

Wireless Communications and Mobile Computing

IoT Big Data Analytics

Lead Guest Editor: Salimur Choudhury

Guest Editors: Qiang Ye, Mianxiong Dong, and Qingchen Zhang





IoT Big Data Analytics

Wireless Communications and Mobile Computing

IoT Big Data Analytics

Lead Guest Editor: Salimur Choudhury

Guest Editors: Qiang Ye, Mianxiong Dong, and Qingchen Zhang



Copyright © 2019 Hindawi. All rights reserved.

This is a special issue published in “Wireless Communications and Mobile Computing.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

Javier Aguiar, Spain
Ghufran Ahmed, Pakistan
Wessam Ajib, Canada
Muhammad Alam, China
Eva Antonino-Daviu, Spain
Shlomi Arnon, Israel
Leyre Azpilicueta, Mexico
Paolo Barsocchi, Italy
Alessandro Bazzi, Italy
Zdenek Becvar, Czech Republic
Francesco Benedetto, Italy
Olivier Berder, France
Ana M. Bernardos, Spain
Mauro Biagi, Italy
Dario Bruneo, Italy
Jun Cai, Canada
Zhipeng Cai, USA
Claudia Campolo, Italy
Gerardo Canfora, Italy
Rolando Carrasco, UK
Vicente Casares-Giner, Spain
Luis Castedo, Spain
Ioannis Chatzigiannakis, Italy
Lin Chen, France
Yu Chen, USA
Hui Cheng, UK
Ernestina Cianca, Italy
Riccardo Colella, Italy
Mario Collotta, Italy
Massimo Condoluci, Sweden
Daniel G. Costa, Brazil
Bernard Cousin, France
Telmo Reis Cunha, Portugal
Laurie Cuthbert, Macau
Donatella Darsena, Italy
Pham Tien Dat, Japan
André L. F. de Almeida, Brazil
Antonio De Domenico, France
Antonio de la Oliva, Spain
Gianluca De Marco, Italy
Luca De Nardis, Italy
Liang Dong, USA
Mohammed El-Hajjar, UK
Oscar Esparza, Spain

Maria Fazio, Italy
Mauro Femminella, Italy
Manuel Fernandez-Veiga, Spain
Gianluigi Ferrari, Italy
Ilario Filippini, Italy
Jesus Fontecha, Spain
Luca Foschini, Italy
A. G. Fragkiadakis, Greece
Sabrina Gaito, Italy
Óscar García, Spain
Manuel García Sánchez, Spain
L. J. García Villalba, Spain
José A. García-Naya, Spain
Miguel Garcia-Pineda, Spain
A.- J. García-Sánchez, Spain
Piedad Garrido, Spain
Vincent Gauthier, France
Carlo Giannelli, Italy
Carles Gomez, Spain
Juan A. Gómez-Pulido, Spain
Ke Guan, China
Antonio Guerrieri, Italy
Daojing He, China
Paul Honeine, France
Sergio Ilarri, Spain
Antonio Jara, Switzerland
Xiaohong Jiang, Japan
Minho Jo, Republic of Korea
Shigeru Kashiara, Japan
Dimitrios Katsaros, Greece
Minseok Kim, Japan
Mario Kolberg, UK
Nikos Komninos, UK
Juan A. L. Riquelme, Spain
Pavlos I. Lazaridis, UK
Tuan Anh Le, UK
Xianfu Lei, China
Hoa Le-Minh, UK
Jaime Lloret, Spain
Miguel López-Benítez, UK
Martín López-Nores, Spain
Javier D. S. Lorente, Spain
Tony T. Luo, Singapore
Maode Ma, Singapore

Imadeldin Mahgoub, USA
Pietro Manzoni, Spain
Álvaro Marco, Spain
Gustavo Marfia, Italy
Francisco J. Martinez, Spain
Davide Mattera, Italy
Michael McGuire, Canada
Nathalie Mitton, France
Klaus Moessner, UK
Antonella Molinaro, Italy
Simone Morosi, Italy
Kumudu S. Munasinghe, Australia
Enrico Natalizio, France
Keivan Navaie, UK
Thomas Neue, Ireland
Tuan M. Nguyen, Vietnam
Petros Nicopolitidis, Greece
Giovanni Pau, Italy
Rafael Pérez-Jiménez, Spain
Matteo Petracca, Italy
Nada Y. Philip, UK
Marco Picone, Italy
Daniele Pinchera, Italy
Giuseppe Piro, Italy
Sara Pizzi, Italy
Vicent Pla, Spain
Javier Prieto, Spain
Rüdiger C. Pryss, Germany
Sujan Rajbhandari, UK
Rajib Rana, Australia
Luca Reggiani, Italy
Daniel G. Reina, Spain
Jose Santa, Spain
Stefano Savazzi, Italy
Hans Schotten, Germany
Patrick Seeling, USA
Muhammad Z. Shakir, UK
Mohammad Shojafar, Italy
Giovanni Stea, Italy
Enrique Stevens-Navarro, Mexico
Zhou Su, Japan
Luis Suarez, Russia
Ville Syrjäla, Finland
Hwee Pink Tan, Singapore





Pierre-Martin Tardif, Canada
Mauro Tortonesi, Italy
Federico Tramarin, Italy
Reza Monir Vaghefi, USA

Juan F. Valenzuela-Valdés, Spain
Aline C. Viana, France
Enrico M. Vitucci, Italy
Honggang Wang, USA

Jie Yang, USA
Sherali Zeadally, USA
Jie Zhang, UK
Meiling Zhu, UK

Contents


IoT Big Data Analytics

Salimur Choudhury , Qiang Ye, Mianxiong Dong , and Qingchen Zhang
Editorial (1 page), Article ID 9245392, Volume 2019 (2019)



Hybrid Parallel FDTD Calculation Method Based on MPI for Electrically Large Objects

Qingwu Shi, Bin Zou , Lamei Zhang , and Desheng Liu
Research Article (9 pages), Article ID 7309431, Volume 2019 (2019)

Robust and Privacy-Preserving Service Recommendation over Sparse Data in Education

Xuening Chen, Hanwen Liu, Yanwei Xu, and Chao Yan 
Research Article (13 pages), Article ID 2401857, Volume 2019 (2019)





Roads and Intersections Extraction from High-Resolution Remote Sensing Imagery Based on Tensor Voting under Big Data Environment

Ke Sun , Junping Zhang , and Yingying Zhang
Research Article (11 pages), Article ID 6513418, Volume 2019 (2019)

A Selective Mirrored Task Based Fault Tolerance Mechanism for Big Data Application Using Cloud

Hao Wu , Qinggeng Jin , Chenghua Zhang, and He Guo
Research Article (12 pages), Article ID 4807502, Volume 2019 (2019)





Predicting Fine-Grained Traffic Conditions via Spatio-Temporal LSTM

Xiaojuan Wei , Jinglin Li , Quan Yuan , Kaihui Chen, Ao Zhou , and Fangchun Yang
Research Article (12 pages), Article ID 9242598, Volume 2019 (2019)

Attribute Reduction Based on Genetic Algorithm for the Coevolution of Meteorological Data in the Industrial Internet of Things

Yong Cheng, Zhongren Zheng, Jun Wang, Ling Yang, and Shaohua Wan 
Research Article (8 pages), Article ID 3525347, Volume 2019 (2019)

Homomorphic Evaluation of the Integer Arithmetic Operations for Mobile Edge Computing

Changqing Gong , Mengfei Li , Liang Zhao , Zhenzhou Guo, and Guangjie Han 
Research Article (13 pages), Article ID 8142102, Volume 2018 (2019)

Editorial

IoT Big Data Analytics

Salimur Choudhury¹,¹ Qiang Ye,² Mianxiong Dong³,³ and Qingchen Zhang⁴

¹Lakehead University, Thunder Bay, Canada

²University of Prince Edward Island, Charlottetown, Canada

³Muroran Institute of Technology, Muroran, Hokkaido, Japan

⁴St. Francis Xavier University, Antigonish, Canada

Correspondence should be addressed to Salimur Choudhury; salimur.choudhury@lakeheadu.ca

Received 14 July 2019; Accepted 15 July 2019; Published 30 July 2019

Copyright © 2019 Salimur Choudhury et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A successful Internet of things (IoT) environment requires standardization that contains interoperability, compatibility, reliability, and effectiveness of the operations on a global scale. The rapid growth of the IoT causes a sharp growth of data. Enormous amounts of networking sensors are continuously collecting and transmitting data to be stored and processed in the cloud. Such data can be environmental data, geographical data, astronomical data, logistic data, and so on. Mobile devices, transportation facilities, public facilities, and home appliances are the primary data acquisition equipment in IoT. The volume of such data will surpass the capacities of the IT architectures and infrastructure of existing enterprises and, due to real-time analysis character, will also greatly impact the computing capacity. Management of these increasingly growing data is a challenge for the community in general. Due to the generation of big data by IoT, the existing data-processing capacity of IoT is becoming ineffective, and it is imperative to incorporate big data technologies to promote the development of IoT. It is important to understand that the success of IoT lies upon the effective incorporation of big data analytics. The widespread deployment of IoT also gives a challenge to big data community to propose new techniques since both big data and IoT are interdependent themselves. On the one hand, the widespread deployment of IoT provides data both on quantity and on category, thus providing the opportunity for the application and development of big data; on the other hand, the incorporation of big data analytics in IoT simultaneously accelerates the research advances and business models of IoT.

This special issue provides some exciting papers in this area covering topics like the fault tolerant mechanism for big data applications using the cloud, hybrid parallel load-balancing algorithm for big data application, privacy-preserving service recommendation over sparse data, and genetic algorithm for the coevolution of meteorological data in the industrial Internet of things.

Conflicts of Interest

The editors declare that they have no conflicts of interest regarding the publication of this special issue.

Salimur Choudhury
Qiang Ye
Mianxiong Dong
Qingchen Zhang

Research Article

Hybrid Parallel FDTD Calculation Method Based on MPI for Electrically Large Objects

Qingwu Shi,^{1,2} Bin Zou ,¹ Lamei Zhang ,¹ and Desheng Liu²

¹Department of Information Engineering, School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China

²College of Information Science & Electronic Technique, Jiamusi University, Jiamusi 15407, China

Correspondence should be addressed to Bin Zou; zoubin@hit.edu.cn and Lamei Zhang; lmzhang@hit.edu.cn

Received 5 December 2018; Revised 22 April 2019; Accepted 14 May 2019; Published 23 June 2019

Guest Editor: Qiang Ye

Copyright © 2019 Qingwu Shi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

At present, the Internet of Things (IoT) has attracted more and more researchers' attention. Electromagnetic scattering calculation usually has the characteristics of large-scale calculation, high space-time complexity, and high precision requirement. For the background and objectives of complex environment, it is difficult for a single computer to achieve large-scale electromagnetic scattering calculation and to obtain corresponding large data. Therefore, we use Finite-Difference Time-Domain (FDTD) combined with Internet of Things, cloud computing, and other technologies to solve the above problems. In this paper, we focus on the FDTD method and use it to simulate electromagnetic scattering of electrically large objects. FDTD method has natural parallelism. A computing network cluster based on MPI is constructed. POSIX (Portable Operating System Interface of UNIX) multithreading technology is conducive to enhancing the computing power of multicore CPU and to realize multiprocessor multithreading hybrid parallel FDTD. For two-dimension CPU and memory resources, the Dominant Resource Fairness (DRF) algorithm is used to achieve load balancing scheduling, which guarantees the computing performance. The experimental results show that the hybrid parallel FDTD algorithm combined with load balancing scheduling can solve the problem of low computational efficiency and improve the success rate of task execution.

1. Introduction

The rapid growth of the IoT causes a sharp growth of data. Enormous amounts of networking sensors are continuously collecting and transmitting data to be stored and processed in the cloud [1, 2]. Such data can be remote sensing data, geographical data, and astronomical data. Radar is one type of the important remote sensing data, which plays a significant role in the society. Radar raw echo data must meet specific conditions on radar system design, imaging processing algorithm, geometric distortion correction, and other occasions. If these data are obtained through flight radar carrier, the cost is too high. Therefore, it is an important solution to obtain the required echo data through signal simulation [3]. The main method of obtaining echo data depends on the development of computational electromagnetics.

The core problem of electromagnetic computing is to achieve high precision and high efficiency calculation and to

ensure accuracy while reducing costs. Traditional distributed computing and parallel computing methods use accumulated CPU resources in local area networks to achieve high-performance collaborative computing. However, the electromagnetic computing tasks that need to be solved are becoming more and more complex. Exploring elastic computing based on Internet of Things and cloud computing technology has become an economical, feasible, and efficient solution [4, 5]. After evaluating the characteristics of computational tasks with predictive learning model, it can schedule computational resources and improve computational efficiency and success by using adaptive load balancing method [6–9].

The calculation accuracy and stability conditions restrict the discrete size of FDTD in time and space. When calculating electrically large objects, especially, the serial FDTD method cannot meet the requirements in engineering because of the long time and excessive memory requirements [10, 11]. Therefore, expanding the calculation scale and

reducing the calculation time becomes the key to scattering calculation and simulation.

Aiming to solve the above problems, our novelties in this paper are as follows:

- (1) POSIX multithreading technology is utilized to improve the computing power of multicore CPU.
- (2) The DRF algorithm is used to achieve load balancing scheduling; this can guarantee the computing performance.
- (3) We design a novel calculation and exchange strategy in the same time to save computing time.

Based on the characteristics of FDTD suitable for parallel computing, a parallel FDTD method based on MPI (MPI: Message Passing Interface) and POSIX (POSIX: Portable Operating System Interface of UNIX) is proposed. Taking the resource called and load balancing allocation in the COW (COW: Cluster of Workstation) into consideration, the DRF (DRF: Domain Resource Fairness) algorithm is adopted to achieve load balancing scheduling for electromagnetic computing tasks in terms of CPU and memory resources.

2. The Basic Principle of Parallel FDTD Algorithm

2.1. Maxwell's Equations. The FDTD algorithm was proposed by Yee in 1966. The starting point of FDTD algorithm is that the curl equation in Maxwell equation is differentiated and the electromagnetic field is discretized in space and time. Therefore, when deriving the FDTD

equation, only two curl equations [12] need to be considered, namely,

$$\nabla \times \mathbf{E} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J} \quad (1)$$

$$\nabla \times \mathbf{E} = \frac{\partial \mathbf{B}}{\partial t} - \mathbf{J}_m \quad (2)$$

In Cartesian coordinates, (1) and (2) can be written as

$$\begin{aligned} \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} &= \varepsilon \frac{\partial E_x}{\partial t} + \sigma E_x \\ \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} &= \varepsilon \frac{\partial E_y}{\partial t} + \sigma E_y \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} &= \varepsilon \frac{\partial E_z}{\partial t} + \sigma E_z \end{aligned} \quad (3)$$

And

$$\begin{aligned} \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} &= -\mu \frac{\partial H_x}{\partial t} - \sigma_m H_x \\ \frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} &= -\mu \frac{\partial H_y}{\partial t} - \sigma_m H_y \\ \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} &= -\mu \frac{\partial H_z}{\partial t} - \sigma_m H_z \end{aligned} \quad (4)$$

The spatial arrangement of the discrete electric and magnetic fields in the FDTD is shown in Figure 1.

2.2. FDTD Iteration Formula. In Cartesian coordinates, (3) and (4) are deduced. Taking (3) as an example, the three-dimensional FDTD iteration formulas are obtained.

$$\begin{aligned} E_x^{n+1} \left(i + \frac{1}{2}, j, k \right) &= CA(m) \cdot E_x^n \left(i + \frac{1}{2}, j, k \right) \\ &+ CB(m) \cdot \left[\frac{H_z^{n+1/2} (i + 1/2, j + 1/2, k) - H_z^{n+1/2} (i + 1/2, j - 1/2, k)}{\Delta y} - \frac{H_y^{n+1/2} (i + 1/2, j, k + 1/2) - H_y^{n+1/2} (i + 1/2, j, k - 1/2)}{\Delta z} \right] \end{aligned} \quad (5)$$

In (5), the coefficients are related to the medium parameters and the discrete parameters. It can be seen from Figure 1 and (5) that each magnetic field component is surrounded by four electric field components. Similarly, each electric field component is surrounded by four magnetic field components, and the electric field and the magnetic field are alternately sampled in time series, sampling interval. It is half a time step away from each other so that it can be solved iteratively in time [12].

2.3. Parallelism of FDTD Method. The electric/magnetic field values of each grid are only correlated with the magnetic /electric field values of the adjacent grid. Therefore, a complete computational domain can be divided into several

subdomains to exchange data of tangential fields at the boundary of adjacent subdomains. There is no correlation between the grid points in the subdomain and other subdomains, and each subdomain can execute independently.

Firstly, the electric field component is calculated. As shown in Figure 2, at the N time step, the four magnetic fields needed to calculate E_z in subregion 1 are within the region 1, so the calculation of E_z can be performed correctly without any additional operation. However, for E_z located on the boundary of region 1, the four magnetic fields H_y needed are located in subregion 2. So, at the $N - 1/2$ time step, H_y must be passed from subregion 2 to subregion 1 to sure that E_z on the boundary of region 1 can be iterated correctly.

TABLE 1: MPI Function and Description.

Function Name	Description
MPI_Init()	Start MPI computing environment
MPI_Comm_size()	Number of parallel processes
MPI_Comm_rank()	Self-process identification number
MPI_Send()	Send a message to the numbered process
MPI_Recv()	Receive messages from a numbered process
MPI_Finalize()	End MPI running environment

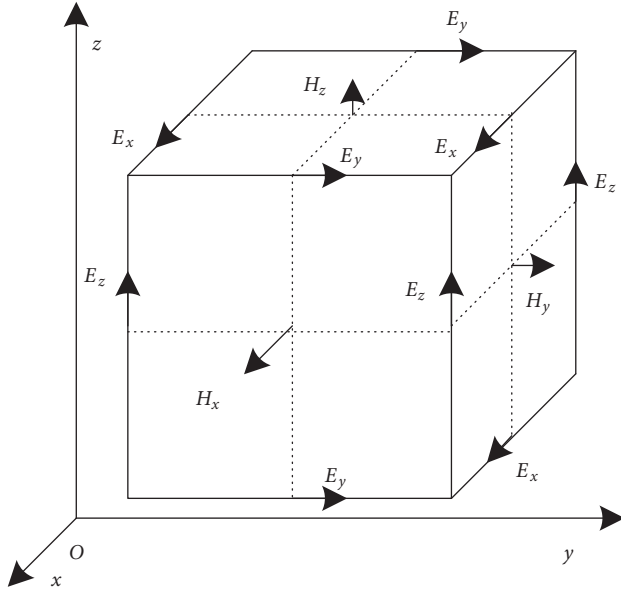


FIGURE 1: Discrete Yee Cellular of FDTD.

Similarly, at the $N+1/2$ time step to calculate the magnetic field component H_y on the boundary of subregion 2, it needs the electric field component E_z on the boundary of subregion 1 of the N time step. So, when E_z is computed, E_z must be passed from subregion 1 to subregion 2 to ensure the correct iteration of H_y . It is through the alternating transmission of electric and magnetic fields in time that the correct iteration of field components on the boundary of each sub-region is ensured, and the parallelization of the algorithm is realized.

The field value exchange process based on domain decomposition parallel algorithm has the typical characteristics of message transmission, so it is very convenient to implement network parallel operation in the cluster of microcomputers by means of MPI communication function.

3. MPI Function Libraries and Local Area Network Configuration

3.1. Common Library Functions and Communication Modes of MPI. MPI is one of the standards for message passing parallel programming. MPI provides a library that can interface with C language [13]. In parallel computing, there are six common used functions, including starting and ending the

MPI environment, identifying process, and sending and receiving messages. In the MPI program, each node has a unique process identification number. The master node allocates computing tasks according to the process identification number. Different processes, that is, nodes, perform different tasks in parallel, and each node uses the MPI_Comm_rank() function. It obtains its own process identification number and runs this process. At the same time, exchanged data between each process is required to realize parallel computing of the program [14]. The detailed explanation of function is shown in Table 1.

MPI provides two communication modes: blocking communication mode and nonblocking communication mode.

Blocking Send. The process is allowed to continue executing the next statement only after it is determined that the message has been sent to the message buffer.

Block Receive. Unless the message is received accurately in the message buffer, the process terminates the pending state and continues to execute the next instruction.

For the blocking mode, if the frequency of communication operation calls and the time of calls are too high or too long, the computing unit will be waiting for interruption long time, and the computing resources will not be fully utilized, thus reducing the calculation efficiency.

Nonblocking Send. It sends the primitive to inform the system that the message has been in the message buffers and then return. The sending process can continue to perform subsequent work without waiting for the system to actually send the message.

Nonblocking Receive. The primitive will be received and returned regardless of whether there is a notification of sending the primitive in the message buffer.

In nonblocking message transmission, while a message is sent or received accurately, the system will use interruption signal to inform the sender or receiver. Before that, the system can periodically query, suspend the process, or execute other instructions, achieving overlap between computation and communication, thereby, improving the computational efficiency of the system. However, using nonblocking messaging will make program debugging difficult.

3.2. Local Area Network Configuration and Structure. MPI-based parallel computing needs to build a computing network. The typical parallel computing structure of cluster is

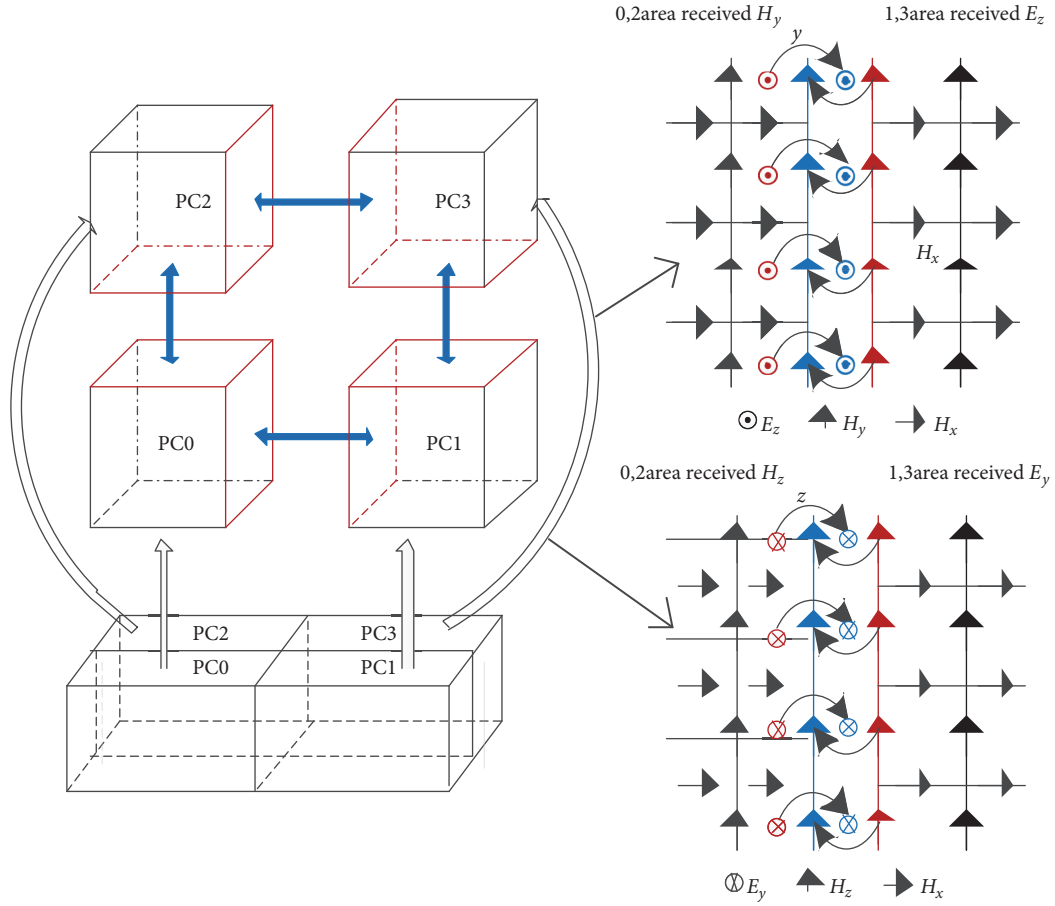


FIGURE 2: Parallel FDTD computational sketch of two-dimensional domain decomposition.

