trademark Office, 2020, https://www.uspto.gov/.

[19] European Patent Office, 2020, https://www.epo.org/index.html.

[20] Gov.UK, 2020, https://www.gov.uk/government/organisations/intellectual-property-office.

[21] Japan Patent Office, 2020, https://www.jpo.go.jp/e/index.html.

[22] Korean Intellectual Property Office, 2020, https://www.kipo.go.kr/en/MainApp?c=1000.

[23] Korea Intellectual Property Rights Information Service, 2020, http://eng.kipris.or.kr/enghome/main.jsp.

[24] German Patent and Trade Mark Office, 2020, https://www.dpma.de/english/index.html.

[25] Canadian Intellectual Property Office, 2020, http://www.ic.gc.ca/eic/site/cipointernet-internetopic.nsf/eng/h_wr00001.html.

[26] W. Shalaby and W. Zadrozny, "Patent retrieval: a literature review," *Knowledge and Information Systems*, vol. 61, pp. 631–660, 2019.

[27] L. Ma, T. Li, J. X. Wu et al., "The impact of E-hailing competition on the urban taxi ecosystem and governance strategy from a rent-seeking perspective: the China E-hailing platform," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 4, no. 35, 2018.

[28] A. Tolstaya, I. Suslina, and P. Tolstaya, "Information provision of patent research," *Biosciences, Biotechnology Research Asia*, vol. 13, no. 3, pp. 1479–1491, 2016.

[29] S. Ranaei, A. Knutas, J. Salminen, and A. Hajikhani, "Cloud-based patent and paper analysis tool for comparative analysis of research," in *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016*, pp. 315–332, Palermo, Italy, June 2016.

[30] Q. X. Xu, R. Wang, and B. Q. Liu, "SIPO patent retrieval resources," *Shandong Chemical Industry*, vol. 43, no. 5, pp. 65–69, 2014.

[31] L. Yu and X. Y. Zhao, "Search strategy of patentics intelligent search system in the field of analysis and detection," *Chinese Invention and Patent*, vol. 16, no. 11, pp. 104–110, 2019.

[32] L. X. Zhang and W. Li, "Statistical analysis of patent results based on Innojoy," *Journal of Changchun University of Technology*, vol. 39, no. 6, pp. 613–618, 2018.

[33] J. Guo and K. Shi, "On the application of patent search in patent search system," *Chinese Invention and Patent*, vol. 14, no. 8, pp. 123–127, 2017.

[34] C. L. Sun and L. B. Sun, "IncoPat-based patent analysis of China pharmaceutical university," *Journal of China Pharmaceutical University*, vol. 50, no. 3, pp. 374–378, 2019.

[35] D. Liang and Y. F. Wang, "Analysis of low-fat food research and development patents by PatSnap," *Journal of Food Safety and Quality*, vol. 10, no. 8, pp. 2420–2424, 2019.

[36] A. Barth, T. Stengel, E. Litterst et al., "A novel concept for the search and retrieval of the derwent markush resource database," *Journal of Chemical Information and Modeling*, vol. 56, no. 5, pp. 821–829, 2016.

[37] Z. Zhang and Y. Guo, "The development and application of analysis tool of TDA (Thomson data analyzer)," *Shanxi Science and Technology*, vol. 28, no. 4, pp. 103-104, 2013.

[38] VatagePoint," 2020, https://www.thevantagepoint.com/.

[39] G. Fischer and N. Lalyre, "Analysis and visualisation with host-based software—the features of STN®AnaVist™," *World Patent Information*, vol. 28, no. 4, pp. 312–318, 2006.

[40] STN_Anavist, 2020, http://www.stn-international.com/stn_anavist.html.

[41] Y. Z. Cao and S. Z. Li, "Introduction of STN anavist analysis function in STN system," *Patent Literature Research*, vol. 1, no. 2, pp. 46–51, 2008.

[42] M. K. Raturi, P. K. Sahoo, and A. K. Tiwari, "Delphion: a world class patent database—a comprehensive analysis from patent information professional's perspective," *SSRN Electronic Journal*, pp. 1–11, 2009.

[43] Delphion, 2020, http://www.delphion.com.

[44] H. Tang and H. Yang, "Analysis of patent information of supercapacitor based on innography," in *Proceedings of the 2018 International Conference on Robots & Intelligent System (ICRIS)*, pp. 321–325, Changsha, China, 2018.