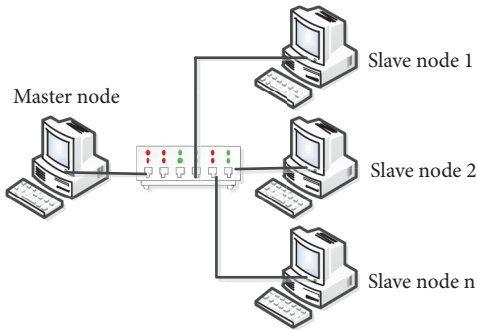


FIGURE 3: Network topology for parallel computing.

usually used to build MPI computing network, as shown in Figure 3.

4. Several Key Problems in Parallel FDTD Programming

4.1. Computational Region Segmentation of Parallel FDTD. Before parallel FDTD computation, it is necessary to divide the computational region of electromagnetic field into several

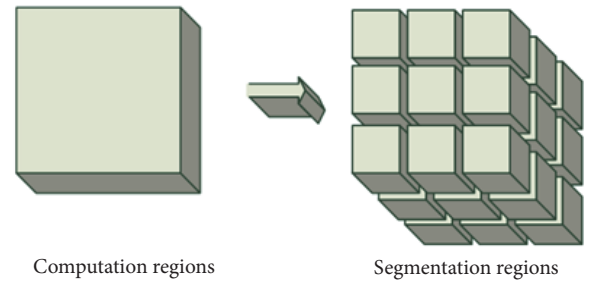


FIGURE 4: Divided according to the three-dimensional mode.

computational subregions according to certain topological rules [15]. Each computing node in the cluster calculates one or several subregions. The electromagnetic data on the boundary surface is exchanged amongst nodes. Finally the master node collects these electromagnetic fields and saves them in the data file. Generally, spatial topological structures can be divided into one-dimensional, two-dimensional, and three-dimensional structures. In this experiment, the computational area is divided into three-dimensional, as shown in Figure 4.

4.2. Load Balancing amongst Nodes. Load balancing is the main indicator to measure the performance of a cluster

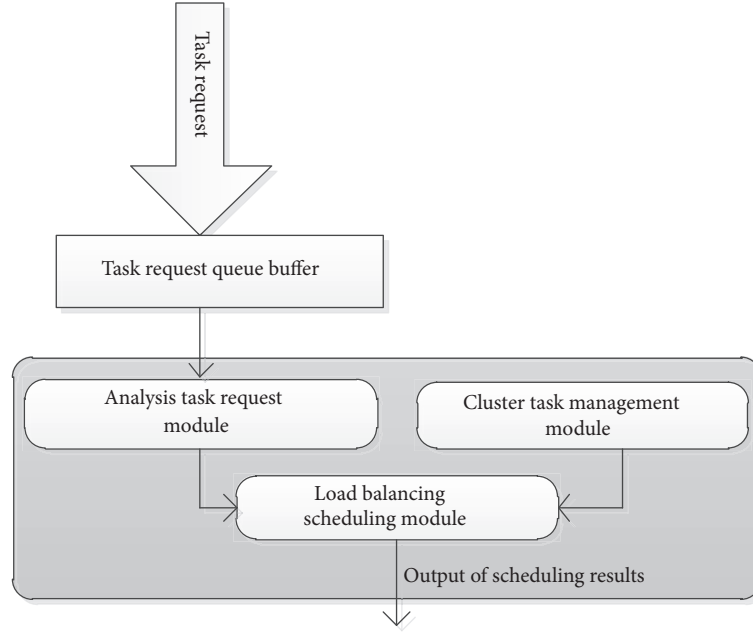


FIGURE 5: Resource scheduling principle diagram.

system, which refers to the allocation of computing tasks amongst nodes, so that each process can complete computing tasks at the same time, thus reducing the maximum running time of the process. For resource scheduling for large-scale electromagnetic computing tasks, achieving load balancing is more important. Load balancing amongst resources ensure that resources are not overloaded, thereby, improving resource performance, improving the successful execution rate of electromagnetic computing tasks, and improving the performance of high-reliability resource scheduling mechanisms. Load balancing can be implemented in parallel programming, which effectively reduces the running time of parallel programs and improves the speedup and program performance [16].

This paper mainly focuses on the two dimensions of CPU and memory resources. The load balancing DRF (Dominant Resource Fairness) algorithm is used to design the scheduling mechanism to achieve load balancing scheduling for electromagnetic computing tasks. The resource scheduling block diagram is shown in Figure 5.

According to the above principles and Figure 4, the cluster task management module collects the status information of electromagnetic computing platform resources, extracts information such as CPU utilization rate and memory utilization rate of each cluster resource, and reports it to the electromagnetic computing platform control centre which provides the corresponding resource acquisition interface. The task request processing module receives allocation request from the electromagnetic computing platform control centre. The dispatching request sent by the control centre parses and operates different types of computing tasks, generates corresponding sequence of task requests, and reports to the load balancing scheduling module. Based on the above, we can get the flow chart of the whole process as shown in Figure 6.

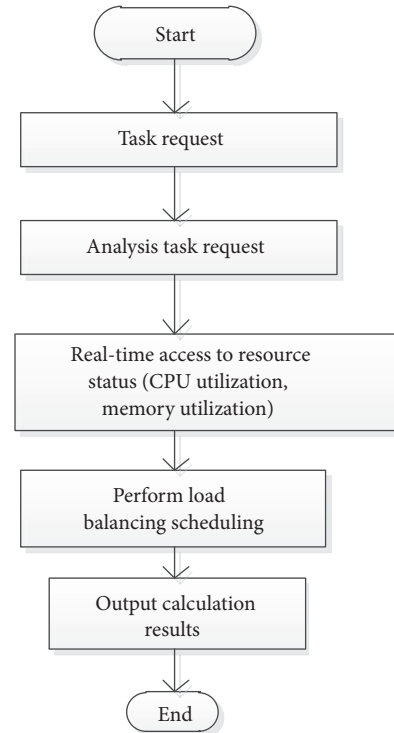


FIGURE 6: Resource scheduler flow chart.

4.2.1. Building System Model. The experiment is conducted on Homogeneous Network Environments. In theory, it can be assumed that the total amount of computing tasks is T_{total} . The total number of nodes is N and each node's RAM is M in the cluster. The total resource is N processors and $N * M$

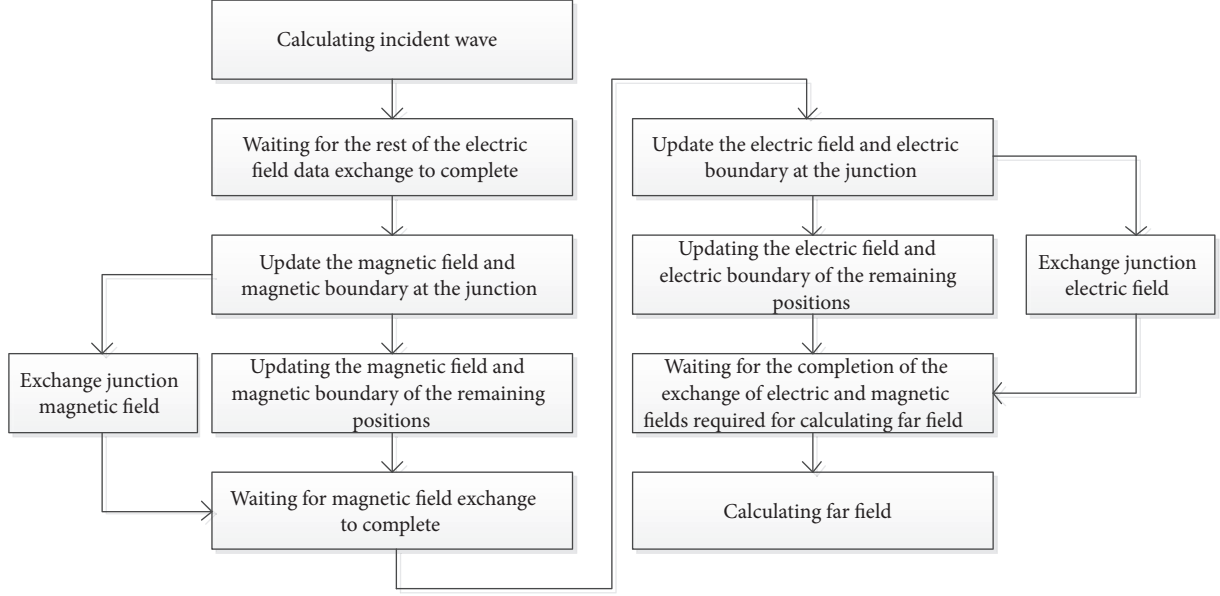


FIGURE 7: Flow charts of programs computing and exchanging data simultaneously.

RAM number. Suppose they are all involved in computing. Then the amount of tasks assigned to each computing node is T_{N_i} :

$$T_{N_i} = \frac{T_{total}}{N} \quad (6)$$

In the heterogeneous network, resource of each node is different. Assuming that there are k nodes performing computational tasks, the amount of resources allocated to K nodes is T_1, T_2, \dots, T_k , respectively. The total amount of computing tasks is T_{total} . The coefficients of the tasks allocated by each node are $\alpha_1, \alpha_2, \dots, \alpha_K$. The processor number of each node is (c_1, c_2, \dots, c_k) and the RAM number of each node is (m_1, m_2, \dots, m_k) . $S = \sum_{i=1}^k c_i * m_i$. The DRF algorithm can be used to give a fair strategy considering multiresource load balancing. The optimization equation is as follows:

$$\begin{aligned} \max \quad & (T_1, T_2, \dots, T_k) \\ \sum_{i=1}^k T_i & \leq T_{total} \\ \sum_{i=1}^k \alpha_i T_i & \leq S \\ \sum_{i=1}^k \alpha_i & = 1 \end{aligned} \quad (7)$$

where \max denotes the maximum resource allocation. The first inequality constraint is that it cannot exceed the total number of the tasks. The second inequality constraint is that it cannot exceed the total resource number. Equality constraint represents the main share of the balanced resources in two dimensions.

4.3. Hybrid Programming of Multithreads in Nodes. In order to make full use of the multicore of the CPU, multicore parallel computing is implemented by using shared memory inside a single node. In the shared memory programming mode, the CPU can access the data in the shared memory, and the cores of the CPU do not need to transfer data to each other. POSIX is a portable multithreaded library. POSIX-based Pthreads multithreading technology can implement internal parallelism of a single node. This standard provides a way to create and terminate new threads. The created thread executes a given function [17]. At the same time, the standard also provides three synchronization mechanisms for semaphores, condition variables, and mutexes, which make it easy to synchronize amongst threads.







When designing the program, the FDTD algorithm only needs to exchange the electromagnetic data of a grid at the junction of nodes. If the blocking exchange mode is adopted, the time required for data exchange will become an additional overhead of the program. This reduces the efficiency of program execution. In order to improve the efficiency of program operation, the method of simultaneous calculation and data exchange is adopted in the design of program. The flow chart is shown in Figure 7.

In the hybrid parallel FDTD program, the two steps of “updating the magnetic field and magnetic boundary of the remaining positions” and “updating the electric field and electric boundary of the remaining positions” in Figure 6 is the maximum calculation. Because the electromagnetic field needed to be exchanged at the junction has only one grid layer, the computation amount is only 1% or even smaller than that of the whole electrically large object. Therefore, in order to avoid creating and destroying threads frequently, which results in a large amount of time overhead, the program does not destroy the threads of these two steps in the whole iteration process. When calculating these two steps, the

TABLE 2: Simulation platform and simulation parameters.

Simulation Platform	Simulation Parameters
HP-Z600 graphics workstations	incident wave: L-band
CPU: Inter (R) Xeon (R) CPU E5640@2.67GHz (2 processor)	centre frequency:1.3GHz
RAM:8GB	bandwidth: 1GHz
OS: Windows 7 Professional Edition	pulse width:1 μ s
	grid size:0.023m

TABLE 3: Model information and the SAR imaging results of models.

model	3-D information	(L*W*H)(cm)	Grid number	Visible grid number	SAR Imaging results
dumbbell		140*40	376608	20899	
car		440*159*138	387791	22103	
Helicopter		500*510*320	662550	36991	

program uses the synchronization mechanism of Pthreads to perform multithreaded parallel computation. In the other steps, only one thread occupies the CPU, while the other threads wait for conditional variables.

4.4. Performance Evaluation of Parallel Computing. The performance of parallel computing is usually measured by speedup and efficiency. Suppose a serial program runs for T_s time. The running time of a parallel program on p processors is T_p . The definition of the speedup is [18]

$$S_p = \frac{T_s}{T_p} \quad (8)$$

The definition of efficiency is

$$E_p = \frac{S_p}{p} \quad (9)$$

Here, p is the number of processors in all nodes.

5. Experimental Simulation and Data Analysis

5.1. Experimental Hardware Platform. In order to verify the correctness and acceleration performance of the method, the simulation experiment uses several HP-Z600 graphics workstations to form a cluster through a Gigabit switch. Three models are calculated: dumbbell, car, and helicopter. The cluster configuration is shown in Table 2.

The computed model information and the SAR imaging results are shown in Table 3.

In the simulation process, one computer serial FDTD calculation is used as the comparison reference. Three computers and four computers are used for hybrid parallel computing. The data of the calculation are recorded, as shown in Table 4. * denotes that it cannot finish the task.

5.2. Result Analysis. Assuming that the speedup of serial computing is 1 and through analysing Table 4, the conclusions can be as follows.

(1) When using one computer to calculate serial FDTD, the calculation time and speed of FDTD are only related to the main frequency of CPU and its calculation ability. At this time, the Windows operating system will retain some CPU resources, so the CPU utilization cannot reach 100%, and the actual CPU utilization can reach 90%.

(2) Under the same excitation source, the larger the size of the object and the more grids it gets, the more memory it needs. For the same purpose, parallel FDTD computing requires much less memory for each node in the cluster than serial FDTD computing, which reflects the superiority of parallel computing. Therefore, parallel computing is an effective method to solve the scattering problem of electrically large objects, which requires a lot of computing resources.

(3) The computational efficiency of the program (assuming that the efficiency of serial is 1) is related to the number of nodes in the system network. Experiments show that with the increase of the number of computing nodes, the computational efficiency will decrease correspondingly. The main reason why this phenomenon is appearing is that the CPU utilization of the intermediate nodes is usually low

TABLE 4: FDTD calculation and comparison of different models.

Calculation method	Objects	Time steps	time (s)	CPU usage rate (%)	one using memory (MB)	Speedup	Efficiency
one, serial	dumbbell		7245	90	1250	1	1
	Car		8902	90	1532	1	1
three, parallel	helicopter		*	*	*	*	*
	dumbbell		3832	75~88	350~430	1.891	0.6303
	car	26069	4705	78~90	430~530	1.892	0.6307
	helicopter		39842	68~72	280~300	1.785	0.5951
four, parallel	dumbbell		2912	68~78	260~350	2.488	0.6220
	car		3646	65~82	330~430	2.442	0.6105
	helicopter		25751	65~78	230~260	2.052	0.5131

after the increase of the number of computing nodes, and it also increases the additional consumption of time when the system exchanges data.

(4) In parallel FDTD computing, there is no limit on the number of nodes in the network, but the experiment shows that with the increase of the number of nodes, the computing time is shortened and the speedup is increased, which exactly reflects the advantages of parallel FDTD computing. In addition, it should be emphasized that the relationship between shortening the computing time and increasing the number of nodes is nonlinear. With the increasing of the number of nodes, the time needed for data exchanges in the network takes up more CPU computing time, and it can also explain the reason why the efficiency of the system decreases.

6. Conclusion

This paper mainly studies the hybrid parallel FDTD algorithm to solve the electromagnetic scattering calculation of electrically large objects. In the design of the program and algorithm, the rules of calculating region partition and load balancing algorithm are applied comprehensively. The program achieves parallel computing in cluster and multi-threads computing in single node. The experimental results and analysis conclude that hybrid parallel FDTD algorithm with load balancing can improve the computing speed and greatly shorten the computing time. Meanwhile, the parallel algorithm can also provide ideas and solutions for solving the shortage of computing resources. Compared with the serial FDTD algorithm, it has great advantages and engineering application value.

Data Availability

The data will be perfected, so it will be released in the future.

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 (61401124, 61871158), in part by Scientific Research Foundation for the Returned Overseas Scholars of Heilongjiang Province (LC2018029), in part by Foundation for Returnees of Heilongjiang Province of China (NO: LC2017027), and in part by Jiamusi University Science and Technology Innovation Team Construction Project (Project Number: CXTPDY-2016-3).

References

- [1] J. Gao, P. Li, and Z. Chen, "A canonical polyadic deep convolutional computation model for big data feature learning in Internet of Things," *Future Generation Computer Systems*, vol. 99, pp. 508–516, 2019.
- [2] J. Gao, J. Li, and Y. Li, "Approximate event detection over multi-modal sensing data," *Journal of Combinatorial Optimization*, vol. 32, no. 4, pp. 1002–1016, 2016.
- [3] Y. Shi, L. Li, and Y. Yu, "Simulation technique of the synthetic aperture radar," *Journal of Telemetry, Tracking and Command*, vol. 28, 2007.
- [4] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, "A double deep q-learning model for energy-efficient edge scheduling," *IEEE Transactions on Services Computing*, 2018.
- [5] Q. Zhang, L. T. Yang, Z. Chen, P. Li, and F. Bu, "An adaptive dropout deep computation model for industrial IoT big data learning with crowdsourcing to cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2330–2337, 2018.
- [6] P. Li, Z. Chen, L. T. Yang et al., "Deep convolutional computation model for feature learning on big data in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 790–798, 2018.
- [7] P. Li, Z. Chen, L. T. Yang, J. Gao, Q. Zhang, and J. Deen, "An incremental deep convolutional computation model for feature learning on industrial big data," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1341–1349, 2018.
- [8] Q. Zhang, L. T. Yang, Z. Yan, Z. Chen, and P. Li, "An efficient deep learning model to predict cloud workload for industry informatics," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3170–3178, 2018.
- [9] P. Li, Z. Chen, L. T. Yang, J. Gao, Q. Zhang, and M. J. Deen, "An improved stacked auto-encoder for network traffic flow classification," *IEEE Network*, vol. 32, no. 6, pp. 22–27, 2018.
- [10] S. Jiang, Z. Lv, Y. Zhang et al., "Analysis of parallel performance of MPI based parallel FDTD on supercomputer," in *Proceedings of the IET International Radar Conference*, pp. 1–4, Xi'an, China, 2013.
- [11] W. Yu, X. Yang, Y. Liu et al., "New development of parallel conformal FDTD method in computational electromagnetics engineering," *IEEE Antennas and Propagation Magazine*, vol. 53, no. 3, pp. 15–41, 2011.
- [12] D. Ge and Y. Yan, *Finite-Difference Time-Domain Method for Electromagnetic Waves*, Xidian University Press, Xi'an, China, 3rd edition, 2011.
- [13] Z. Duo, *High Performance Computing Parallel Programming Technology-MPI Parallel Programming*, Tsinghua University Press, Beijing, China, 2001.
- [14] Z. Lu, J. Zhang, J. Shi et al., "Dynamic load balancing strategies in MPI parallel environment," *Computer Technology and Development*, vol. 20, no. 5, pp. 133–135, 2010.
- [15] H. Li, *Research of Parallel Algorithm Based on Lan*, Harbin Institute of Technology, Harbin, China, 2011.
- [16] L. Kezhong and L. Xiaohui, "Implementing load balance in MPI parallel program," *Microcomputer Information*, vol. 23, no. 5-3, pp. 226–227, 2007.
- [17] G. Chen, *Parallel Computing-Structural Algorithmic Programming*, Higher Education Press, Beijing, China, 3rd edition, 2011.
- [18] X. Pan, *Research on The Electromagnetic Scattering Calculation of Typical Targets in Time Domain and The Fast Algorithm*, Harbin Institute of Technology, Harbin, China, 2014.

Research Article

Robust and Privacy-Preserving Service Recommendation over Sparse Data in Education

Xuening Chen,¹ Hanwen Liu,² Yanwei Xu,³ and Chao Yan ²

¹Student Affairs Office, Qufu Normal University, China

²School of Information Science and Engineering, Qufu Normal University, China

³School of Software, Tianjin University, China

Correspondence should be addressed to Chao Yan; yanchao@qfnu.edu.cn

Received 20 November 2018; Revised 28 December 2018; Accepted 28 May 2019; Published 20 June 2019

Guest Editor: Qingchen Zhang

Copyright © 2019 Xuening Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Service recommendation has become one of the most effective approaches to quickly extract insightful information from big educational data. However, the sparsity of educational service quality data (from multiple platforms or parties) used to make service recommendations often leads to few even null recommended results. Moreover, to protect sensitive business information and obey laws, preserving user privacy during the abovementioned multisource data integration process is a very important but challenging requirement. Considering the above challenges, this paper integrates Locality-Sensitive Hashing (LSH) with hybrid Collaborative Filtering (HCF) techniques for robust and privacy-aware data sharing between different platforms involved in the cross-platform service recommendation process. Furthermore, to minimize the “False negative” recommended results incurred by LSH and enhance the success of recommended results, we propose two optimization strategies to reduce the probability that similar neighbours of a target user or similar services of a target service are overlooked by mistake. Finally, we conduct a set of experiments based on a real distributed service quality dataset, i.e., *WS-DREAM*, to validate the feasibility and advantages of our proposed recommendation approach. The extensive experimental results show that our proposal performs better than three competitive methods in terms of efficiency, accuracy, and successful rate while guaranteeing privacy-preservation.

1. Introduction

With the advent of the Web of Things (WoT), tremendous computing resources or services (e.g., web APIs) are emerging rapidly on the Web [1–4], imposing a heavy burden on the service selection decisions of target users in education domain. In this situation, various lightweight service recommendation techniques, e.g., Collaborative Filtering (CF), are proposed to alleviate the abovementioned service selection burdens. Typically, by analysing the historical service usage data (e.g., the quality data of services invoked by users), a recommender system can capture the personalized preferences of a user and output appropriate services to him/her; this way, complex requirements from the user could be satisfied [5–7].

However, in the big data environment, the recommendation bases for educational decisions are sometimes not centralized but are distributed across multiple platforms

[8, 9]. Considering the example in Figure 1, user u_1 invoked web service ws_1 from platform P_1 and user u_2 invoked web service ws_2 from platform P_2 . Thus previous service quality values of ws_1 and ws_2 are recorded in platforms P_1 and P_2 , respectively. In this situation, to make comprehensive and accurate service recommendations to the target user, it is necessary for the recommender system to integrate or fuse the distributed educational data across platforms P_1 and P_2 properly.

However, there are still several challenges in the abovementioned data integration process. First, to protect sensitive business information [10, 11] and obey laws, platform P_1 is often reluctant to share its data with P_2 and vice versa [12]. Such a cross-platform data sharing failure severely impedes the subsequent service recommendations. Besides, the possible sparsity of service quality data [13, 14] stored in platforms P_1 and P_2 often leads to few (even null) recommended

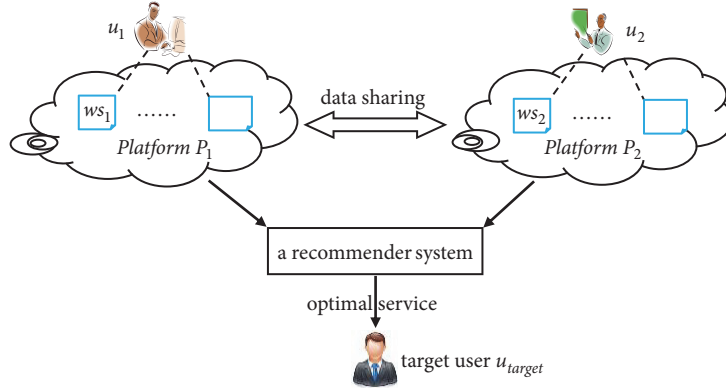


FIGURE 1: Cross-platform service recommendation example.

results, which decreases the target user's satisfaction degree significantly. Namely, the robustness of the recommender system is not as high as expected.

Considering the drawbacks, a time-efficient and privacy-preserving neighbour search technique, Locality-Sensitive Hashing (LSH), is employed for cross-platform service recommendations, so that the multiple platforms involved in the distributed recommendation process can share their data with each other efficiently and securely. Furthermore, we combine the *LSH* technique with hybrid CF (i.e., *HCF*, including user-based CF and item-based CF), to propose a novel privacy-preserving cross-platform service recommendation approach, named $SerRec_{LSH+HCF}$. Benefiting from the advantages of LSH in terms of search efficiency and privacy-preservation, our proposal can achieve a good trade-off among recommendation efficiency, accuracy, successful rate, and privacy-preservation.

In summary, our contributions are three-fold.

(1) We integrate the LSH technique with hybrid CF to guarantee efficient and secure data sharing between different platforms involved in the cross-platform service recommendations in a big educational data environment.

(2) Two solutions are suggested to reduce the probability of "False negative" (i.e., high-quality recommended results are overlooked by mistake) incurred by the inherent shortcoming of LSH and thereby increase the success ratio of recommended list.

(3) A wide range of experiments are conducted on a real distributed service quality dataset, i.e., *WS-DREAM* to validate the feasibility of our proposal. Experiment results show that our proposed $SerRec_{LSH+HCF}$ approach outperforms the other state-of-the-art approaches.

The remainder of this paper is structured as follows. Related work is presented in Section 2. In Section 3, we introduce the preliminary knowledge of the LSH technique to be used in our approach. In Section 4, we introduce the details of our proposed privacy-preserving cross-platform service recommendation approach, i.e., $SerRec_{LSH+HCF}$. In Section 5, a set of experiments are conducted on *WS-DREAM* dataset to validate the feasibility of our proposal. Finally, in Section 6, we conclude the paper and discuss the future research directions.

2. Related Work

To the best of our knowledge, the existing privacy-preservation techniques adopted in the field of service recommendations can be divided into the following four categories: *K-anonymity*, *data obfuscation*, *data decomposition*, and *Locality-Sensitive Hashing*. Next, we introduce the related work from these four perspectives, respectively.

2.1. *K-Anonymity*. As an effective privacy-preservation technique, *K-anonymity* is successfully applied in [15] to protect the sensitive data of users. The authors in [16] recruit *K-anonymity* technique to generalize the location information that users left in the past so as to protect the users' location privacy when making a recommendation decision. Generally, a larger *K* value often means better privacy-preservation performance. However, when the *K* value becomes larger, the availability of anonymous data would be reduced significantly, thereby decreasing the accuracy of recommended results.

2.2. *Data Obfuscation*. Random data obfuscation technique is proposed in various applications where the real service quality data are replaced by the obfuscated data so that the private information hidden in real service quality data can be protected. However, as the data used to make recommendation decisions have already been obfuscated beforehand, the service recommendation accuracy is reduced accordingly. Differential Privacy (DP) technique is recruited in [17] to obfuscate the sensitive service quality data by noise injection so as to hide the real service quality data when making service recommendation decisions. However, the time complexity of Differential Privacy technique is often high. Besides, when the service quality data for recommendation decisions are updated frequently, the accumulated noise amount will become increasingly larger; in this situation, the data availability is reduced, which influences the accuracy of returned results to some extent.

2.3. *Data Decomposition*. In [18], the authors propose a data decomposition mechanism to achieve the privacy-preservation goal in service recommendations. Concretely,

each sensitive quality data is transformed to be multiple segments with less privacy information; afterwards, these service quality segments with little privacy are sent to different user clients for storage. Thus when a user requests service recommendations, the multiple service quality segments kept by each user client are integrated together for subsequent recommendation decision-making process. As each user only possesses multiple service quality segments from different quality data, instead of the whole service quality data, the sensitive information from users is secured, while this approach still fails to secure certain privacy, for example, *the intersection of services executed by different users*.

2.4. Locality-Sensitive Hashing (LSH). As an effective technique for quick neighbour search from massive and high-dimensional data, LSH has recently been introduced into service recommendation for privacy-preservation. In our previous work [19, 20], the LSH technique is combined with user-based CF to protect the sensitive service quality data engaged in recommendation process. In [21], LSH is recruited to build service indices in the distributed environment, so as to reduce the cross-platform data communication cost and improve the recommendation efficiency. However, these LSH-based service recommendation approaches do not consider the low successful rate incurred by the possible sparsity of recommendation data. Moreover, they seldom study the “False negative” recommended results as well as the corresponding resolutions.

With the above analyses, we can conclude that existing privacy-preserving service recommendation approaches either fall short in the efficiency and the capability of privacy-preservation, or they probably overlook high-quality recommended results so that the users’ satisfaction degree is decreased. Considering these drawbacks, we integrate the LSH technique and hybrid CF in this paper to propose a novel privacy-preserving service recommendation approach named $SerRec_{LSH+HCF}$. The details of our proposal will be introduced in Section 4.

3. Preliminary Knowledge

In Section 3.1, we first formulate the privacy-preserving service recommendation problems to be addressed in this paper. Afterwards, in Section 3.2, we briefly introduce the rationale of the LSH technique to be used in our service recommendation approach.

3.1. Problem Formulation. To facilitate the following discussions, we introduce the symbols used in this paper below. $U = \{u_1, \dots, u_m\}$ and $WS = \{ws_1, \dots, ws_n\}$ mean user set and service set, respectively; u_{target} and ws_{target} denote a target user and a target service (i.e., a service preferred by the target user), respectively; q is a quality dimension of web services, e.g., *response time* or *throughput* (for simplicity, only one quality dimension is considered in this paper); $q_{i,j}$ denotes the quality of q of service ws_j ($\in WS$) ever-invoked by user u_i ($\in U$) and the $q_{i,j}$ data are often distributed across different platforms in the big data environment.

With the above formulation, our focused privacy-preserving service recommendation problems can be specified more formally as follows: recommend appropriate services from set WS to target user u_{target} based on the historical $q_{i,j}$ data across different platforms and meanwhile protect the real value of $q_{i,j}$ so that the users’ private information hidden in $q_{i,j}$ data is still secure.

3.2. Locality-Sensitive Hashing. Locality-Sensitive Hashing has been considered as one of the most effective techniques for similar neighbour search due to the following two properties [22]. Here, A and B are two points in original data space, and $h(\cdot)$ denotes a LSH function that is responsible for transforming points A and B into corresponding hash values $h(A)$ and $h(B)$, respectively.

Property 1. If A and B are close in original data space, then they will be projected into the same bucket (i.e., $h(A) = h(B)$) after hashing with high probability.

Property 2. If A and B are not close in original data space, then they will be projected into different buckets (i.e., $h(A) \neq h(B)$) after hashing with high probability.

Thus, inspired by these two properties, we can utilize the hash values $h(A)$ and $h(B)$ (with little or no privacy) to evaluate the approximation degree of original points A and B , without revealing the details of A and B . This way, the private information of points A and B can be protected.

4. $SerRec_{LSH+HCF}$: Service Recommendation Based on LSH and Hybrid CF

In this section, we introduce our proposed privacy-preserving service recommendation approach, i.e., $SerRec_{LSH+HCF}$. Concretely, in Section 4.1, we utilize the LSH technique and user-based CF to make service quality prediction; in Section 4.2, we utilize the LSH technique and item-based CF to make service quality prediction. Finally, in Section 4.3, we integrate the predicted results of Sections 4.1 and 4.2 and then make service recommendations accordingly.

4.1. Service Quality Prediction Based on LSH and User-Based CF. In this subsection, we utilize user-based CF and LSH to look for a target user’s similar neighbours (denoted by set $Neighbour_set(u_{target})$) in a privacy-aware and scalable manner, and then the method makes service quality prediction based on the derived similar neighbours in $Neighbour_set(u_{target})$.

First, for any $u \in U$, the quality data over dimension q are simply converted into an n -dimensional quality vector $\vec{u} = (q_{u,1}, \dots, q_{u,n})$. Here, $q_{u,j}$ denotes the quality value of q of ws_j invoked by user u (typically, $q_{u,j} = 0$ if user u did not rate ws_j in the past) and n is the number of candidate web services. Next, we introduce how to utilize the LSH technique to transform vector \vec{u} with much private information into corresponding user index $h(u)$ with little privacy, based on a pre-selected LSH function $h(\cdot)$.

Concretely, the concrete forms of LSH function $h(\cdot)$ heavily rely on the “distance” for user similarity measurement; in other words, different types of similarity “distance” correspond to different kinds of LSH functions. The Pearson Correlation Coefficient (PCC) is often utilized to calculate user similarity in existing recommender systems, so we choose the LSH function corresponding to the PCC distance in this paper. More concretely, the LSH function $h(\cdot)$ in (1) is adopted [23]. Here, \vec{v} is an n -dimensional vector (v_1, \dots, v_n) , where v_j ($1 \leq j \leq n$) is a random value in the range $[-1, 1]$; symbol “ \circ ” represents the dot product between two vectors. This way, through (1), we can transform \vec{u} with much privacy into a Boolean value $h(u)$ with little privacy.

$$h(u) = \begin{cases} 1 & \text{if } \vec{u} \circ \vec{v} > 0 \\ 0 & \text{if } \vec{u} \circ \vec{v} \leq 0 \end{cases} \quad (1)$$

As LSH is essentially a probability-based similar neighbour search technique, one hash function $h(\cdot)$ is often not enough for finding the similar neighbours of a target user accurately. In view of this observation, we amplify the performance of LSH by adopting r hash functions $\{h_1(\cdot), \dots, h_r(\cdot)\}$ and L hash tables $\{Table_1, \dots, Table_L\}$ into the similar neighbour search processes. Concretely, in each hash table, we can build an index for user u , denoted by $H(u) = (h_1(u), \dots, h_r(u))$. Furthermore, two users u_1 and u_2 are regarded as similar iff condition in (2) holds, where $H_x(u_1)$ and $H_x(u_2)$ denote the indices of u_1 and u_2 in the x -th hash table (i.e., $Table_x$), respectively.

$$\exists x \in \{1, \dots, L\}, \text{ satisfy } H_x(u_1) = H_x(u_2) \quad (2)$$

Likewise, for the target user, i.e., u_{target} , we can calculate his/her user index value $H_x(u_{target})$ in $Table_x$ ($x = 1, \dots, L$), according to the same LSH functions and LSH tables. Then, through the condition in (2), we can determine the similar neighbours of u_{target} and put them into set $Neighbour_set(u_{target})$. The pseudocode of the above neighbour search process is presented in Algorithms 1 and 2, where Algorithm 1 is used to build the L hash tables for users offline and Algorithm 2 is used to search for the similar neighbours of the target user.

However, as LSH is a probability-based neighbour finding technique, the “False negative” search results are inevitable. In other words, some similar neighbours of a target user may be overlooked by mistake according to the abovementioned LSH-based neighbour search process. In view of this drawback, we propose two optimization strategies to reduce the “False negative” probability and improve the successful rate of neighbour search. Next, we introduce these two strategies, respectively.

Strategy 1 (neighbour propagation (for users)). The neighbour relationship between different users is essentially depicted by the similarity of user preferences, while the latter (i.e., user preference similarity) obeys a kind of propagation rule. Let us consider the example in Figure 2 where three users $\{u_1, u_2, u_3\}$ and four web services $\{ws_1, ws_2, ws_3, ws_4\}$

are present. The user-service ratings (1~5*) are shown in Figure 2(a), according to which we can determine the neighbour relationship between u_1 and u_2 as well as the neighbour relationship between u_1 and u_3 . In this situation, we can infer that u_2 and u_3 are possible neighbours (marked with dotted line in Figure 2(b)) as both of them hold the same or similar preferences with u_1 , although u_2 and u_3 are not direct neighbours based on the user-service rating data in Figure 2(a). This way, through the neighbour propagation rule illustrated in Figure 2, we can find more possible neighbours of a target user (in an indirect manner) so as to reduce the “False negative” probability.

Next, we introduce how to integrate the neighbour propagation strategy (for users) into the abovementioned LSH-based neighbouring user search process. Concretely, if users u_a and u_{target} are projected into an identical bucket in any of the L hash tables $\{Table_1, \dots, Table_L\}$ and users u_a and u_b are projected into an identical bucket in any of the L hash tables, then according to the neighbour propagation strategy (for users), we can infer that u_b is a possible neighbour of u_{target} and put u_b into $Neighbour_set(u_{target})$. The pseudocode of Strategy 1 is presented in Algorithm 3.

Strategy 2 (condition relaxation for neighbour search (for users)). According to the inherent characteristic of the LSH technique, the number of hash functions (i.e., r) plays an important role in the neighbour search process. Generally, a larger r value often means stricter filtering condition for neighbour search and thereby leads to higher probability of “False negative” search results. Considering this, we relax the search condition for neighbours of the target user to reduce the “False negative” probability. Next, we introduce the concrete condition relaxation process.

According to the neighbour search condition in (2), u_i is regarded as a neighbour of u_{target} iff $H(u_i) = H(u_{target})$ holds in any hash table, where $H(u_i) = (h_1(u_i), \dots, h_r(u_i))$ and $H(u_{target}) = (h_1(u_{target}), \dots, h_r(u_{target}))$. Namely, all the r bit values in $H(u_i)$ are required to be equal to the r bit values in $H(u_{target})$, respectively. Hence, to relax the search condition for neighbours of u_{target} and guarantee high similarity between u_{target} and his/her neighbours u_i , one bit difference between the indices of u_{target} and u_i is permitted.

For example, if $H_x(u_{target}) = (1, 1, 1)$ holds in hash table $Table_x$, then the neighbour u_i 's index in $Table_x$ is permitted to be $(0, 1, 1)$ or $(1, 0, 1)$ or $(1, 1, 0)$. In other words, any user whose index is equal to $(0, 1, 1)$ or $(1, 0, 1)$ or $(1, 1, 0)$ is a possible neighbour of u_{target} . This is the main idea of our proposed search condition relaxation strategy (for users). The pseudocode of Strategy 2 is presented in Algorithm 4, where $H_x(u_{target})_k$ denotes the relaxed search condition (i.e., k -th bit is different from that of $H_x(u_{target})$) for u_{target} in $Table_x$; for example, if $H_x(u_{target}) = (1, 1, 1)$, then $H_x(u_{target})_2 = (1, 0, 1)$ holds.

Through Strategies 1 and 2, we can obtain an enlarged set of neighbours of the target user, i.e., $Neighbour_set(u_{target})$. Next, we make service quality prediction based on the

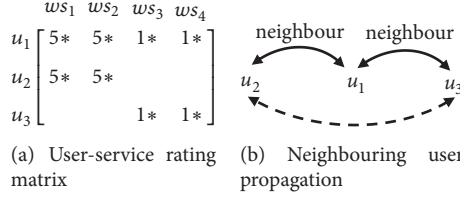


FIGURE 2: Neighbouring user propagation example.

Inputs: $U = \{u_1, \dots, u_m\}$
 $WS = \{ws_1, \dots, ws_n\}$
 L : number of LSH tables
 r : number of LSH functions
Output: $Table_1, \dots, Table_L$
For $x = 1, \dots, L$ **do** // build hash tables $\{Table_1, \dots, Table_L\}$ offline
 For $k = 1, \dots, r$ **do**
 For $i = 1, \dots, m$ **do**
 Build user sub-index $h_k(u_i)$ based on random LSH function $h_k(\cdot)$
 For $i = 1, \dots, m$ **do**
 Build user index $H_x(u_i) = (h_1(u_i), \dots, h_r(u_i))$
 Return hash table $Table_x$ constituted by all the " $u_i \rightarrow H_x(u_i)$ " mappings

ALGORITHM 1: User-hash-table-building (U, WS, L, r).

Inputs: u_{target} // a target user
 $TB = \{Table_1, \dots, Table_L\}$
Output: $Neighbour_set(u_{target})$
For $x = 1, \dots, L$ **do**
 Find the bucket bt corresponding to $H_x(u_{target})$ in $Table_x$
 If $u_i \in bt$ and $u_i \neq u_{target}$
 Then put u_i into $Neighbour_set(u_{target})$
Return $Neighbour_set(u_{target})$

ALGORITHM 2: Neighbouring-user-search (u_{target}, TB).

elements in $Neighbour_set(u_{target})$. Concretely, for web service ws_j never invoked by u_{target} before, its predicted quality over dimension q by u_{target} , denoted by $q_{target,j}$, can be calculated by

$$q_{target,j} = \frac{1}{|Neighbour_set(u_{target})|} \sum_{u_i \in Neighbour_set(u_{target})} q_{i,j} \quad (3)$$

4.2. Service Quality Prediction Based on LSH and Item-Based CF. Similar to Section 4.1, in this subsection, we first utilize item-based CF and LSH techniques to look for the similar services (named "neighbouring services") of target service ws_{target} (denoted by set $Neighbour_set(ws_{target})$) in a privacy-aware and scalable manner, and the techniques then make service quality predictions based on the elements in $Neighbour_set(ws_{target})$.

First, for any web service $ws_j \in WS$, its historical quality data over dimension q ever-invoked by users can be specified

by an m -dimensional vector $\overrightarrow{ws_j} = (q_{1,j}, \dots, q_{m,j})$, where $q_{i,j}$ ($1 \leq i \leq m$) denotes the quality value of q of service ws_j invoked by user u_i and m is the number of users. Next, we utilize the LSH technique to transform $\overrightarrow{ws_j}$ with private information into a corresponding service index $h(ws_j)$ with little privacy, based on the random LSH function $h(\cdot)$ in (1). Here, \vec{v} is an m -dimensional real vector (v_1, \dots, v_m) , where v_i ($1 \leq i \leq m$) is a random value in the range $[-1, 1]$.

This way, we can transform $\overrightarrow{ws_j}$ with much privacy into a Boolean value $h(ws_j)$ with little privacy.

Likewise, we amplify LSH through integrating r hash functions $\{h_1(\cdot), \dots, h_r(\cdot)\}$ and L hash tables $\{Table_1, \dots, Table_L\}$. Then, in each hash table, we build an index for service ws_j , denoted by $H(ws_j) = (h_1(ws_j), \dots, h_r(ws_j))$. Furthermore, two services ws_1 and ws_2 are regarded as neighbouring services if the condition in (4) holds where $H_x(ws_1)$ and $H_x(ws_2)$ denote the indices of services ws_1 and ws_2 in the x -th hash table (i.e., $Table_x$), respectively.

$$\exists x \in \{1, \dots, L\}, \text{ satisfy } H_x(ws_1) = H_x(ws_2) \quad (4)$$

Then, through (4), we can find out the neighbouring services of ws_{target} and put them into $Neighbour_set(ws_{target})$. Note that if multiple target services are present, then it is necessary to repeat the above process for each target service to discover all the qualified neighbours. The pseudocode is presented in Algorithms 5 and 6, where Algorithm 5 is used to build the L hash tables for services offline and Algorithm 6 is used to search for the neighbouring services of the target service (repeat Algorithm 6 if multiple target services are present).


```

Inputs:  $U = \{u_1, \dots, u_m\}$ 
 $u_{target}$  // a target user
 $Neighbour\_set(u_i)$  // before neighbour propagation (for users)
Output:  $Neighbour\_set(u_{target})$  // after neighbour propagation (for users)
For each  $u_a \in Neighbour\_set(u_{target})$  do
  For each  $u_b \in Neighbour\_set(u_a)$  do
    If  $u_b \notin Neighbour\_set(u_{target})$ 
      Then put  $u_b$  into  $Neighbour\_set(u_{target})$ 
Return  $Neighbour\_set(u_{target})$ 

```

ALGORITHM 3: Neighbouring user search based on Strategy 1.

```

Inputs:  $U = \{u_1, \dots, u_m\}$ 
 $u_{target}$  // a target user
 $TB = \{Table_1, \dots, Table_L\}$ 
Output:  $Neighbour\_set(u_{target})$  // after condition relaxation for neighbour search (for users)
For  $x = 1, \dots, L$  do
  For  $k = 1, \dots, r$  do
     $H_x(u_{target}) = H_x(u_{target})_k$ 
    Neighbouring-user-search ( $u_{target}, TB$ ) // Algorithm 2
Return  $Neighbour\_set(u_{target})$ 

```

ALGORITHM 4: Neighbouring user search based on Strategy 2.

However, similar to Section 4.1, “False negative” search results are also inevitable; in other words, certain real neighbours of a target user are probably deemed as non-neighbors. Considering the drawback, Strategies 3 and 4 (actually the variants of Strategies 1 and 2 in Section 4.1) are proposed to reduce the “False negative” probability.

Strategy 3 (neighbour propagation (for services)). Let’s consider the example in Figure 3 where four users $\{u_1, u_2, u_3, u_4\}$ and three web services $\{ws_1, ws_2, ws_3\}$ are present. The user-service ratings ($1 \sim 5$) are shown in Figure 3(a), according to which we can determine that ws_1 and ws_2 are neighbouring services and ws_1 and ws_3 are neighbouring services. In this situation, we can infer that ws_2 and ws_3 are possible neighbouring services (marked with dotted line in Figure 3(b)). Thus through the propagation rule illustrated in Figure 3, we can obtain more neighbouring services of a target service so that the “False negative” probability is reduced.

Next, we introduce how to integrate the neighbour propagation strategy (for services) into the abovementioned LSH-based neighbouring service search process. Concretely, if services ws_a and ws_{target} are projected into an identical bucket in any of the L hash tables $\{Table_1, \dots, Table_L\}$ and services ws_a and ws_b are projected into an identical bucket in any of the L hash tables, then according to the neighbour propagation strategy (for services), we can infer that ws_b is probably a neighbouring service of ws_{target} and hence put ws_b into $Neighbour_set(ws_{target})$. The pseudocode of Strategy 3 is presented in Algorithm 7.