[45] J. Kim and S. Lee, "Patent databases for innovation studies: a comparative analysis of USPTO, EPO, JPO and KIPO," *Technological Forecasting and Social Change*, vol. 92, pp. 332–345, 2015.

[46] Y. H. Chen and S. T. Wang, "Comparison of patentics and Incopat in semantic search," *China Invention & Patent*, vol. 16, no. 3, pp. 125–128, 2019.

[47] Z. M. Xie, "An comparative study on the effectiveness of patent value evaluation tools," *Modern Information*, vol. 38, no. 4, pp. 124–129, 2018.

[48] M. Burhan and S. K. Jain, "Tools for search, analysis and management of patent portfolios," *DESIDOC Journal of Library & Information Technology*, vol. 32, no. 3, pp. 204–213, 2012.

[49] X. Jain, Z. Y. Peng, and B. Liu, "Summary of patent search and analysis," *Journal of Wuhan University (Engineering Edition)*, vol. 47, no. 3, pp. 420–425, 2014.

[50] S. Ozcan and N. Islam, "Patent information retrieval: approaching a method and analysing nanotechnology patent collaborations," *Scientometrics*, vol. 111, no. 2, pp. 941–970, 2017.

[51] L. P. Zhi and B. Q. Shi, "Research on the function mechanism of patent information in the patent operation of enterprises in China," *Humanities and Social Science Research*, vol. 1, no. 1, p. 60, 2018.

[52] J. Shin, S. Lee, and T. Wang, "Semantic patent analysis system based on big data," in *Proceedings of the 2017 IEEE 11th International Conference on Semantic Computing (ICSC)*, pp. 284-285, San Diego, CA, USA, 2017.

[53] D. Prokhorenkov and P. Panfilov, "Notice of violation of IEEE publication principles: discovery of technology trends from patent data on the basis of predictive analytics," in *Proceedings of the 2018 IEEE 20th Conference on Business Informatics (CBI)*, vol. 2, pp. 148–152, Vienna, Austria, 2018.

[54] X. Wang, Y. Yang, J. Zhang et al., "Design and application of the business hosting cloud platform faced to intellectual property service provider," in *Proceedings of the 2015 5th International Conference on Communication Systems and Network Technologies*, pp. 1000–1004, Gwalior, India, April 2015.

[55] J.-Y. Huang, "Patent portfolio analysis of the cloud computing industry," *Journal of Engineering and Technology Management*, vol. 39, pp. 45–64, 2016.

[56] C. Cheng, P. H. Lyu, and C. Fu, "Cloud computing knowledge domains mining based on patent networks," in *Proceedings of the 24th International Conference on Industrial Engineering and Engineering Management 2018*, G. Huang, CF. Chien, and R. Dou, Eds., pp. 677–686, Changsha, China, 2019.

[57] M. Bayramusta and V. A. Nasir, "A fad or future of IT?: a comprehensive literature review on the cloud computing research," *International Journal of Information Management*, vol. 36, no. 4, pp. 635–644, 2016.

[58] L. L. Zhang, X. B. Hu, and F. Liu, "Patent analysis of data security protection technology for cloud storage platform," *Technology Outlook*, vol. 26, no. 23, pp. 268–270, 2016.

[59] G. Guo, T. Yang, and Y. Liu, "Search engine based proper privacy protection scheme," *IEEE Access*, vol. 6, pp. 78551–78558, 2018.

[60] M. Yang, Y. Man, N. Liu et al., "Design of searchable algorithm for biological databased on homomorphic encryption," *Lecture Notes in Computer Science*, vol. 11956, pp. 537–545, Springer, Berlin, Germany, 2019.

[61] K. H. Law, S. Taduri, G. T. Law et al., "An ontology-based approach for retrieving information from disparate sectors in government: the patent system as an exemplar," in *Proceedings of the 2015 48th Hawaii International Conference on System Sciences*, pp. 2096–2105, Kauai, HI, USA, January 2015.

[62] C.-P. Phan, H.-Q. Nguyen, and T.-T. Nguyen, "Ontology-based heuristic patent search," *International Journal of Web Information Systems*, vol. 15, no. 3, pp. 258–284, 2019.

[63] X. J. Chen, S. Y. Cheng, X. R. Yang, and X. W. Zhang, "Establishment of patent knowledge function model based on function ontology," *Journal of Guangdong University of Technology*, vol. 36, no. 2, pp. 26–30, 2019.