Strategy 4 (condition relaxation for neighbour search (for services)). Similar to Strategy 2, in Strategy 4, we relax the search condition for neighbouring services of the target service to reduce the “False negative” probability of search results. Concretely, according to the neighbouring service search condition in (4), service ws_j is regarded as a neighbouring service of ws_{target} iff all the r bit values in $H(ws_j)$ are equal to the r bit values in $H(ws_{target})$, respectively. Therefore, to relax the search condition for neighbouring services of ws_{target} and meanwhile guarantee the high similarity between ws_{target} and its neighbouring services ws_j , one bit difference between the indices of ws_{target} and ws_j is permitted.

For example, if condition $H_x(ws_{target}) = (1, 1, 1)$ holds in hash table $Table_x$, then any service whose index is equal to $(0, 1, 1)$ or $(1, 0, 1)$ or $(1, 1, 0)$ is a possible neighbouring service of ws_{target} . This is the main idea of our proposed search condition relaxation strategy (for services). The pseudocode of Strategy 4 is presented in Algorithm 8, where $H_x(ws_{target})_k$ denotes the relaxed search condition (i.e., the k -th bit is different from that of $H_x(ws_{target})$) for ws_{target} in $Table_x$; e.g., $H_x(ws_{target})_2 = (1, 0, 1)$ holds if $H_x(ws_{target}) = (1, 1, 1)$.

Through Strategies 3 and 4, we can obtain an enlarged set of neighbouring services of the target services, i.e., $Neighbour_set(ws_{target})$. Next, for each service ws_j never invoked by u_{target} , its predicted quality over dimension q rated by u_{target} , denoted by $q_{target,j}$, is calculated by equation (5) where $ws_j \in Neighbour_set(ws_{target})$. Here, $q_{target,target}$ denotes the real service quality of ws_{target} observed by u_{target} . Furthermore, if service ws_j appears multiple times in

```

Inputs:  $U = \{u_1, \dots, u_m\}$ 
           $WS = \{ws_1, \dots, ws_n\}$ 
           $L$ : number of LSH tables
           $r$ : number of LSH functions
Output:  $Table_1, \dots, Table_L$ 
For  $x = 1, \dots, L$  do // build hash tables  $\{Table_1, \dots, Table_L\}$  offline
  For  $k = 1, \dots, r$  do
    For  $j = 1, \dots, n$  do
      Build service sub-index  $h_k(ws_j)$  based on random LSH function  $h_k(\cdot)$ 
    For  $j = 1, \dots, n$  do
      Build service index  $H_x(ws_j) = (h_1(ws_j), \dots, h_r(ws_j))$ 
  Return hash table  $Table_x$  constituted by all " $ws_j \rightarrow H_x(ws_j)$ " mappings

```

ALGORITHM 5: Service-hash-table-building (U, WS, L, r).

```

Inputs:  $ws_{target}$  // a target service
           $TB = \{Table_1, \dots, Table_L\}$ 
Output:  $Neighbour\_set(ws_{target})$ 
For  $x = 1, \dots, L$  do
  Find the bucket  $bt$  corresponding to  $H_x(ws_{target})$  in  $Table_x$ 
  If  $ws_j \in bt$  and  $ws_j \neq ws_{target}$ 
  Then put  $ws_j$  into  $Neighbour\_set(ws_{target})$ 
Return  $Neighbour\_set(ws_{target})$ 

```

ALGORITHM 6: Neighbouring-service-search (ws_{target}, TB).

$Neighbour_set(ws_{target})$, then the average predicted quality is adopted.

$$q_{target,j} = q_{target,target} \quad (5)$$

4.3. Aggregation of Predicted Service Quality and Service Recommendation. We aggregate the two pieces of quality data predicted by (3) and (5) into a comprehensive quality in (6). Here, q_{user} and q_{item} denote the $q_{target,j}$ values predicted in (3) and (5), respectively; α and β ($0 \leq \alpha, \beta \leq 1$ and $\alpha + \beta = 1$) are the aggregation coefficients. At last, we choose the service ws_j whose predicted value (i.e., $q_{target,j}$ in (6)) is the best and return it to u_{target} .

$$q_{target,j} = \alpha * q_{user} + \beta * q_{item} \quad (6)$$

5. Experiments

In this section, we deploy a group of experiments to validate the feasibility of our proposed $SerRec_{LSH+HCF}$ approach in terms of service recommendation efficiency, accuracy, and successful rate. Concretely, in Section 5.1, we introduce the experiment dataset and configurations that we adopted for experiments; in Section 5.2, experiment comparison results are presented; in Section 5.3, further discussions are given.

5.1. Experiment Configurations. Our experiments are based on a real distributed web service quality dataset *WS-DREAM*

[24] that collects real-world service quality data from 339 users on 5825 web services (hosted in different countries). Each country that hosts a group of services is considered to be an individual platform for recommendation scenario simulation. Additionally, partial real values in the dataset are dropped for prediction needs. Moreover, only one quality dimension of services, i.e., *response time*, is considered in our experiments for simplicity. The target user is selected randomly from the user set in *WS-DREAM*, whose invoked services are regarded as the target services recruited in Section 4.2.

In order to validate the feasibility of our proposed $SerRec_{LSH+HCF}$ approach, we test the time cost and *MAE* of our proposal and compare them with three other state-of-the-art recommendation approaches including *UPCC* [25], *P-UIPCC* [17], and *PPICF* [18]. Concretely, *UPCC* is the benchmark service recommendation approach that is based on user-based CF; *P-UIPCC* utilizes the “divide-merge” operations over sensitive service quality data; while in *PPICF*, the real service quality data is transformed into the obfuscated data and then the obfuscated data are used to make service quality prediction and service recommendations.

The experiments were conducted on a Dell laptop with 2.80 GHz processors and 2.0 GB RAM. The machine runs Windows XP and JAVA 1.5. Each experiment was carried out ten times, and the average experimental results were adopted finally.

5.2. Experiment Results and Analyses. In the experiments, five profiles are tested and compared to validate the feasibility of our proposal. Here, ω denotes the density of the user-service quality matrix recruited to make service recommendations; L and r denote the number of hash tables and the number of hash functions, respectively; $\alpha = \beta = 0.5$ holds in (6).

Profile 1 (computational time of four approaches w.r.t. ω). Next, we measure the service computational time for recommendation process and scalability of four approaches with respect to matrix density ω . Concrete experimental parameters are set as follows: ω is varied from 5% to 25%, $L = 10$ and $r = 14$ hold. Experimental results are shown in Figure 4.

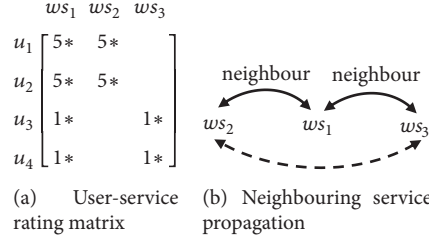


FIGURE 3: Neighbouring service propagation example.

```

Inputs:  $WS = \{ws_1, \dots, ws_n\}$ 
           $ws_{target}$  // a target service
           $Neighbour\_set(ws_i)$  // before neighbour propagation (for services)
Output:  $Neighbour\_set(ws_{target})$  // after neighbour propagation (for services)
For each  $ws_a \in Neighbour\_set(ws_{target})$  do
  For each  $ws_b \in Neighbour\_set(ws_a)$  do
    If  $ws_b \notin Neighbour\_set(ws_{target})$ 
      Then put  $ws_b$  into  $Neighbour\_set(ws_{target})$ 
Return  $Neighbour\_set(ws_{target})$ 

```

ALGORITHM 7: Neighbouring service search based on Strategy 3.

As the experimental results in Figure 4 indicate, the computational time of the four different approaches all increase with the growth of service quality matrix density, i.e., ω , because all the user-service quality data need to be considered in the four approaches and, therefore, more computational time is often required when the quality matrix becomes denser (i.e., when ω grows). However, our proposed $SerRec_{LSH+HCF}$ approach outperforms the other three approaches in terms of recommendation efficiency and scalability because most jobs in our approach (e.g., user indices building) can be done offline before a service recommendation request arrives, while the time complexity of the remaining jobs (e.g., online neighbour search) is rather small. So generally, our proposal can satisfy the quick response requirements of target users.

Profile 2 (accuracy of returned results by four approaches w.r.t. ω). We test and compare the recommendation accuracy (i.e., MAE , the smaller the better) of four approaches. The following are the experiment parameter settings: ω is varied from 5% to 25%, $L = 10$, and $r = 14$. Concrete comparison results are presented in Figure 5.

Figure 5 indicates that the accuracy of returned results by $P-UIPCC$ and $PPICF$ are not high; the reason is that, in order to secure the sensitive user privacy, the service quality data engaged in recommendation process have already been obfuscated in $UIPCC$ and $PPICF$, while our $SerRec_{LSH+HCF}$ approach performs better than the other three approaches in terms of recommendation accuracy; this is because only the “most similar” neighbouring users and neighbouring services can be returned by LSH and recruited to make service recommendations. Therefore, the recommendation accuracy is improved considerably.

Profile 3 (recommendation efficiency of $SerRec_{LSH+HCF}$ w.r.t. L and r). In this profile, we test the recommendation efficiency of our $SerRec_{LSH+HCF}$ approach with respect to L and r . The parameters are set as follows: $\omega = 25\%$, L is varied from 6 to 14, and r is varied from 8 to 14. Experimental results are shown in Figure 6.

As shown in Figure 6(a), the time cost of our proposal increases approximately with the growth of L , as all the L hash tables need to be traversed in order to find the similar neighbours of the target user by (2) and find the similar neighbouring services of the target services by (4), respectively, while Figure 6(b) shows that the time cost decreases when r grows. This is because a larger r value often means stricter search condition for neighbouring users or neighbouring services; and therefore, few search results are obtained when r is large; in this situation, less time is needed to evaluate and rank the few search results.

Profile 4 (recommendation accuracy of $SerRec_{LSH+HCF}$ w.r.t. L and r). We test the recommendation accuracy of our proposed $SerRec_{LSH+HCF}$ approach with respect to L and r . The following are the experimental parameter settings: $\omega = 25\%$, L is varied from 6 to 14, and r is varied from 8 to 14. The experiment results are offered in Figure 7.

As Figure 7 shows, the recommendation accuracy of $SerRec_{LSH+HCF}$ increases (i.e., MAE drops) with the decrease of L and the growth of r . This is because a smaller L value or a larger r value often means stricter search condition for neighbouring users and services; in this situation, only the “most similar” neighbouring users or neighbouring services are returned to make service recommendations. Therefore, the recommendation accuracy is improved accordingly.

```

Inputs:  $WS = \{ws_1, \dots, ws_n\}$ 
           $ws_{target}$  //a target service
           $TB = \{Table_1, \dots, Table_L\}$ 
Output:  $Neighbour\_set(ws_{target})$  //after condition relaxation for neighbour search (for services)
For  $x = 1, \dots, L$  do
  For  $k = 1, \dots, r$  do
     $H_x(ws_{target}) = H_x(ws_{target})_k$ 
    Neighbouring-service-search ( $ws_{target}, TB$ ) //Algorithm 6
Return  $Neighbour\_set(ws_{target})$ 

```

ALGORITHM 8: Neighbouring service search based on Strategy 4.

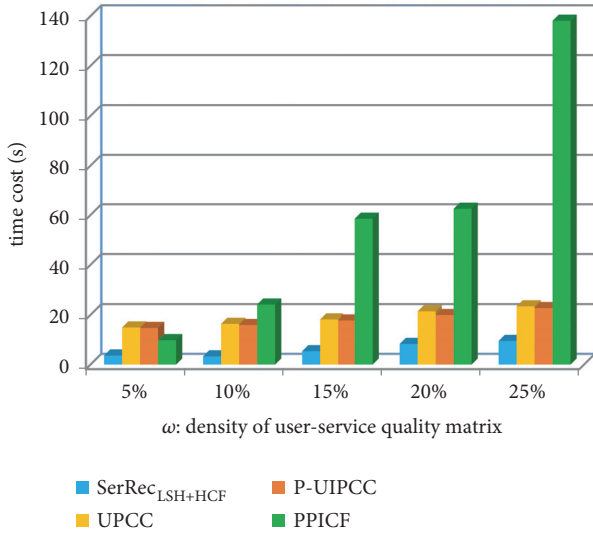


FIGURE 4: Computational time of different approaches.

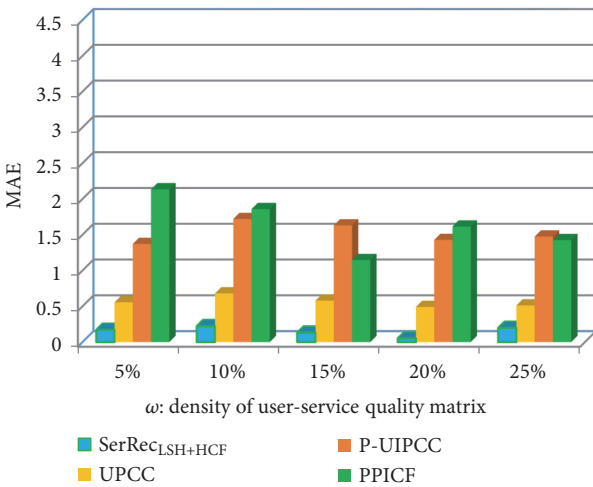


FIGURE 5: Recommendation accuracy of four approaches.

Profile 5 (recommendation successful rate comparison). LSH is essentially a probability-based similar neighbour search technique; therefore, our proposed LSH-based service recommendation approach $SerRec_{LSH+HCF}$ cannot always guarantee to return a satisfying recommended result to the target

user. In other words, recommendation failure is inevitable. However, as discussed in Section 2, the hybrid CF method can reduce the failure rate to some extent. Therefore, in this profile, we test the recommendation successful rate of our proposal and compare it with the following two benchmark approaches: $SerRec_{LSH+UCF}$ (i.e., the $DistSR_{LSH}$ approach in [26]) and $SerRec_{LSH+ICF}$.

- (1) $SerRec_{LSH+UCF}$: integrate LSH with user-based CF
- (2) $SerRec_{LSH+ICF}$: integrate LSH with item-based CF

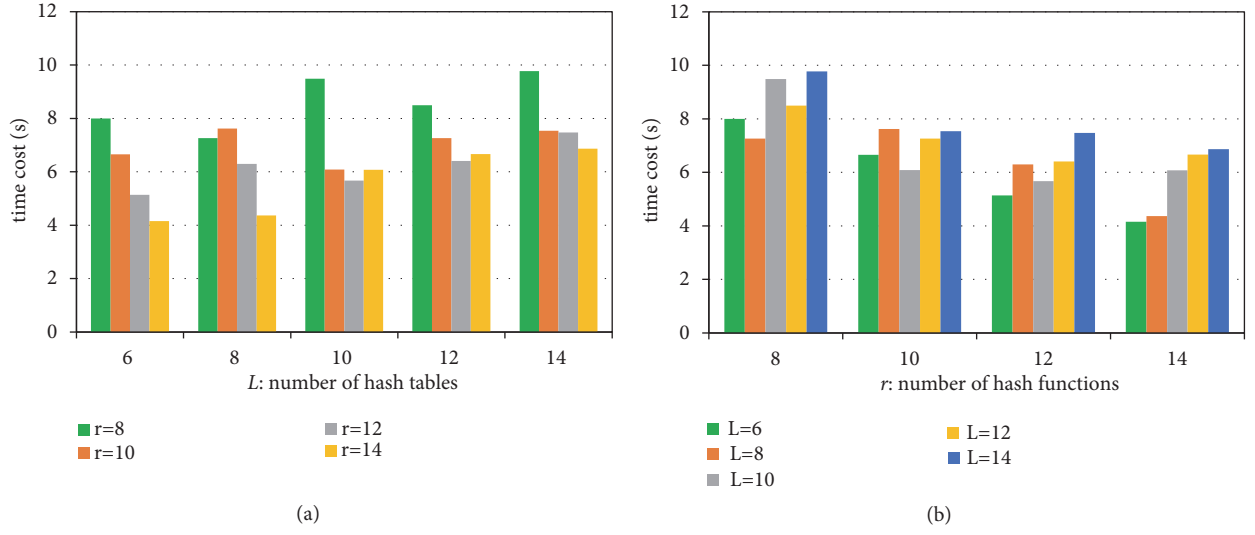
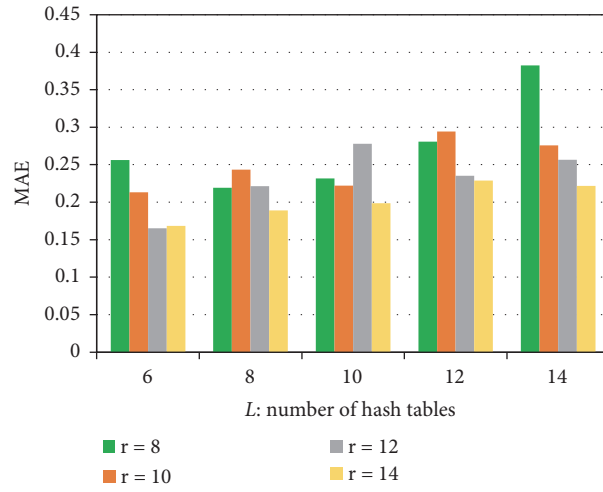
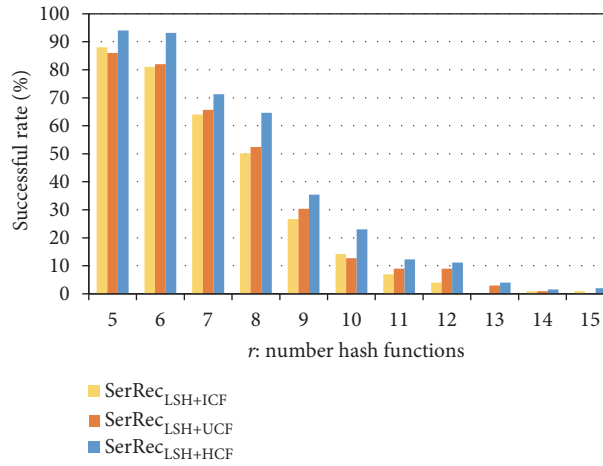
Here, we define the successful rate of a recommendation approach as the ratio between the successful recommendation times and the total recommendation times ($\in [0, 100\%]$). Parameters settings are $\omega = 25\%$, $L = 1$, and r is varied from 8 to 14. Concrete experimental results are presented in Figure 8.

As Figure 8 shows, the successful rates of three recommendation approaches all decrease with the growth of r . This is because a larger r value often means stricter filtering condition for the search of neighbouring users and neighbouring services; and, therefore, the successful rate of recommendations is reduced accordingly. Namely, there is a trade-off between successful rate and r ; specifically, when r is large enough (e.g., when $r = 14, 15, \dots$), the successful rate approaches 0. However, as Figure 6 shows, our approach still outperforms the other two approaches in terms of successful rate as our approach recruits hybrid CF for recommendation, integrating the advantages of both user-based CF and item-based CF.

5.3. Further Discussions. Our experiments only adopt one service quality dimension, i.e., *response time*, without considering the probably existed multiple dimensions [27–37] and their respective weight significance values [38–44]. In the future research, we will integrate the dimension and weight information into $SerRec_{LSH+HCF}$ to make the approach more comprehensive. Besides, only one type of service quality data is considered in the experiments. So in the future, we will further extend our proposal by considering the possible data diversity in the big data environment [45–50].

6. Conclusions

Collaborative service recommendation has become an effective technique to quickly extract insightful information from

FIGURE 6: Efficiency of $SerRec_{LSH+HCF}$ w.r.t. L and r .FIGURE 7: Accuracy of $SerRec_{LSH+HCF}$ w.r.t. L and r .FIGURE 8: Successful rate of three approaches (w.r.t. r).

big educational data. However, traditional service recommendation approaches often assume that the service usage data used to make recommendations are centralized, without considering the multisource property of service usage data as well as the privacy leakage risks during the multisource educational data integration. Besides, existing service recommendation approaches often suffer from low robustness due to the possible data sparsity. In view of these drawbacks, we combine the LSH technique and hybrid Collaborative Filtering (HCF) for distributed service recommendations in the big data environment. Furthermore, to minimize the “False negative” recommended results incurred by the inherent shortcoming of LSH, two solutions are introduced in this paper, to reduce the probability that similar users and similar services are overlooked by mistake and thereby enhance the success rate. A wide range of experiments deployed on real-world dataset shows the performances of $SerRec_{LSH+HCF}$ in terms of efficiency, accuracy, and successful rate while securing the sensitive user information.

However, only one quality dimension of web services is considered in the recommendation model, which is often not enough for the practical recommendation requirements. In the future, we will further refine our work by considering multiple quality dimensions as well as their linear correlations [51–53] and nonlinear correlations [54–58]. Besides, data type diversity is another challenge in the big data environment. Therefore, in the future research, we will continue to extend our proposal by integrating the multisource data with diverse data types, e.g., discrete data [59–63], binary data [64], and fuzzy data [65–67].

Data Availability

The [web service quality] data used to support the findings of this study have been deposited in the [WS-DREAM] repository (<http://inpluslab.com/wsdream/>)

Conflicts of Interest

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

Acknowledgments

This paper is partially supported by the Natural Science Foundation of China (No. 61872219).

References

- [1] X. Wang, L. T. Yang, X. Xie, J. Jin, and M. Jamal Deen, “A cloud-edge computing framework for cyber-physical-social services,” *IEEE Communications Magazine*, vol. 55, no. 11, pp. 80–85, 2017.
- [2] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, “A double deep q-learning model for energy-efficient edge scheduling,” *IEEE Transactions on Services Computing*, 2018.
- [3] L. Qi, P. Dai, J. Yu, Z. Zhou, and Y. Xu, “Time-location-frequency-aware internet of things service selection based on historical records,” *International Journal of Distributed Sensor Networks*, vol. 13, no. 1, pp. 1–9, 2017.
- [4] Q. Zhang, L. T. Yang, Z. Chen, P. Li, and F. Bu, “An adaptive dropout deep computation model for industrial IoT big data learning with crowdsourcing to cloud computing,” *IEEE Transactions on Industrial Informatics*, 2018.
- [5] X. Wang, W. Wang, L. T. Yang, S. Liao, D. Yin, and M. J. Deen, “A distributed HOSVD method with its incremental computation for big data in cyber-physical-social systems,” *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 481–492, 2018.
- [6] K. Dou, B. Guo, and L. Kuang, “A privacy-preserving multimedia recommendation in the context of social network based on weighted noise injection,” *Multimedia Tools and Applications*, pp. 1–20, 2017.
- [7] L. Qi, W. Dou, and J. Chen, “Weighted principal component analysis-based service selection method for multimedia services in cloud,” *Computing*, vol. 98, no. 1, pp. 195–214, 2016.
- [8] L. T. Yang, X. Wang, X. Chen et al., “A multi-order distributed HOSVD with its incremental computing for big services in cyber-physical-social systems,” *IEEE Transactions on Big Data*, 2018.
- [9] X. Wang, L. T. Yang, H. Liu, and M. J. Deen, “A big data-as-a-service framework: state-of-the-art and perspectives,” *IEEE Transactions on Big Data*, vol. 4, no. 3, pp. 325–340, 2018.
- [10] Q. Zhang, L. T. Yang, A. Castiglione, Z. Chen, and P. Li, “Secure weighted possibilistic c-means algorithm on cloud for clustering big data,” *Information Sciences*, vol. 479, pp. 515–525, 2018.
- [11] S. Zhang, G. Wang, M. Z. Bhuiyan, and Q. Liu, “A dual privacy preserving scheme in continuous location-based services,” *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4191–4200, 2018.
- [12] 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.
- [13] L. Qi, X. Xu, W. Dou et al., “Time-aware IoE service recommendation on sparse data,” *Mobile Information Systems*, vol. 2016, Article ID 4397061, 12 pages, 2016.
- [14] L. Qi, Z. Zhou, J. Yu, and Q. Liu, “Data-sparsity tolerant web service recommendation approach based on improved collaborative filtering,” *IEICE Transaction on Information and Systems*, vol. E100D, no. 9, pp. 2092–2099, 2017.
- [15] S. Zhang, X. Li, Z. Tan, T. Peng, and G. Wang, “A caching and spatial K-anonymity driven privacy enhancement scheme in continuous location-based services,” *Future Generation Computer Systems*, vol. 94, pp. 40–50, 2019.
- [16] S. Zhang, X. Mao, K. R. Choo, T. Peng, and G. Wang, “A trajectory privacy-preserving scheme based on a dual-K mechanism for continuous location-based services,” *Information Sciences*, 2019.
- [17] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, “A privacy-preserving qos prediction framework for web service recommendation,” in *Proceedings of the IEEE International Conference on Web Services (ICWS ’15)*, pp. 241–248, New York, NY, USA, July 2015.
- [18] D. Li, C. Chen, Q. Lv et al., “An algorithm for efficient privacy-preserving item-based collaborative filtering,” *Future Generation Computer Systems*, vol. 55, pp. 311–320, 2016.
- [19] 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.

- [20] 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.
- [21] C. Yan, X. Cui, L. Qi, X. Xu, and X. Zhang, "Privacy-aware data publishing and integration for collaborative service recommendation," *IEEE Access*, vol. 6, pp. 43021–43028, 2018.
- [22] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," *The VLDB Journal*, vol. 99, no. 6, pp. 518–529, 1999.
- [23] *Data Mining and Query Log Analysis for Scalable Temporal and Continuous Query Answering*, 2015, <http://www.optique-project.eu/>.
- [24] Z. Zheng, Y. Zhang, and M. R. Lyu, "Investigating QoS of real-world web services," *IEEE Transactions on Services Computing*, vol. 7, no. 1, pp. 32–39, 2014.
- [25] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pp. 43–52, 1998.
- [26] L. Qi, H. Xiang, W. Dou, C. Yang, Y. Qin, and X. Zhang, "Privacy-preserving distributed service recommendation based on locality-sensitive hashing," in *Proceedings of the 24th IEEE International Conference on Web Services, ICWS 2017*, pp. 49–56, USA, June 2017.
- [27] X. Wang, L. T. Yang, L. Kuang, X. Liu, Q. Zhang, and M. J. Deen, "A tensor-based big-data-driven routing recommendation approach for heterogeneous networks," *IEEE Network Magazine*, vol. 33, no. 1, pp. 64–69, 2019.
- [28] M. Wang and G.-L. Tian, "Robust group non-convex estimations for high-dimensional partially linear models," *Journal of Nonparametric Statistics*, vol. 28, no. 1, pp. 49–67, 2016.
- [29] X. Wang and M. Wang, "Variable selection for high-dimensional generalized linear models with the weighted elastic-net procedure," *Journal of Applied Statistics*, vol. 43, no. 5, pp. 796–809, 2016.
- [30] P. Wang and L. Zhao, "Some geometrical properties of convex level sets of minimal graph on 2-dimensional Riemannian manifolds," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 130, pp. 1–17, 2016.
- [31] X. Wang and M. Wang, "Adaptive group bridge estimation for high-dimensional partially linear models," *Journal of Inequalities and Applications*, vol. 2017, article no. 158, pp. 1–18, 2017.
- [32] X. Wang, S. Zhao, and M. Wang, "Restricted profile estimation for partially linear models with large-dimensional covariates," *Statistics & Probability Letters*, vol. 128, pp. 71–76, 2017.
- [33] H. Tian and M. Han, "Bifurcation of periodic orbits by perturbing high-dimensional piecewise smooth integrable systems," *Journal of Differential Equations*, vol. 263, no. 11, pp. 7448–7474, 2017.
- [34] P. Wang and X. Wang, "The geometric properties of harmonic function on 2-dimensional Riemannian manifolds," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 103, pp. 2–8, 2014.
- [35] M. Wang and X. Wang, "Adaptive Lasso estimators for ultrahigh dimensional generalized linear models," *Statistics & Probability Letters*, vol. 89, no. 1, pp. 41–50, 2014.
- [36] X. Wang, M. Wang, and X. Wang, "A note on the one-step estimator for ultrahigh dimensionality," *Journal of Computational and Applied Mathematics*, vol. 260, pp. 91–98, 2014.
- [37] G.-L. Tian, M. Wang, and L. Song, "Variable selection in the high-dimensional continuous generalized linear model with current status data," *Journal of Applied Statistics*, vol. 41, no. 3, pp. 467–483, 2014.
- [38] S. Yang, Z.-A. Yao, and C.-A. Zhao, "The weight distributions of two classes of p-ary cyclic codes with few weights," *Finite Fields and Their Applications*, vol. 44, pp. 76–91, 2017.
- [39] Y.-F. Wang, C.-C. Yin, and X.-S. Zhang, "Uniform estimate for the tail probabilities of randomly weighted sums," *Acta Mathematicae Applicatae Sinica*, vol. 30, no. 4, pp. 1063–1072, 2014.
- [40] S. Yang, Z.-A. Yao, and C.-A. Zhao, "A class of three-weight linear codes and their complete weight enumerators," *Cryptography and Communications*, vol. 9, no. 1, pp. 133–149, 2017.
- [41] J. Cai, "An implicit sigma (3) type condition for heavy cycles in weighted graphs," *Ars Combinatoria*, vol. 115, pp. 211–218, 2014.
- [42] S. Yang and Z.-A. Yao, "Complete weight enumerators of a family of three-weight linear codes," *Designs, Codes and Cryptography*, vol. 82, no. 3, pp. 663–674, 2017.
- [43] S. Yang and Z.-A. Yao, "Complete weight enumerators of a class of linear codes," *Discrete Mathematics*, vol. 340, no. 4, pp. 729–739, 2017.
- [44] S. Yang, X. Kong, and C. Tang, "A construction of linear codes and their complete weight enumerators," *Finite Fields and Their Applications*, vol. 48, pp. 196–226, 2017.
- [45] H. Liu and F. Meng, "Some new generalized Volterra-Fredholm type discrete fractional sum inequalities and their applications," *Journal of Inequalities and Applications*, vol. 2016, no. 1, article no. 213, 2016.
- [46] P. Li and G. Ren, "Some classes of equations of discrete type with harmonic singular operator and convolution," *Applied Mathematics and Computation*, vol. 284, pp. 185–194, 2016.
- [47] P. Li, "Singular integral equations of convolution type with Hilbert kernel and a discrete jump problem," *Advances in Difference Equations*, vol. 2017, no. 1, article no. 360, 2017.
- [48] Y. Bai and L. Liu, "New oscillation criteria for second-order delay differential equations with mixed nonlinearities," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 796256, 9 pages, 2010.
- [49] Y. Wang and C. Yin, "Approximation for the ruin probabilities in a discrete time risk model with dependent risks," *Statistics & Probability Letters*, vol. 80, no. 17–18, pp. 1335–1342, 2010.
- [50] Q. Feng, F. Meng, and Y. Zhang, "Generalized gronwall-bellman-type discrete inequalities and their applications," *Journal of Inequalities and Applications*, vol. 2011, article no. 47, 2011.
- [51] G. Guo, W. Shao, L. Lin, and X. Zhu, "Parallel tempering for dynamic generalized linear models," *Communications in Statistics—Theory and Methods*, vol. 45, no. 21, pp. 6299–6310, 2016.
- [52] L. L. Liu and Y. Li, "Recurrence relations for linear transformations preserving the strong q-log-convexity," *The Electronic Journal of Combinatorics*, vol. 23, no. 3, pp. 1–11, 2016.
- [53] H. Li and S. Wang, "Partial condition number for the equality constrained linear least squares problem," *Calcolo. A Quarterly on Numerical Analysis and Theory of Computation*, vol. 54, no. 4, pp. 1121–1146, 2017.
- [54] H. Liu and F. Meng, "Some new nonlinear integral inequalities with weakly singular kernel and their applications to FDEs," *Journal of Inequalities and Applications*, vol. 2015, no. 209, pp. 1–17, 2015.
- [55] X. Zhang, L. Liu, Y. Wu, and L. Caccetta, "Entire large solutions for a class of Schrödinger systems with a nonlinear random operator," *Journal of Mathematical Analysis and Applications*, vol. 423, no. 2, pp. 1650–1659, 2015.

- [56] Z. Zong, F. Hu, C. Yin, and H. Wu, "On Jensen's inequality, Hölder's inequality, and Minkowski's inequality for dynamically consistent nonlinear evaluations," *Journal of Inequalities and Applications*, vol. 2015, no. 1, pp. 1–18, 2015.
- [57] X. Hao, L. Liu, and Y. Wu, "Positive solutions for nonlinear fractional semipositone differential equation with nonlocal boundary conditions," *Journal of Nonlinear Science and Applications*, vol. 9, no. 6, pp. 3992–4002, 2016.
- [58] X. Hao, L. Liu, and Y. Wu, "Iterative solution for nonlinear impulsive advection- reaction-diffusion equations," *Journal of Nonlinear Science and Applications*, vol. 9, no. 6, pp. 4070–4077, 2016.
- [59] P. Li, "Two classes of linear equations of discrete convolution type with harmonic singular operators," *Complex Variables and Elliptic Equations*, vol. 61, no. 1, pp. 67–75, 2016.
- [60] Z. Zheng, "Invariance of deficiency indices under perturbation for discrete Hamiltonian systems," *Journal of Difference Equations and Applications*, vol. 19, no. 8, pp. 1243–1250, 2013.
- [61] M. Han, X. Hou, L. Sheng, and C. Wang, "Theory of rotated equations and applications to a population model," *Discrete and Continuous Dynamical Systems- Series A*, vol. 38, no. 4, pp. 2171–2185, 2018.
- [62] J. Cai and H. Li, "A new sufficient condition for pancyclability of graphs," *Discrete Applied Mathematics*, vol. 162, pp. 142–148, 2014.
- [63] L. L. Liu and B.-X. Zhu, "Strong q-log-convexity of the Eulerian polynomials of Coxeter groups," *Discrete Mathematics*, vol. 338, no. 12, pp. 2332–2340, 2015.
- [64] B. Zhang, "Remarks on the maximum gap in binary cyclotomic polynomials," *Bulletin Mathématique De La Société Des Sciences Mathématiques De Roumanie*, vol. 59, no. 1, pp. 109–115, 2016.
- [65] L. Wang, "Intuitionistic fuzzy stability of a quadratic functional equation," *Fixed Point Theory and Applications*, vol. 2010, Article ID 107182, 7 pages, 2010.
- [66] X. Du and Z. Zhao, "On fixed point theorems of mixed monotone operators," *Fixed Point Theory and Applications*, vol. 2011, Article ID 563136, 8 pages, 2011.
- [67] B. Zhu, L. Liu, and Y. Wu, "Local and global existence of mild solutions for a class of nonlinear fractional reaction-diffusion equations with delay," *Applied Mathematics Letters*, vol. 61, pp. 73–79, 2016.

Research Article

Roads and Intersections Extraction from High-Resolution Remote Sensing Imagery Based on Tensor Voting under Big Data Environment

Ke Sun ^{1,2}, Junping Zhang ¹, and Yingying Zhang¹

¹*School of Electronics and Information Engineering, Harbin Institute of Technology, 92 West Dazhi Street, Harbin 150001, China*

²*Software College, Shenyang Normal University, 253 Northern Huanghe Street, Shenyang 110034, China*

Correspondence should be addressed to Junping Zhang; zhangjp@hit.edu.cn

Received 3 December 2018; Revised 30 January 2019; Accepted 13 February 2019; Published 4 March 2019

Guest Editor: Qingchen Zhang

Copyright © 2019 Ke Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Currently, big data is a new and hot object of research. In particular, the development of the Internet of things (IoT) results in a sharp increase in data. Enormous amounts of networking sensors are constantly collecting and transmitting data for storage and processing in the cloud including remote sensing data, environmental data, geographical data, etc. Road information extraction from remote sensing data is mainly researched in this paper. Roads are typical man-made objects. Extracting roads from remote sensing imagery has great significance in various applications such as GIS data updating, urban planning, navigation, and military. In this paper a multistage and multifeature method to extract roads and detect road intersections from high-resolution remotely sensed imagery based on tensor voting is presented. Firstly, the input remote sensing image is segmented into two groups including road candidate regions and nonroad regions using template matching; then we can obtain preliminary road map. Secondly, nonroad regions are removed by geometric characteristics of road (large area and long strip). Thirdly, tensor voting is used to overcome the broken roads and discontinuities caused by the different disturbing factors and then delete the nonroad areas that are mixed into the road areas due to mis-segmentation, improving the completeness of extracted roads. And then, all the road intersections are extracted by using tensor voting. The experiments are conducted on different remote sensing images to test the effectiveness of our method. The experimental results show that our method can get more complete and accurate extracted results than the state-of-the-art methods.

1. Introduction

Recently, with the continuous development of Internet of things (IoT) big data, mobile Internet, grid computing, cloud computing, and other new technologies, system integration becomes more complex. When processing information and data, it encounters many challenges such as data storage and management, efficient processing of massive data, structured and unstructured data fusion and analysis, and multitype data visualization. In particular, remote sensing technologies are promoted quickly; the spectral, spatial, radiative, and temporal resolution of remote sensing data are becoming higher and higher, which contain abundant data. Remote sensing data has the distinctive big data characteristics such as large capacity, high efficiency, multitype, and high value. All the above trends indicate that remote sensing has entered

into a big data era. Based on the aerospace science and technologies, an integrated space-air information network has been formed, which provides ultrahigh dimensional and frequency earth observation data. Remote sensing big data is a revolution of traditional data processing and information extraction methods [1]. The traditional processing methods cannot meet the precision and efficiency requirements of remote sensing big data [2, 3].

How to extract information of interest from remote sensing images quickly and efficiently has always been a research hotspot in the field of remote sensing data processing. The acquisition of road attribute information is an important part of it [4]. In high-resolution remote sensing images, many narrow roads that are difficult to discern on the low-resolution image can be distinguished. However, nonobject noise also increases. Currently, there are two main problems

for extracting road information from high-resolution remote sensing images: (1) straight line inside road has the same direction with road (such as road boundary). (2) Straight line has the different directions with road (such as zebra crossing) [5]. If the resolution is higher, the buildings are clearer. The top or the shadow of the buildings often forms road parallel lines. What is more, there are many cars and trees. These factors would result in difficult problems of road extraction.

Due to the complexity and diversity of roads in the real world, existing road extraction methods partially solves the problems at some stages (such as filtering [8], segmentation [9] in preprocessing stage, the extraction stage [10], split [11], and merge [12] in postprocessing stage). The methods of road extraction are reviewed in detail in [13, 14], including seed point-based [15], knowledge based method [16], and dynamic programming [17]. In addition, some researchers divided road extraction methods into automatic and semi-automatic methods. Wang *et al.* [18] proposed an object-oriented method for extracting roads. They selected some spectral features and textures parameters and used object-oriented methods to extract roads from the input images. Saati *et al.* [19] proposed an automatic method to extract road centerlines from SAR images. They extracted three features of the road, defining the road features by the backscattering coefficient of each pixel and the adjacent pixels of the SAR images. The feature extracted by the fusion was then used to detect the road regions by using a fuzzy inference system. Wei *et al.* [20] proposed an end-to-end road centerline extraction method by learning a confidence graph. They extracted road centerline directly from images, rather than obtaining the road centerline by thinning the road segments. Gupta *et al.* [21] developed an automatic method to extract roads by using fuzzy, genetic algorithm, and mathematical morphology.

Recently, tensor voting algorithm is widely used for feature extraction, especially in remote sensing images. Miao *et al.* [22] extracted road intersection areas by tensor voting, and the roads were decomposed to isolated parts at the detected junction areas; then they were able to extract the centerlines for each individual section of the road. Zhang *et al.* [23] used tensor voting to extract roads and road intersections from remote sensing images. Ishida *et al.* [24] proposed two voting schemes to estimate and classify the position accurately. The first was based on geometric feature extraction of multiframe sparse tensor voting, and the second was contour localization using the resulting tensor field. Zhu *et al.* [25] presented a tensor voting method for image denoising; they considered that the calculation of voting field was the key step of image denoising based on tensor voting, and it was a robust feature extraction method.

For the last few years, the rapid development of the earth observation capability and the intelligent computing technology [26, 27] has provided opportunities for the advancement and even revolution of remote sensing information technology. Remote sensing information technology is gradually entering the era of remotely sensed big data [28] era. This will inevitably put forward higher requirements for automatic analysis and mining of big data. Deep learning [29–31] has been widely used to extract information from remote sensing images, and its accuracy has exceeded the accuracy of manual

recognition. The great success of deep learning in the field of computer vision [32–34] provides an important opportunity for big data to extract information intelligence from remote sensing imagery.

At present, many researchers have proposed many methods for road extraction from different perspectives for different remotely sensed imagery and have made great progress. Due to the variety of road forms and the complexity of surrounding environment in reality, most of the existing methods extract roads from specific remote sensing images and road information of specific areas. There is no extraction method that can be applied to all remote sensing images or all types of roads. This paper attempts to divide the road extraction method into different steps and clarify the specific tasks of each step. We propose a multistage and multifeature method based on tensor voting, which includes template matching, geometric feature of road and tensor voting to extract roads, and road intersections in high-resolution remotely sensed imagery. The structure of this paper is as follows. Section 2 introduces the proposed road and intersection extraction method. The experiments and analysis are shown in Section 3. Conclusions are given in Section 4.

2. Methodology

The proposed method for extracting roads and road intersections is based on multistage and multifeature from high-resolution remotely sensed imagery. The method can extract pure road regions and accurately detect all road intersections from the input image. It consists of three stages. We first segment the input images and get the candidate road regions and then eliminate non-road areas by using road geometric features and get initial road maps and finally purify and smooth the initial road areas and extract road intersections. The overall strategy of the proposed method is shown in Figure 1.

2.1. Image Segmentation. The purpose of this step is to segment the input image and obtain preliminary road regions. The SUSAN (Smallest Univalued Segment Assimilating Nucleus) [35] algorithm is adopted to segment images. SUSAN algorithm is the representative of template matching, which was proposed by Smith and Brady. SUSAN algorithm moves the template on image, whether the template center is an edge point determined by whether the matching degree of it reaches the threshold. Because the SUSAN algorithm is based on the grayscale comparison of the pixels in the neighborhood and does not need to calculate the gradient, the interference range of the noise is obviously smaller than the edge detection based on the gradient.

The SUSAN algorithm uses a circular template to move over the image [36]. If the difference between the gray level of the pixel in the template and that of the central pixel is lower than the given threshold [37], then it is considered that the point has a similar gray level with the central pixel of the template [38]. A region consisting of pixels satisfying such a condition is called an USAN (Univalued Segment Assimilating Nucleus). In Figure 2, *a* and *b* are completely located in the

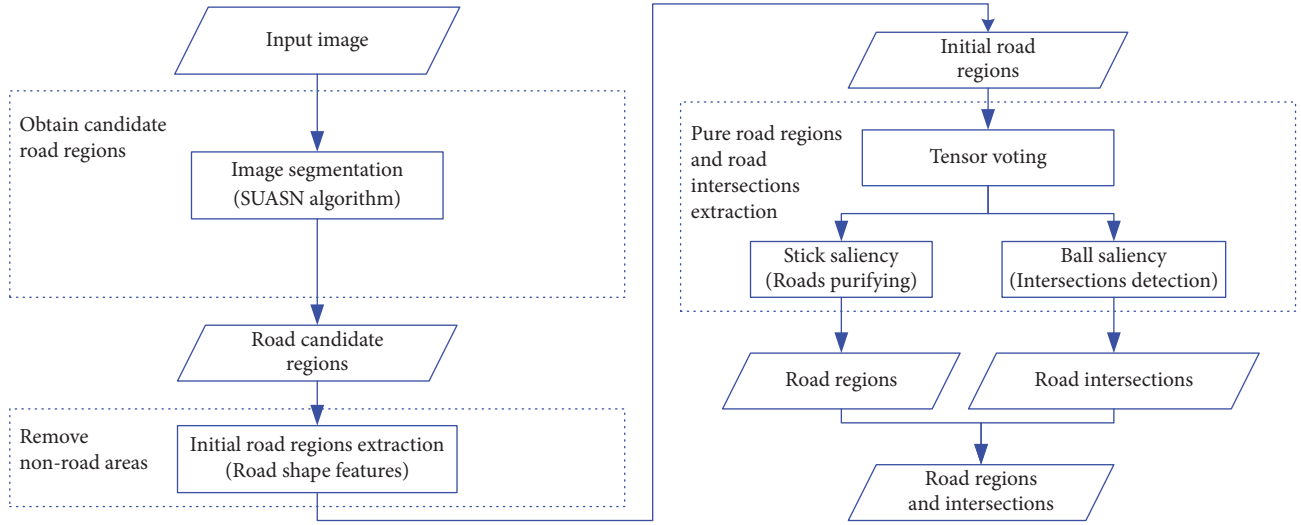


FIGURE 1: Overall strategy of the proposed method.

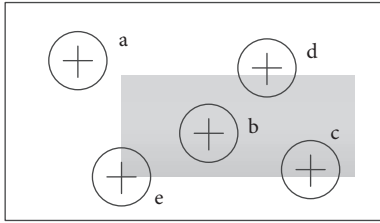


FIGURE 2: Principle of SUSAN feature detection.

foreground and background of the image, respectively. The area of USAN is the largest. In addition, c , d , e are moving closer to the edge, while the area of USAN is approaching to the minimum. The results show that the USAN values of edge pixels are less than or equal to half of the maximum value of the period. The edge points can be detected.

The steps of detecting image edge information by using SUSAN algorithm are as follows:

(1) Find out the maximum and minimum gray value I_{max} and I_{min} from the image, and calculate the best detection threshold:

$$t = \frac{(I_{max} - I_{min})}{10} \quad (1)$$

(2) Traverse the whole image and check location characteristics of each pixel according to the following formula:

$$c(r, r_0) = \begin{cases} 1 & |I(r) - I(r_0)| \leq t \\ 0 & |I(r) - I(r_0)| > t \end{cases} \quad (2)$$

where $c(r, r_0)$ is a discriminant function. If its value is 1, then the pixel is located in the USAN region; $I(r_0)$ is the gray value of the central pixel of the template; $I(r)$ is the gray value of any other pixel in the template.

The USAN area statistics for each pixel are as follows:

$$n(r_0) = \sum_{r \in D(r_0)} c(r, r_0) \quad (3)$$

where $D(r_0)$ is a circular template region and r_0 is the center of the circle.

(3) The threshold value g is set. When $n(r_0) < g$, the detected pixel position r_0 is assumed to be at the edge position of the image.

(4) Traverse the whole image to detect the complete edge information.

According to the characteristics of the road in the remote sensing image, in the obtained edge results map, the roads are located in the nonedge homogeneous regions. The results of edge extraction map are segmented using a threshold, and road candidate areas are obtained.

2.2. Road Information Extraction. After the previous algorithm is implemented, the image can be segmented into two categories, homogeneous regions and edges. Road segments are in homogeneous regions. But buildings or bare soil segments are also mixed into road areas. The geometric characteristics of road will be used to find out the potential road regions and eliminate other nonroad segments.

According to the geometric features, roads do not have small areas, and the length is much larger than the width. So roads are obviously displayed as narrow, long, linear feature [39]. Through the identification of the big-area regions and linear features, it is easy to remove the small areas and nonroad regions.

In this paper, the result of image segmentation is processed from the two aspects aiming to eliminate the nonroad regions.

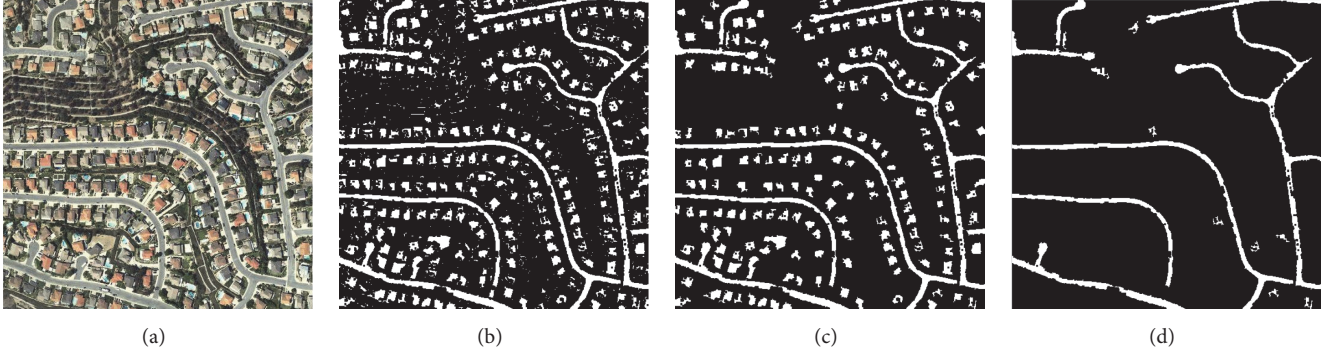


FIGURE 3: Results of removing nonroad regions using shape features on QuickBird image. (a) Input image. (b) Results of preliminary road extraction. (c) Results of removing small area regions. (d) Results of eliminating nonlinear features.

(1) Compute the area of each independent region in the result of segmentation, and remove the regions whose value is less than the threshold. It can be expressed as

$$\text{Region} = \begin{cases} \text{road region} & R_{\text{area}} > R_{\text{thre}} \\ \text{non-road region} & \text{otherwise} \end{cases} \quad (4)$$

where R_{area} is the area of region and R_{thre} is the threshold.

(2) The ratio of length to width of the minimum external rectangle in each region is calculated in the result of segmentation. If the ratio of length to width is greater than the set threshold in this region, then the region can be marked as the road area. The ratio of length to width of the region is defined as

$$R = \frac{L_{\min}}{W_{\min}} \quad (5)$$

where L_{\min} is the length of the minimum external rectangle of the region and W_{\min} is the width of the rectangle.

When road is bent and the linear features cannot be well described, there will be errors in calculating the aspect ratio by using the above formula. The formula needs to be modified to overcome this limitation. We take the total number of pixels in the detection region as the area of the external rectangle and create a new rectangle with the diagonal line of the smallest external rectangle as the long edge. Therefore,

$$W = \frac{n}{L} \quad (6)$$

where W is the width of the new rectangle; n is the number of pixels of the region; and L is the length of the new rectangle; the value of L can be calculated by

$$L = \sqrt{L_{\min}^2 + W_{\min}^2} \quad (7)$$

Therefore, in practical application, the aspect ratio of the region can be calculated by

$$R_{\min} = \frac{L}{W} = \frac{(L_{\min}^2 + W_{\min}^2)}{n} \quad (8)$$

The results of removing nonroad regions using road's shape features are shown in Figure 3. The test image is a QuickBird image; the resolution is 0.61m/pixel.

2.3. Tensor Voting. After previous extraction, the main road regions have been extracted. However, there are still some large holes, gaps, and many other nonroad regions in the results of detected roads. Generally, mathematical morphology is used to delete the isolated nonroad regions and connect the small gaps of the road. However, this method cannot fill the large holes in the results. It cannot connect the large gaps and completely delete the nonroad regions [40]. Therefore, this paper uses tensor voting [41–43] algorithm to solve the above problems.

Tensor voting is a robust method for feature extraction. The main idea of tensor voting algorithm is that every point in the space collects the tensor information from other points in the neighborhood and encodes it as a new tensor to be used in the next voting. After the voting, it decomposes the new tensor. Thus, the salience map of various characteristics is obtained. It can be used to detect the geometric structures, which consists of two parts, the tensor representation of the data (tensor coding) and the nonlinear voting between tensors (tensor voting).

Tensor voting can obtain the salient features of the image, so it is possible to detect the geometric features of the typical objects. Road has obvious elongated shape, and intersection has obvious ball shape. We can use tensor voting to extract roads and detect road intersections from remotely sensed images.

Firstly, the second-order positive semidefinite symmetric tensor is used to represent the direction and significance of the pixels in the image. In a two-dimensional space, tensor can be decomposed into a linear combination of eigenvalues and eigenvectors:

$$T = (\lambda_1 - \lambda_2) \vec{e}_1 \vec{e}_1^T + \lambda_2 (\vec{e}_1 \vec{e}_1^T + \vec{e}_2 \vec{e}_2^T) \quad (9)$$

where λ_1 and λ_2 are nonnegative eigenvalues and $\lambda_1 > \lambda_2$. \vec{e}_1 and \vec{e}_2 are the corresponding eigenvectors. $\vec{e}_1 \vec{e}_1^T$ is stick tensor (representing curve characteristics), \vec{e}_1 represents the direction of a curve, and $(\lambda_1 - \lambda_2)$ is a significant index of the curve. $\vec{e}_1 \vec{e}_1^T + \vec{e}_2 \vec{e}_2^T$ is ball tensor (representing node characteristics), and λ_2 is a significant index of the node.

Then tensor voting is executed. Firstly, it initializes the pixels to the ball tensor and vote; the voting field is shown

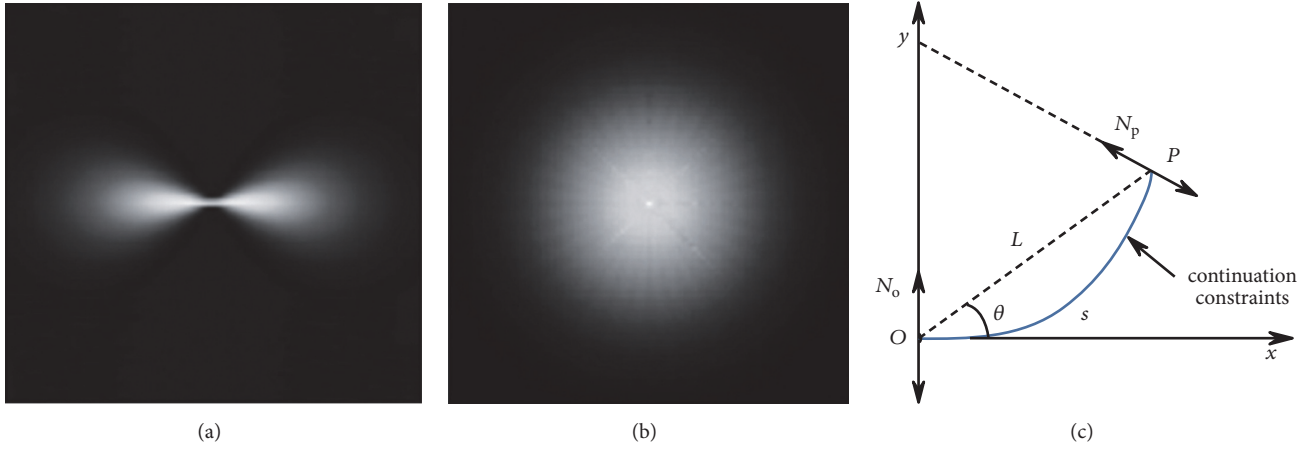


FIGURE 4: Tensor voting. (a) Ball voting field. (b) Stick voting field. (c) Schematic of tensor voting rules.

in Figure 4(a). Next, the initial direction is obtained by analyzing the number of votes received at each point, and the initial direction is assigned to the stick voting field as shown in Figure 4(b). Then voting in the stick field, the voting rules are as follows (see Figure 4(c)); in coordinate system Oxy , there are two tensors at O and P , respectively, where O is the voting point, P is the receiving point, and N_O and N_P are their normal vectors, respectively.

Let L denote the distance between the voting point and the receiving point; then θ is the angle between the tangent line of the voting point on a close circle and the straight line between the voting point and the receiving point, and s and k denote the arc length and curvature, respectively.

Then voting of P received by tensor at O can be defined as

$$V(P) = DF(s, k, \sigma) N_P N_P^T$$

$$N_P = N_O [-\sin(2\theta), \cos(2\theta)]^T, \quad (10)$$

$$DF(s, k, \sigma) = e^{-((s^2 + ck^2)/\sigma^2)}$$

where $DF(s, k, \sigma)$ is saliency decay function and σ is the scale factor that determines the size of the voting fields, and the only parameter that can be changed. c is the parameter to control the degree of attenuation.

After tensor voting, every pixel collects all votes projected by the tensor in its neighborhood and integrates them into a new tensor. The accumulation of votes is obtained by adding up the tensor. Finally, the new tensor is decomposed into the form of (9) and the eigenvalue is calculated. The saliency of the probability of each point in the image is obtained. The structural features of the image can be judged by calculating the saliency.

When the saliency of a pixel is $(\lambda_1 - \lambda_2) > \lambda_2$, the pixel is a point on the curve, which can be judged as a road point. If $\lambda_1 \approx \lambda_2 > 0$, the point is a region or junction point, but the area belongs to road intersection and has more than two branches, so it can be detected. In other cases, the pixel is singular and does not require processing.

The directionless point does not have the stick tensor part, for the segmented road with no direction; it cannot vote out

the stick, so it is impossible to extract the saliency of the curve. Therefore, all road points are coded by ball tensor, and each road point is coded as $T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$; then all coding points are voted sparsely in a sphere of voting, which makes the road points have a certain directivity. Then the ball tensor encodes the nonroad points and reconducts a dense voting in the stick voting field for all points. After two votes, the stick voting tensor obtained from each point is decomposed, and the curve saliency $\lambda_1 - \lambda_2$ of each road point can be extracted. To extract all points, it should satisfy $\lambda_1 = \lambda_2$, which is a cross region; it is as a candidate area of road intersections.

3. Experiments

3.1. Road and Intersection Extraction. The first experiment is conducted on a IKONOS remote sensing image with a spatial resolution 1.0 m/pixel and the size 1024×1024 . There are circular, straight, and curved roads in the image, which belong to the hybrid road network (see Figure 5(a)) [44]. The SUSAN algorithm is applied to segment the input image, the image is segmented into two categories, road regions and nonroad regions, and then shape features of road are used to delete nonroad regions from preliminary road extraction results which are shown in Figure 5(b). Then we use tensor voting to delete the nonroad areas and connect the broken road areas. The results of stick saliency and ball saliency after vote analysis are illustrated in Figures 5(c) and 5(d), respectively. Figures 5(e) and 5(f) show the results of ball regions and stick regions, respectively, which are potential roads and intersections. Figure 5(g) shows the results by our method. Figure 5(h) shows the extracted roads and road intersections overlapped on the test image.

According to the experimental results, we purify and optimize the preliminary results of road extraction. After tensor voting, the results are very satisfactory. The ball tensor voting to the road intersection detection is also very accurate. From the results of the experiment, the proposed method can extract complete road segments accurately; meanwhile it can find all the road intersections from the input image. Therefore, it can be concluded that the proposed method can

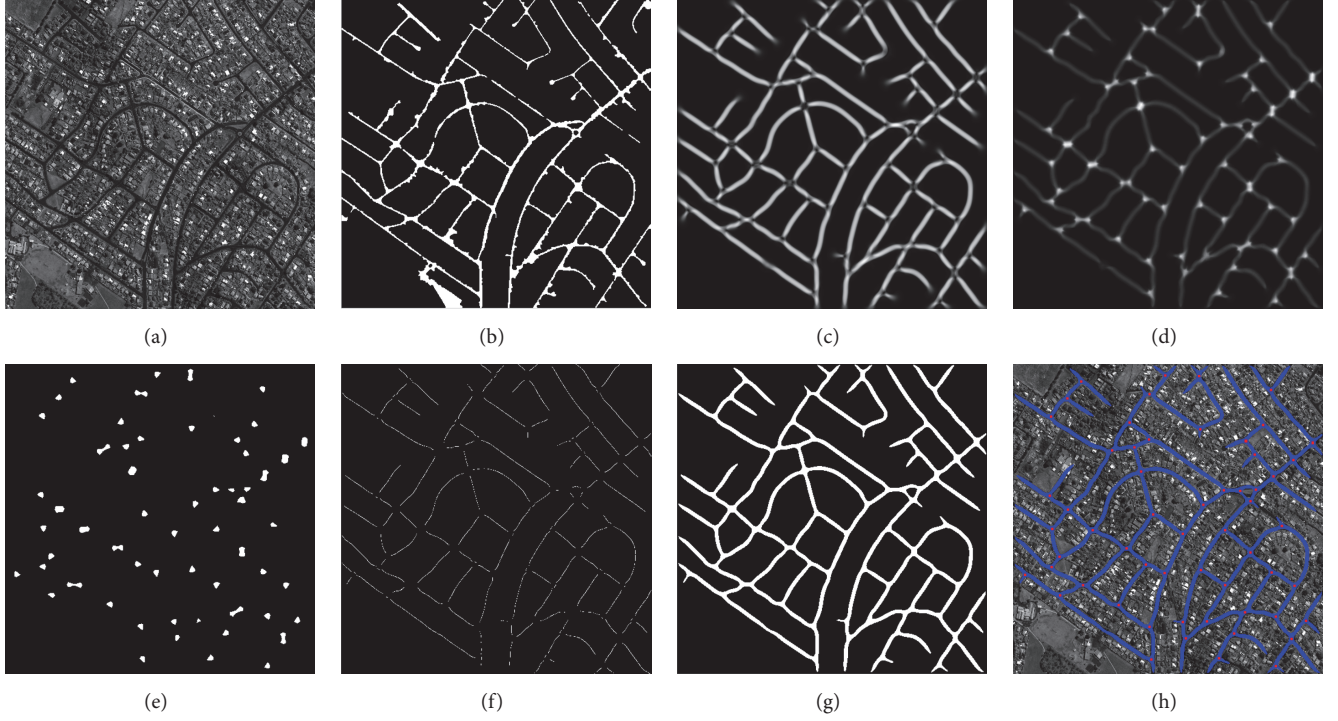


FIGURE 5: Results of the first experiment. (a) Testing image. (b) Coarse road regions extracted by template matching and road shape features. (c) Results of stick salience by vote analysis. (d) Results of ball salience by vote analysis. (e) Ball regions. (f) Stick regions. (g) Results of extracted road by the proposed method. (h) The extracted roads and road intersections overlapped on the test image.

effectively detect all the road regions and road intersections in IKONOS panchromatic image.

The results of road and road intersection extraction using proposed method from QuickBird image, the size of the test image being 512×512 , are shown in Figure 6. The result of road extraction using SUSAN algorithm and geometric filtering is illustrated in Figure 6(a). Figure 6(b) shows the ball regions detected by tensor voting; we can see that all the potential intersections have been detected. The result of roads extraction by our method is shown in Figure 6(c). Figure 6(d) illustrates the result of the overlay of the detected roads and intersections on the original remote sensing image.

The next two experiments use Geoeeye image and WordView-I image. The size of the input image is 512×512 and 1500×1500 , respectively. The results of each step in the experiment using Geoeeye image are shown in Figures 7(a)–7(c), and those using WordView-I image are shown in Figures 7(d)–7(f).

From all the experimental results in Figures 5–7, we can see that our method can extract pure roads obtained by different sensors and can detect almost all road intersections. Then the accuracy of the method will be quantitatively analyzed in the next section.

3.2. Accuracy Assessment of Purification Result. For evaluating the proposed method, there are five indexes of accuracy used in this paper [45], which are defined as [46]

$$\text{Completeness} = \frac{TP}{TP + FN} \times 100\% \quad (11)$$

$$\text{Correctness} = \frac{TP}{TP + FP} \times 100\% \quad (12)$$

$$\text{Quality} = \frac{TP}{TP + FP + FN} \times 100\% \quad (13)$$

$$\text{Omit} = \frac{FN}{TP + FN} \times 100\% \quad (14)$$

$$\text{Redundancy} = \frac{FP}{TP + FN} \times 100\% \quad (15)$$

where TP is the area where the extracted road regions and reference roads coincide with each other, FP is the area where the roads are extracted but does not exist in the reference roads, and FN is the area that exists in the reference roads not extracted.

We use the number of pixels in the regions as the area of the regions. The reference road maps are hand-drawn according to the input image. Figure 8 illustrates the evaluation principle [47].

Then we use completeness, correctness, quality, omit and redundancy indicators to analyze the previous experimental results. The reference road, preliminary results, and results extracted by using our method are shown in Figure 9. Reference road maps of the test image being hand-drawn are shown in Figures 9(a), 9(d), 9(g), and 9(j). Figures 9(c), 9(f), 9(i), and 9(l) show the results of road extracted by using our method. Figures 9(b), 9(e), 9(h), and 9(k) show the results without further processing by tensor voting. By comparing the results obtained with proposed method and those without

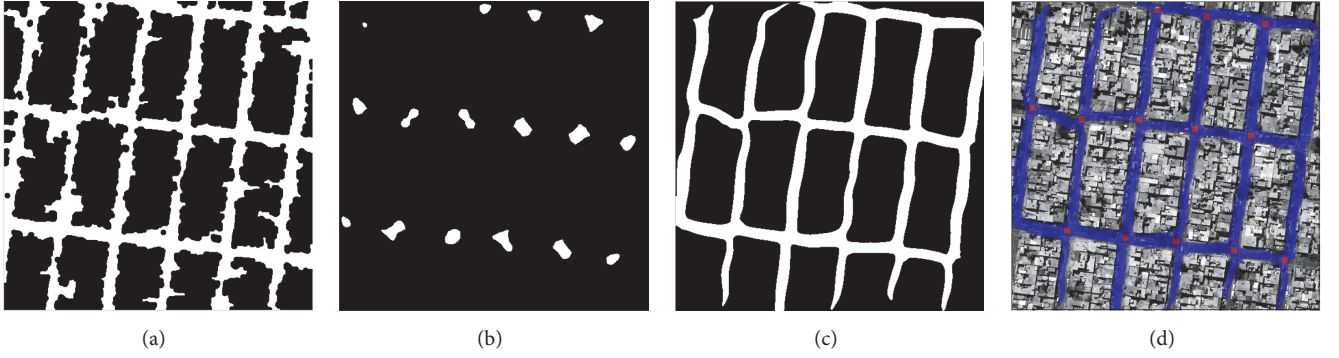


FIGURE 6: Results for QuickBird image with spatial size of 512×512 . (a) Coarse road regions. (b) Results of ball salience. (c) Results of stick salience. (d) The extracted roads and road intersections overlapped on the input image.

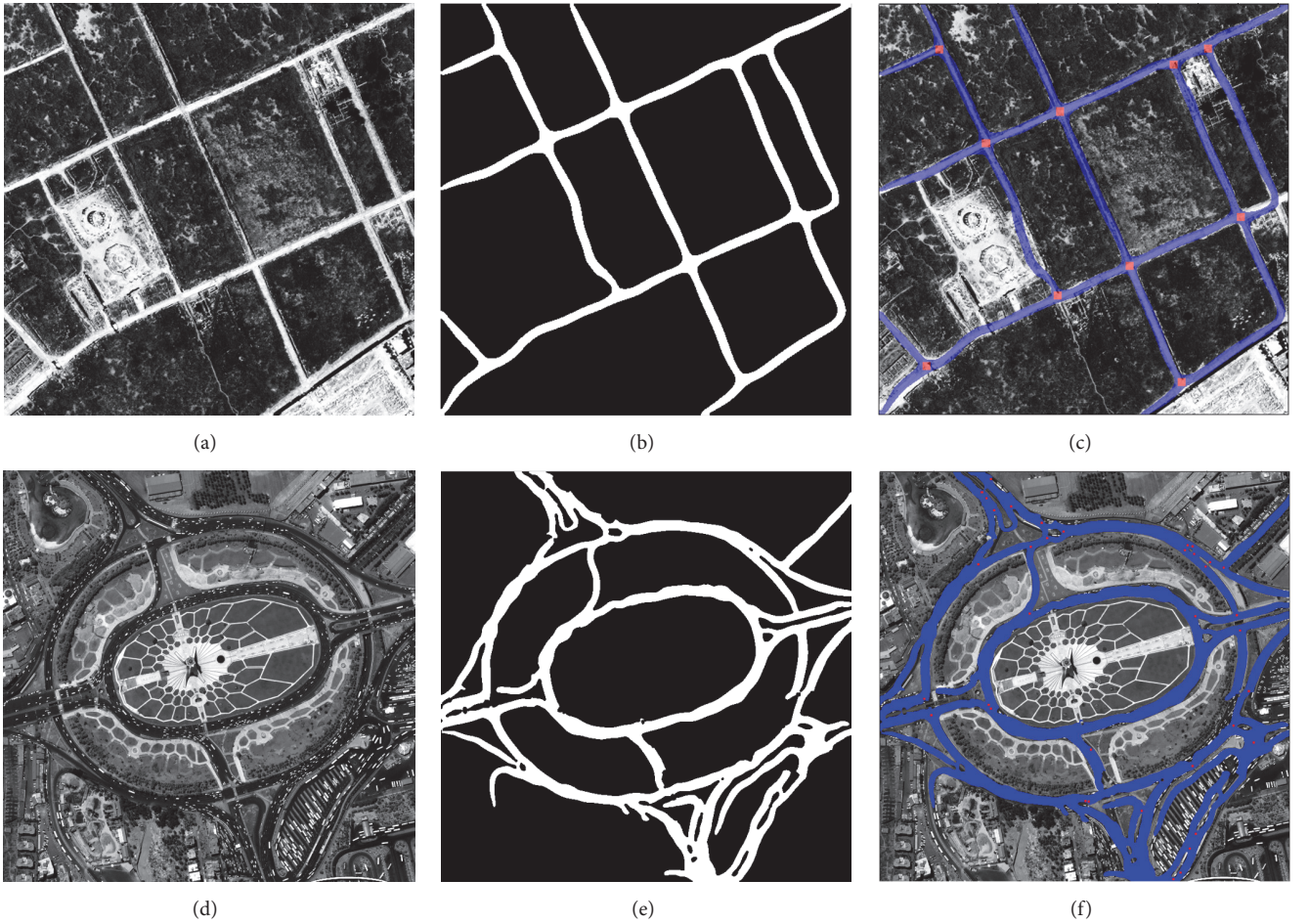


FIGURE 7: Results for Geoeye image and WorldView-I image. (a)(d) Testing image. (b)(e) Coarse road regions. (c)(f) The extracted roads and road intersections.

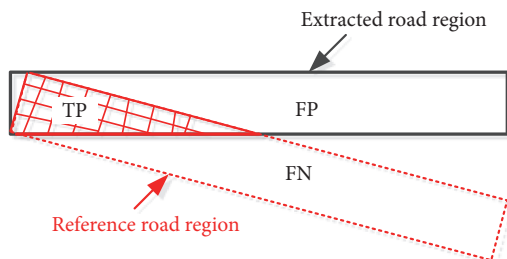


FIGURE 8: The evaluation principle.

tensor voting, we can see that the results extracted by our method are smoother than those without tensor voting, and they are much more similar to the reference roads.

All indicators with and without using tensor voting method to extract roads are shown in Table 1. It can be seen that the precision and extraction rate of the road extracted by this method are very high, while the false alarm rate and the missing rate are very low, which shows that the method can extract road information effectively.

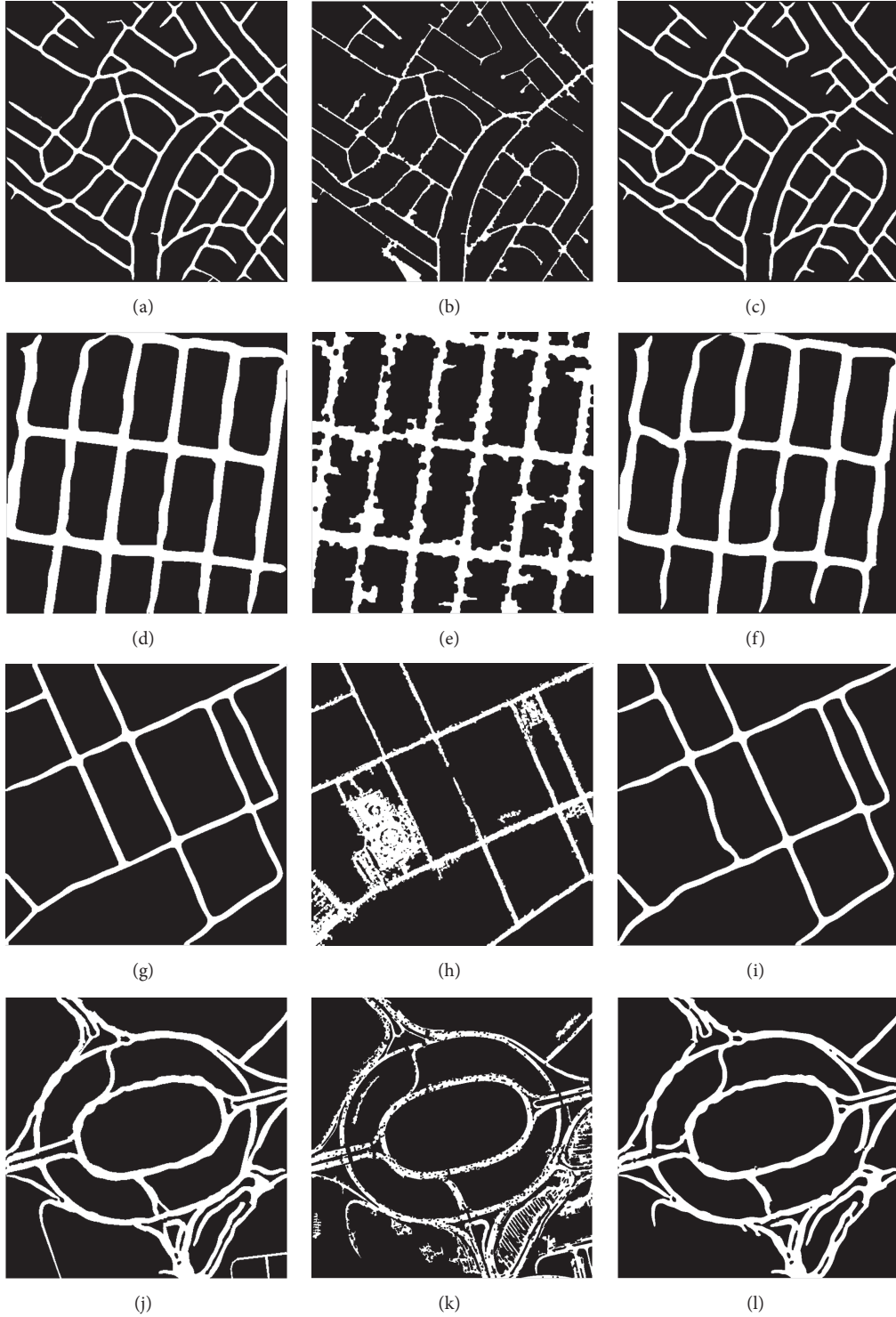


FIGURE 9: Results of road extraction without using and using the proposed method. (a)(d)(g)(j) Reference road maps. (b)(e)(h)(k) Results of coarse roads extracted without using the proposed method. (c)(f)(i)(l) Results of roads extracted by the proposed method.

3.3. Comparison with the State-of-the-Art Methods. The method in this paper is compared with SSC [6] and knowledge-based [7]. Figure 10 gives the comparison results of the methods mentioned in SSC, knowledge-based, and proposed methods. From the results of extraction, we can

see that the results extracted by our method are better than those obtained by the other two methods. The proposed method by using tensor voting is superior in eliminating nonroad information and connecting the broken roads.

TABLE 1: Performance evaluation of the proposed method.

Image	IKONOS		QuickBird		Geoeye	WorldView-I
	Figure 9(c)	Figure 9(b)	Figure 9(f)	Figure 9(e)	Figure 9(i)	Figure 9(l)
TP(pixels)	148922	125545	62612	64020	28767	429342
FP(pixels)	1502	27816	2805	20843	2553	9701
FN(pixels)	2583	26030	8541	7133	1969	19618
Completeness	98.30%	82.83%	88.00%	89.98%	93.59%	95.63%
Correctness	99.00%	81.86%	95.71%	75.44%	91.85%	97.79%
Quality	97.33%	69.98%	84.66%	69.59%	86.42%	93.61%
Omit	1.70%	17.17%	12.00%	10.02%	6.41%	4.37%
Redundancy	0.99%	18.35%	3.94%	29.29%	8.31%	2.16%

TABLE 2: Comparison of the proposed method with the state-of-the-art methods.

Method	Completeness	Correctness	Quality	Omit	Redundancy
SSC [6]	85.05%	85.94%	74.66%	14.95%	13.92%
Knowledge-based [7]	64.43%	76.03%	53.55%	35.57%	20.31%
Proposed method	97.45%	93.70%	93.52%	2.55%	1.77%

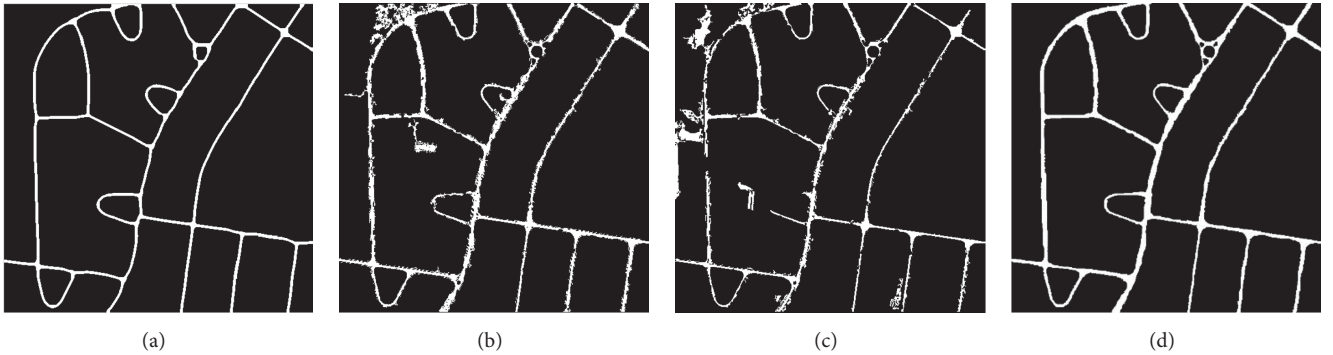


FIGURE 10: The results of the proposed method and the state-of-the-art methods. (a) Reference maps. (b) SSC [6]. (c) Knowledge-based method [7]. (d) The proposed method.

In order to evaluate these methods quantitatively and compare the extraction effect results of the three methods, the above five measurement indexes are calculated, and the results are shown in Table 2. It is clear from the quantitative results that the method proposed in this paper is the best.

4. Conclusions

A multistage and multifeature method is proposed to extract roads and road intersections from high-resolution remote sensing images based on template matching and tensor voting. Firstly, SUSAN algorithm is used to segment the input image, to find out the potential road candidate points, and obtain initial road regions. Secondly, shape features of road are used to identify the initial road areas and remove the nonroad areas for the purpose of obtaining purer segments of road. Due to the influence of cars, pedestrians, and roadside trees on the road, there will be some holes and gaps in the road regions, and some nonroad areas are stuck in the road areas. Thirdly, tensor voting is used to delete the nonroad areas that

adhere to the road regions, fill the holes in the roads and connect the gaps, and extract the pure and complete segments of the road. Finally, we extract all the ball regions from the road results and then detect all the information of the road intersections from the ball regions according to the characteristics of the road intersections. The experimental results show that our method can extract pure road information and accurately detect road intersections. Compared with other methods, our method significantly improves the accuracy of road extraction. Future work will focus on reducing the effects of interference factors in the preprocessing stage, automatically selecting the optimal parameters for image segmentation during the image segmentation phase, so as to make the road extraction more accurate.

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 they have no conflicts of interest.

Acknowledgments

The work was supported by the National Natural Science Foundation of China under Grant 61871150.

References

- [1] Q. Zhang, H. Zhong, L. T. Yang, Z. Chen, and F. Bu, "PPHOCFS: privacy preserving high-order CFS algorithm on the cloud for clustering multimedia data," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 4s, pp. 66:1–66:15, 2016.
- [2] Y. Ma, H. Wu, L. Wang et al., "Remote sensing big data computing: challenges and opportunities," *Future Generation Computer Systems*, vol. 51, pp. 47–60, 2015.
- [3] J. Gao, J. Li, and Y. Li, "Approximate event detection over multi-modal sensing data," *Journal of Combinatorial Optimization*, vol. 32, no. 4, pp. 1002–1016, 2016.
- [4] Y. Nakaguro, S. S. Makhanov, and M. N. Dailey, "Numerical experiments with cooperating multiple quadratic snakes for road extraction," *International Journal of Geographical Information Science*, vol. 25, no. 5, pp. 765–783, 2011.
- [5] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan, "Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 6, pp. 3322–3337, 2017.
- [6] W. Shi, Z. Miao, Q. Wang, and H. Zhang, "Spectral-spatial classification and shape features for urban road centerline extraction," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 4, pp. 788–792, 2014.
- [7] J. Wang, Q. Qin, J. Zhao et al., "A knowledge-based method for road damage detection using high-resolution remote sensing image," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2015*, pp. 3564–3567, July 2015.
- [8] J. Q. Zhao, J. Yang, P. X. Li et al., "Semi-automatic road extraction from SAR images using EKF and PF" in *Proceedings of the ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-7/W4, pp. 227–230, 2015.
- [9] Z. L. Miao, W. Z. Shi, A. Samat et al., "Information fusion for urban road extraction from VHR optical satellite images," *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing*, vol. 9, no. 5, pp. 1–14, 2016.
- [10] C. Poullis, "Tensor-Cuts: a simultaneous multi-type feature extractor and classifier and its application to road extraction from satellite images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 95, pp. 93–108, 2014.
- [11] R. Liu, Q. Miao, B. Huang, J. Song, and J. Debayle, "Improved road centerlines extraction in high-resolution remote sensing images using shear transform, directional morphological filtering and enhanced broken lines connection," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 300–311, 2016.
- [12] M. Song and D. Civco, "Road extraction using SVM and image segmentation," *Photogrammetric Engineering and Remote Sensing*, vol. 70, no. 12, pp. 1365–1371, 2004.
- [13] J. B. Mena, "State of the art on automatic road extraction for GIS update: a novel classification," *Pattern Recognition Letters*, vol. 24, no. 16, pp. 3037–3058, 2003.
- [14] S. Das, T. T. Mirnalinee, and K. Varghese, "Use of salient features for the design of a multistage framework to extract roads from high-resolution multispectral satellite images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 10, pp. 3906–3931, 2011.
- [15] H. R. R. Bakhtiari, A. Abdollahi, and H. Rezaeian, "Semi automatic road extraction from digital images," *Egyptian Journal of Remote Sensing and Space Science*, vol. 20, no. 1, pp. 117–123, 2017.
- [16] J. D. Wegner, J. A. Montoya-Zegarar, and K. Schindler, "Road networks as collections of minimum cost paths," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 108, pp. 128–137, 2015.
- [17] M. Barzohar and D. B. Cooper, "Automatic finding of main roads in aerial images by using geometricstochastic models and estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 707–721, 1996.
- [18] J. H. Wang, Q. M. Qin, X. C. Yang et al., "Automated road extraction from multi-resolution images using spectral information and texture," in *Proceedings of the IEEE Geoscience and Remote Sensing Symposium*, pp. 533–536, 2014.
- [19] M. Saati, J. Amini, and M. Maboudi, "A method for automatic road extraction of high resolution SAR imagery," *Journal of the Indian Society of Remote Sensing*, vol. 43, no. 4, pp. 697–707, 2015.
- [20] Y. J. Wei, X. Y. Hu, and J. Q. Gong, "End-to-end road centerline extraction via learning a confidence map," in *Proceedings of the 10th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS)*, pp. 1–5, 2018.
- [21] S. Gupta and G. Singh, "A new technique for road extraction using mathematical, morphology, fuzzy and genetic algorithm," *International Journal of Engineering Research & Applications*, vol. 4, no. 2, pp. 341–346, 2014.
- [22] Z. Miao, W. Shi, H. Zhang, and X. Wang, "Road centerline extraction from high-resolution imagery based on shape features and multivariate adaptive regression splines," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 3, pp. 583–587, 2013.
- [23] Y. Zhang, J. Zhang, T. Li, and K. Sun, "Road extraction and intersection detection based on tensor voting," in *Proceedings of the 36th IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2016*, pp. 1587–1590, China, July 2016.
- [24] H. Ishida, K. Kidono, Y. Kojima, and T. Naito, "Road marking recognition for map generation using sparse tensor voting," in *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012*, pp. 1132–1135, Japan, November 2012.
- [25] Y. Zhu, W. Huang, P. Wen et al., "Tensor voting based method for Image de-noising," in *Proceedings of the IEEE 2010 International Conference on Intelligent Computing and Integrated Systems (ICISS)*, pp. 209–212, 2010.
- [26] D. Zhang, D. Zhang, H. Xiong, L. T. Yang, and V. Gauthier, "NextCell: predicting location using social interplay from cell phone traces," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 64, no. 2, pp. 452–463, 2015.
- [27] L. Shu, Y. Zhang, L. T. Yang, Y. Wang, and M. Hauswirth, "Geographic routing in wireless multimedia sensor networks," in *Proceedings of the 2008 2nd International Conference on Future Generation Communication and Networking, FGCN 2008*, pp. 68–73, China, December 2008.

- [28] P. Li, Z. Chen, L. T. Yang, Q. Zhang, and M. J. Deen, "Deep convolutional computation model for feature learning on big data in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 790–798, 2018.
- [29] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, "A double deep q-learning model for energy-efficient edge scheduling," *IEEE Transactions on Services Computing*, 2018.
- [30] Q. Zhang, L. T. Yang, Z. Chen, P. Li, and F. Bu, "An adaptive dropout deep computation model for industrial IoT big data learning with crowdsourcing to cloud computing," *IEEE Transactions on Industrial Informatics*, 2018.
- [31] P. Li, Z. Chen, L. T. Yang, J. Gao, Q. Zhang, and J. Deen, "An incremental deep convolutional computation model for feature learning on industrial big data," *IEEE Transactions on Industrial Informatics*, 2018.
- [32] P. Li, Z. Chen, L. T. Yang, L. Zhao, and Q. Zhang, "A privacy-preserving high-order neuro-fuzzy c-means algorithm with cloud computing," *Neurocomputing*, vol. 256, no. 20, pp. 82–89, 2017.
- [33] M. Dong, H. Li, K. Ota, L. T. Yang, and H. Zhu, "Multicloud-based evacuation services for emergency management," *IEEE Cloud Computing*, vol. 1, no. 4, pp. 50–59, 2014.
- [34] Q. Zhang, L. T. Yang, Z. Yan, Z. Chen, and P. Li, "An efficient deep learning model to predict cloud workload for industry informatics," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3170–3178, 2018.
- [35] X. Fan, Y. Cheng, and Q. Fu, "Moving target detection algorithm based on SUSAN edge detection and frame difference," in *Proceedings of the 2015 2nd International Conference on Information Science and Control Engineering, ICISCE 2015*, pp. 323–326, April 2015.
- [36] S. R. Gong, C. P. Liu, Y. Ji et al., *Advanced Image and Video Processing Using MATLAB*, Springer Nature America, Inc., 2019.
- [37] G. Sharma, F. Zhou, J. Liu et al., "An improved corner detection algorithm for image sequence," in *Proceedings of the International Symposium on Optoelectronic Technology and Application 2014 Image Processing and Pattern Recognition 2014*, Beijing, China, 2014.
- [38] B. Yu, A. Wang, Z. Liu, and M. Zhou, "Automatic localization and marking for features of skull by CT image," in *Proceedings of the 2009 International Conference on Information Technology and Computer Science (ITCS 2009)*, pp. 313–316, Kiev, Ukraine, July 2009.
- [39] C. Cao and Y. Sun, "Automatic road centerline extraction from imagery using road GPS data," *Remote Sensing*, vol. 6, no. 9, pp. 9014–9033, 2014.
- [40] W. Fengping and W. Weixing, "Road extraction using modified dark channel prior and neighborhood FCM in foggy aerial images," *Multimedia Tools and Applications*, no. 7, pp. 1–18, 2018.
- [41] H. B. Lin and W. Wang, "Feature preserving holes filling of scattered point cloud based on tensor voting," in *Proceedings of the IEEE 2016 International Conference on Signal and Image Processing (ICSIP)*, pp. 402–406, 2016.
- [42] J. Sreevalsan-Nair, A. Jindal, and B. Kumari, "Contour extraction in buildings in airborne lidar point clouds using multiscale local geometric descriptors and visual analytics," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 7, pp. 2320–2335, 2018.
- [43] H. Guan, J. Li, Y. Yu et al., "Iterative tensor voting for pavement crack extraction using mobile laser scanning data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1527–1537, 2015.
- [44] H. Gao, S. M. Feng, and C. X. Guo, "Research on selection method of urban road network structure form," in *Proceedings of the IEEE 2010 WASE International Conference on Information Engineering (ICIE)*, vol. 3, pp. 360–364, 2010.
- [45] Z. Miao, B. Wang, W. Shi, and H. Zhang, "A semi-automatic method for road centerline extraction from VHR images," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 11, pp. 1856–1860, 2014.
- [46] W. Shi, Z. Miao, and J. Debayle, "An integrated method for urban main-road centerline extraction from optical remotely sensed imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 6, pp. 3359–3371, 2014.
- [47] M. O. Sghaier and R. Lepage, "Road extraction from very high resolution remote sensing optical images based on texture analysis and beamlet transform," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 5, pp. 1946–1958, 2016.

Research Article

A Selective Mirrored Task Based Fault Tolerance Mechanism for Big Data Application Using Cloud

Hao Wu ¹, Qinggeng Jin ², Chenghua Zhang,¹ and He Guo¹

¹*School of Software Technology, Dalian University of Technology, Dalian, Liaoning, China*

²*Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Guangxi University for Nationalities, Nanning, Guangxi, China*

Correspondence should be addressed to Qinggeng Jin; jinqinggeng@aliyun.com

Received 6 December 2018; Accepted 29 January 2019; Published 26 February 2019

Guest Editor: Salimur Choudhury

Copyright © 2019 Hao Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the wide deployment of cloud computing in big data processing and the growing scale of big data application, managing reliability of resources becomes a critical issue. Unfortunately, due to the highly intricate directed-acyclic-graph (DAG) based application and the flexible usage of processors (virtual machines) in cloud platform, the existing fault tolerant approaches are inefficient to strike a balance between the parallelism and the topology of the DAG-based application while using the processors, which causes a longer makespan for an application and consumes more processor time (computation cost). To address these issues, this paper presents a novel fault tolerant framework named Fault Tolerance Algorithm using Selective Mirrored Tasks Method (FAUSIT) for the fault tolerance of running a big data application on cloud. First, we provide comprehensive theoretical analyses on how to improve the performance of fault tolerance for running a single task on a processor. Second, considering the balance between the parallelism and the topology of an application, we present a selective mirrored task method. Finally, by employing the selective mirrored task method, the FAUSIT is designed to improve the fault tolerance for DAG based application and incorporates two important objects: minimizing the makespan and the computation cost. Our solution approach is evaluated through rigorous performance evaluation study using real-word workflows, and the results show that the proposed FAUSIT approach outperforms existing algorithms in terms of makespan and computation cost.

1. Introduction

Recent years have witnessed that the big data analysis grows dramatically, and the related applications have been used everywhere in both academia [1] and industry [2]. There is no denying that the developmental cloud platform technologies played a key role in this process; the plenty of processors in cloud make sure that the scholars can handle the significant large-scale big data processing [3–6]. However, due to voltage fluctuation, cosmic rays, thermal changes, or variability in manufacturing, the chip level soft errors and the physical flaws are inevitable for a processor even when the probability of that is extremely low [7, 8]. Furthermore, the abundant use of processors by a big data application induces that the probability cannot be ignored.

Big data and big data analysis have been proposed for describing data sets as analytical technologies in large-scale

complex programs, which need to be analyzed with advanced analytical methods [9, 10]. No matter whether the big data applications are developed for commercial purposes or scientific researches, most of these applications require significant amount of computing resources, such as market structure analysis, customer trade analysis, environmental research, and astrophysics data processing. Motivated by the reasonable price, rapid elasticity, and shifting responsibility of maintenance, backups, and management to cloud providers, more and more big data applications have been deployed to clouds, such as EC2 [11], Google Cloud [12], and Microsoft Azure [13].

The clouds provide unlimited computing resources (from the user's point of view) including CPU resources and GPU resources. The on-demand recourses facilitate users to choose apposite processors (with CPU or GPU resources) for executing their big data applications efficiently [14]. However,

according to [15], there are many factors such as voltage fluctuation, cosmic rays, thermal changes, or variability in manufacturing, which cause the processors (both CPU and GPU) to be more vulnerable. Indeed, the probability of fault rate is really low. But, as already noted, plenty of processors participate in computing the big data application, and the computing time of each processor may be very long. These potential factors will lead to an exponential growth for the fault rate in a cycle of running a big data application.

The failures caused by processors are disastrous for a big data application which is deployed on the processors; i.e., once a fault occurs on any processor, the application will have to be executed all over again, and that will waste lots of monetary cost and time cost. Thus, improving the robustness (or reducing the fault rate) for running a big data application has attracted many scholars' attention; many studies have been exploring this problem. According to the literature, these studies are classified into two main categories: resolving the problem in hardware or software level.

From the hardware level's perspective, improving the mean time between failures (MTBF) [16] of the processors is the key to reduce the fault rate for running a big data application. As everyone knows, it is impossible to eradicate the failures in a processor. Apart from lifting tape-out technology, the [15] proposed an adaptive low-overhead fault tolerance mechanism for many-core processor, which treats fault tolerance as a device that can be configured and used by application when high reliability is needed. Although [15] improves the MTBF, but the risk of occurring failures remains. Therefore, the other scholars seek solutions in software level.

From the software level's perspective, the developed check-point technology [17] makes sure that a big data application can be completed under any size of MTBF. Thus, many check-point strategies are proposed to resolve this problem such as [18, 19]. These strategies only pay a little extra cost, but they can complete the applications under any size of MTBF. As a result, almost all of cloud platforms provide check-point interface for users. However, these strategies did not consider the data dependencies between the processors, which make it inappropriate to big data application running on cloud.

In order to handle the DAG based applications, the copy task based method (also known as primary backup based method) is proposed to resolve the problem. In [20], Qin and Jiang proposed an algorithm eFRD to enable a systems fault tolerance and maximize its reliability. On the basis of eFRD, Zhu et al. [21] developed an approach for task allocation and message transmission to ensure faults can be tolerated during the workflow execution. But the task based methods will make the makespan have a long delay due to the fact that the backup tasks are not starting with the original tasks.

To the best of our knowledge, for the big data application, there are no proper check-point strategies which can handle the data dependencies among the processors simultaneously. In this paper, we propose a novel check-point strategy for big data applications, which considers the effect of the data communications. The proposed strategy adopts high level failure model to resolve the problem, which makes it closer

to practice. Meanwhile, the subsequent effect caused by data dependencies after a failure is also considered in our strategy.

The main contributions of this paper are as follows:

- (i) A selective mirrored task method is proposed for the fault tolerance of the key subtasks in a DAG based application.
- (ii) On the basis of the selective mirrored task method, a novel check-point framework is proposed to resolve the fault tolerance for a big data application running on cloud; the framework is named Fault Tolerance Algorithm using Selective Mirrored Tasks Method (FAUSIT).
- (iii) A thorough performance analysis is conducted for FAUSIT through experiments on randomly generated test big data application as well as real-world application traces.

The rest of this paper is structured as follows. The related work is summarized in Section 2. Section 3 introduces the models of the big data application, the cloud platform, and the MTBF and then formally defines the problem the paper is addressing. Section 4 presents the novel check-point strategy for the big data application running on cloud. Section 5 conducts extensive experiments to evaluate the performance of our algorithm. Section 6 concludes the paper with summary and future directions.

2. Related Work

Over the last two decades, owing to the increasing scale of the big data applications [22], the fault tolerance for big data application is becoming more and more crucial. Considerable research has been explored by scholars. In this section, we summarize the research in terms of theoretic methods and heuristic methods.

A number of theoretic methods have been explored by scholars. For a task running on a processor, Young [21] proposed a first order failure model and figured out an approximation to the optimum check-point interval which is $\varphi_{opt} = \sqrt{2\delta M}$, where δ is the time to write a check-point file, M is the mean time between failures for the processor, and φ_{opt} is the optimum computation interval between writing check-point files. However, the model in Young [21] will never have more than a single failure in any given computation interval. This assumption goes against some practical situations; for instance, there may be more than one failure occurring in a computation interval.

Due to the downside of the model in Yang [21], Daly [22] proposed a higher order failure model to estimate the optimum check-point interval. The model of Daly [22] assumes that there may be more than one failure occurring in a computation interval, which is closer to the realistic situation. The optimum computation interval figured out by Daly [22] is

$$\varphi_{opt} = \begin{cases} \sqrt{2\delta M} \left[1 + \frac{1}{3} \left(\frac{\delta}{2M} \right)^{1/2} + \frac{1}{9} \left(\frac{\delta}{2M} \right) \right] - \delta & \text{if } \delta < 2M, \\ M & \text{if } \delta \geq 2M. \end{cases} \quad (1)$$

$$\varphi_{opt} = \begin{cases} \sqrt{2\delta M} - \delta & \text{if } \delta < \frac{1}{2}M, \\ M & \text{if } \delta \geq \frac{1}{2}M. \end{cases} \quad (2)$$

which is a good rule of thumb for most practical systems.

However, the models in both Yang [22] and Daly [22] are aimed at one task running on a processor, which are not applicable to DAG based application running on cloud for the following reasons. First, there are many subtasks running on different processors; the completion time of each subtask may have influence on the successive subtasks. Second, the importance of each subtask in a DAG (a DAG based application) is different; for instance, the subtasks on the critical path of the DAG are more important than the others. Therefore, some scholars proposed heuristic methods aiming at DAG based application running on cloud.

Aiming to resolve the fault tolerance for DAG based application running on cloud, Zhu [23] and Qin [24] proposed copy task based methods. In general, the basic idea of copy task based methods is running an identical copy of each subtasks on different processors, the subtasks and their copies can be mutually excluded in time. However, these approaches assume that tasks are independent of one other, which cannot meet the needs of real-time systems where tasks have precedence constraints. In [24], for given two tasks, the authors defined the necessary conditions for their backup copies to safely overlap in time with each other and proposed a new overlapping scheme named eFRD (efficient fault-tolerant reliability-driven algorithm), which can tolerate processors failures in a heterogeneous system with fully connected network.

In Zhu [23], on the basis of Qin [24], the authors established a real-time application fault-tolerant model that extends the traditional copy task based model by incorporating the cloud characteristics. Based on this model, the authors developed approaches for subtask allocation and message transmission to ensure faults can be tolerated during the application execution and proposed a dynamic fault tolerant scheduling algorithm, named FASTER (fault tolerant scheduling algorithm for real-time scientific workflow). The experiment results show that the FASTER is better than eFRD [24].

Unfortunately, the disadvantage of copy task based methods including Zhu [23] and Qin [24] is very conspicuous. First, the copy of each subtask may consume more resources on the cloud, which makes them uneconomical. Second, the copy of each subtask will be executed only when the original subtask failed; this process will waste a lot of time, and it will be even worse due to the DAG based application; i.e., the deadline of the application will be not guaranteed in most of cases if the deadline is near to the critical path.

Thus, in this paper, we will combine the advantage of theoretic methods and heuristic methods to propose a novel fault tolerance algorithm for a big data application running on cloud.

TABLE 1: Cost optimization factors.

Symbols	Definitions
$T = G(V, E)$	a scientific application
$G(V, E)$	DAG of the tasks in T
CP	the critical path of $G(V, E)$
τ_i	the i -th subtask in V
w_i	the weight of i -th tasks in V
V	the set of τ_i in T
$e_{i,j}$	data dependence from τ_i to τ_j
E	the set of $e_{i,j}$
δ	the time to make a check-point file
φ_{opt}	the optimal check-point interval
$T_w(\varphi)$	the practical completion time for the task
R	the time to read a check-point file
X	the time before a failure in an interval
T_c	the computation time for a task
S	a schedule which maps the tasks to processors
$wt(\tau_i)$	the weight of τ_i
$pred(\tau_i)$	the predecessor subtasks set of τ_i in $G(V, E)$
$succ(\tau_i)$	the successors subtasks set of τ_i in $G(V, E)$
$succP(\tau_i)$	the next subtask τ_j on the same processor
$ft(\tau_i)$	the finish time of τ_i on the S
$estS(\tau_i)$	the earliest start time of τ_i on the S
$lftS(\tau_i)$	the latest finish time of τ_i on the S
$slackT(\tau_i)$	the slack time of τ_i on the S
$\overline{slackT}(\tau_i)$	the indirect slack time of τ_i on the S
$\alpha(\tau_i)$	denotes the quantitative importance of τ_i
$\bar{\alpha}$	The threshold of α
$KeyT$	The set of key subtasks
$Ms(T)$	The makespan of the application T

3. Models and Formulation

In this section, we introduce the models of big data application, cloud platform, and the failure model then formally define the problem this paper is addressing. To improve the readability, we sum up the main notations used throughout this paper in Table 1.

3.1. Big Data Application Model. The model of a big data application T is denoted by $G(V, E)$, where $G(V, E)$ represents a DAG. Besides, we use T_c to denote the execution time of the critical path in $G(V, E)$. Each node $n_i \in V$ represents a subtask τ_i ($1 \leq i \leq v$) of T , and v is the total number of subtasks in T . W is the set of weights, in which each wire presents the execution time of a subtask n_i on a VM. E is the set of edges in $G(V, E)$, and an edge (n_i, n_j) represents the dependence between n_i and n_j ; i.e., a task can only start after all its predecessors have been completed.

Figure 1 shows an example of DAG for a big data workflow, consisting of twelve tasks from τ_1 to τ_{12} . The DAG vertices related to tasks in T are represented by circles, while the directed edges denote the data dependencies among the tasks.

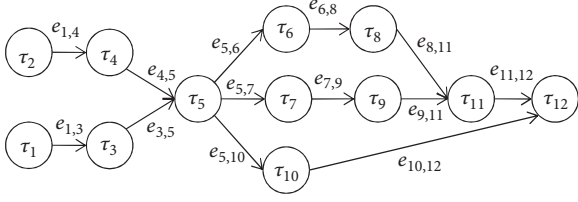


FIGURE 1: A DAG based application.

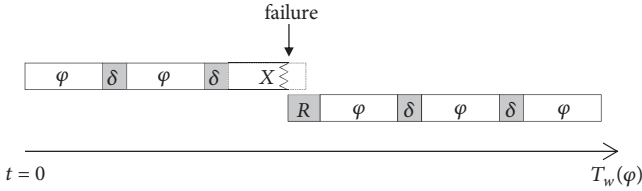


FIGURE 2: An example of a failure.

3.2. Cloud Platform Model. A cloud platform C is modeled as a set of processors $\{P_1, P_2, \dots, P_N\}$ and N is the total number of processors on the cloud. We use N to denote the number of processors rented by users for executing an application. Actually, the N is much greater than the mount which the user need. In general, to reduce the cost (monetary cost or time cost), the users apply proper schedule arithmetic to deploy their big data applications on cloud. But most of schedule algorithms did not consider the failure in each processor, which may consume extra cost (monetary cost and time cost).

3.3. The Check-Point Model. The check-point technology has been used on cloud for years, which makes application complete successfully in the shortest amount of time. In general, the check-point model is defined as bellow: ideally, the time to complete a task on a processor is denoted by T_c ; we use φ to denote the check-point interval. After each computation interval (φ), the processor makes a backup for the current status of the system, and the time consumed by this process is represented by δ . If a failure occurs, the time consumed to read the latest backup and restart the computation is denoted by R . Finally, we use $T_w(\varphi)$ to denote the practical completion time for the task running on a processor while the check-point interval is φ .

Referring to Figure 2, the ideal completion time for the task is $T_c = 5\varphi$. Actually, there is a failure occurring after X time in the third interval, and it takes the processor R time to restart the third interval. At last, the practical time consumed by the tasks is $T_w(\varphi) = 5\varphi + R + X + 4\delta$.

3.4. The Failure Model. For a given MTBF (mean time between failures) which is denoted by M , according to [21], the life distribution model for mechanical and electrical equipment is described by an exponential model. Thus, the probability density function is

$$f(t) = \frac{1}{M} e^{-t/M}. \quad (3)$$

Then, the probability of a failures occurring before time Δt for a processor is represented by a cumulative distribution function

$$P(t \leq \Delta t) = \int_{\Delta t}^0 \frac{1}{M} e^{-t/M} dt = 1 - e^{-\Delta t/M}. \quad (4)$$

Obviously, the probability of successfully completing for a time Δt without a failure is

$$P(t > \Delta t) = 1 - P(t \leq \Delta t) = e^{-\Delta t/M}. \quad (5)$$

We use T_c to denote the computation time for a subtask running on a processor; the φ denotes the compute interval between two check-points. Then, the average number of attempts (represented by $No.a$) needed to complete T_c is

$$No.a = \frac{T_c/\varphi}{P(t > \Delta t)} = \frac{T_c e^{\Delta t/M}}{\varphi}. \quad (6)$$

Therefore, the total number of failures during Δt is the number of attempts minus the number of successes.

$$n(\Delta t) = \frac{T_c e^{\Delta t/M}}{\varphi} - \frac{T_c}{\varphi} = \frac{T_c}{\varphi} (e^{\Delta t/M} - 1). \quad (7)$$

Notice that this assumes that we will never have more than a single failure in any given computation interval. Obviously, this assumption is relaxed in a real-life scenario. Thus, in [22], the scholar presented a multiple failures model.

$$\begin{aligned} n(\varphi) &= \frac{(T_c - \delta + \delta T_c/\varphi)}{M - \{E(\varphi + \delta) + R\} P(\varphi) - E(R + \varphi + \delta) [1 - P(\varphi)]}, \end{aligned} \quad (8)$$

where

$$\begin{aligned} E(\varphi + \delta) &= M + \frac{\varphi + \delta}{1 - e^{-(\varphi + \delta)/M}} \\ E(R + \varphi + \delta) &= M + \frac{R + \varphi + \delta}{1 - e^{-(\varphi + \delta)/M}} \\ P(\varphi) &= e^{-(R + \varphi + \delta)/M} \end{aligned} \quad (9)$$

The derivation process of Formula (8) is detailed in [22]; we will not repeat the process in this paper. In this paper, we will take Formula (8) as the failure model in our framework for two reasons; first, this could only provide the MTBF (M , mean time between failures) determined by statistics [19, 25] and, second, this model is closer to reality than the other model in [21].

3.5. Definitions. In order to make readers have a better understanding of this algorithm, we make some definitions first. For a given big data application $T = G(V, E)$ and a schedule S , we define the following terms.

3.5.1. Schedule (S). A schedule S is a map from the subtasks in $G(V, E)$ to the processors on the cloud; meanwhile, the start time and the finish time of each subtask have been figured out. In general, the S is determined by a static schedule algorithm, such as HEFT [26] and MSMD [27].

3.5.2. *Weight* ($wt(\tau_i)$). The $wt(\tau_i)$ is the weight (execution time) of τ_i running on a processor.

3.5.3. *Predecessors* ($pred(\tau_i)$). For a subtask τ_i , its predecessors subtask set is defined as below:

$$pred(\tau_i) = \{\tau_j \mid \tau_j \in V \wedge (\tau_j, \tau_i) \in E\}. \quad (10)$$

3.5.4. *Successors* ($succ(\tau_i)$). For a subtask τ_i , its successors subtask set is defined as below:

$$succ(\tau_i) = \{\tau_j \mid \tau_j \in V \wedge (\tau_i, \tau_j) \in E\}. \quad (11)$$

3.5.5. *Successor on the Processor* ($SuccP(\tau_i)$). For a given schedule S , the $SuccP(\tau_i)$ of τ_i is the set of the next subtask deployed on the same processor.

3.5.6. *Finish Time* ($ft(\tau_i)$). The $ft(\tau_i)$ denotes the finish time of τ_i on the schedule S .

3.5.7. *Earliest Start Time on the Schedule* ($estS(\tau_i)$). For a given schedule S , the start time of τ_i is the $estS(\tau_i)$. It should be noted that the $estS(\tau_i)$ is different with the traditional earliest start time in DAG.

3.5.8. *Latest Finish Time on the Schedule* ($lftS(\tau_i)$). For a given schedule S , the $lftS(\tau_i)$ of τ_i is defined below:

$$lftS(\tau_i) = \begin{cases} ft(\tau_i), & \text{if } succ(\tau_i) \text{ and } SuccP(\tau_i) \text{ are empty,} \\ \min_{\tau_j \in succ(\tau_i) \vee SuccP(\tau_i)} \{estS(\tau_j)\}, & \text{otherwise.} \end{cases} \quad (12)$$

Same with $estS(\tau_i)$, the $lftS(\tau_i)$ is different with the traditional earliest start time in DAG.

3.5.9. *Slack Time* ($slackT(\tau_i)$). For a given schedule S , the $slackT(\tau_i)$ of τ_i is defined as follows:

$$slackT(\tau_i) = lftS(\tau_i) - estS(\tau_i) - weight(\tau_i). \quad (13)$$

3.5.10. *Indirect Slack Time* ($\overline{slackT}(\tau_i)$). For a given schedule S , the $\overline{slackT}(\tau_i)$ of τ_i is defined as follows:

$$\overline{slackT}(\tau_i) = \begin{cases} slackT(\tau_i), & \text{if } slackT(\tau_i) \neq 0, \\ \min \left\{ \frac{wt(\tau_i)}{wf(\tau_i) + wt(\tau_j)} \overline{slackT}(\tau_j) \right\}, & \text{else if } succ(\tau_i) \vee SuccP(\tau_i) = \emptyset, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

The $\overline{slackT}(\tau_i)$ denotes that the slack time of a subtask can be shared by its predecessors.

3.5.11. *Importance* ($\alpha(\tau_i)$). The $\alpha(\tau_i)$ denotes the quantitative importance of the subtask τ_i in the schedule S . The $\alpha(\tau_i)$ is calculated by

$$\alpha(\tau_i) = \frac{\overline{slackT}(\tau_i)}{wt(\tau_i)}. \quad (15)$$

3.5.12. *Threshold of Importance* ($\bar{\alpha}$). The threshold $\bar{\alpha}$ denotes whether a subtask is a key subtask; i.e., if $\alpha(\tau_i) < \bar{\alpha}$, then the subtask τ_i is a key subtask.

3.5.13. *The Set of Key Subtask* ($KeyT$). The signal $KeyT$ denotes the set of key subtask for a given schedule S .

3.6. *Problem Formalization*. The ultimate objective of this work is to provide a high-performance fault tolerance mechanism and make sure that the proposed fault tolerance mechanism will consume less computation cost and makespan. The computation cost represents the processor time consumed by all the subtasks in T ; thus, the object of minimizing computation cost is defined by

$$\text{Minimize } \sum_{\tau_i \in T} T_w(\tau_i). \quad (16)$$

The makespan can be defined as the overall time to execute the whole workflow by considering the finish time of the last successfully completed task. For an application T , this object is denoted by

$$\text{Minimize } Ms(T). \quad (17)$$

4. The Fault Tolerance Algorithm

In this section, we first discuss the basic idea of our algorithm for the fault tolerance of running big data application on

cloud. Then, on the basis of the idea, we will propose the fault tolerance algorithm using Selective Mirrored Tasks Method.

4.1. The Basic Idea. As show in Section 2, the theoretic methods which are devoted to find the optimal φ_{opt} are not applicable to the DAG based application, even if the φ_{opt} they have determined is very accurate for one task running on a processor. Besides, the heuristic methods based on the copy task will waste a lot of extra resource, and the completion time of the application may be delayed by much more time.

To find a better solution, we will integrate the advantages of theoretic methods and heuristic methods to propose a high-performance and economical algorithm for big data application. The check-point method with an optimal computation interval φ_{opt} is a dominant and economical method for one task running on a processor; thus, the check-point mechanism is the major means in our approach. Furthermore, owing to the parallelism and the dependencies in a DAG based application, the importance of each subtask is different; i.e., the subtasks on the critical path are more important than the others. The fault tolerance performance of these subtasks which adopt check-point method is insufficient, because the completion time of an application depends to a great extent on the completion time of these subtasks. Therefore, for the important subtasks, we will improve the fault tolerance performance of an application by introducing the task copy based method. In the task copy based methods [24], the original task and the copy do not start at the same time; to reduce the completion time, the original task and the copy will start at the same time.

In summary, the basic idea is as follows. First, identify the important subtasks in the DAG based application, which are named as key subtasks in the rest of this article. Then, apply the task copy based methods to the key subtasks; meanwhile, all the subtasks will employ the check-point technology to improve the fault tolerance performance, but the key subtasks and the normal subtasks will use different optimal computation interval φ_{opt} , the details of which will be described in the following sections.

It should be noted that our fault tolerance algorithm will not schedule the subtasks on the processors; we just provide a fault tolerance mechanism based on the existing static scheduler algorithm (such as HEFT and MSMD) to make sure that the application can be completed with the minimum of time.

4.2. Fault Tolerance Algorithm Using Selective Mirrored Tasks Method (FAUSIT). Based on the basic idea above, we propose the FAUSIT to improve the fault tolerance for executing a large-scale big data application; the pseudocode of the FAUSIT is listed in Algorithm 1.

As shown in Algorithm 1, the input of FAUSIT is a map from subtasks to processors which is determined by a static scheduler algorithm and the output is fault tolerance operation determined by FAUSIT. The function **DetermineKeyTasks()** in Line (1) is to find the key subtasks according to the schedule S and the application T . Then, the

Input: a schedule S of an application T
Output: a fault tolerance operation F
 (1) **DetermineKeyTasks**(S, T) \rightarrow $KeyT$
 (2) **DeployKeyTasks**($KeyT$) \rightarrow F
 (3) **return** F

ALGORITHM 1: Fault Tolerance Algorithm using Selective Mirrored Tasks Method (FAUSIT).

function **DeployKeyTasks()** in Line (2) deploys the mirrored subtasks and determine the proper φ_{opt} for the subtasks.

In the following content in this subsection, we will expound the two functions in FAUSIT.

4.2.1. Function of DetermineKeyTasks(). The function of **DetermineKeyTasks()** is to determine the key subtasks in the DAG based application. In order to make readers have a better understanding of this function, we need to expound the key subtask and the indirect slack time clearly.

Definition 1. Key subtask: in a schedule S , the finish time of a subtask has influence on the start time of its successors; if the influence exceeds a threshold, we define the subtask is a key subtask.

The existence of the key subtasks is very meaningful to our FAUSIT algorithm. For a given schedule S , in the ideal case, each subtask as well as the application will be finished in accordance with the S . In practice, a subtask may fail when it has executed for a certain time; then, the processor will load the latest check-point files for continuation. At this point, the delay produced by the failure subtask may affect the successors. For the subtasks which have sufficient slack time, the start time of the successors is free from the failed subtask. On the contrary, if the failed subtask has little slack time, it will affect the start time of the successors undoubtedly. Given all that, we need to deal with the key subtasks which has little slack time.

Definition 2. Indirect slack time: for two subtasks τ_i and τ_j , τ_j is the successor of τ_i , if τ_j has slack time (defined in Section 3.5.9), the slack time can be shared by τ_i , and the shared slack time is indirect slack time for τ_i .

The indirect slack time is a useful parameter in our FAUSIT algorithm, the existence of which will make the FAUSIT save a lot of time (makespan of the application) and cost. For a given schedule S , a subtask may have sufficient slack time which can be shared by predecessors. Thus, the predecessors may have enough slack time to deal with failures; then, the completion time of the predecessors and the subtask will not delay the makespan of the application. Indeed, the indirect slack time is the key parameter to determine whether a subtask is a key subtask. Moreover, the indirect slack time reduces the count of the key subtasks in a big data application, which will save a lot of cost, because the key subtask will apply mirrored task method.

Input: a schedule S of an application T , $\bar{\alpha}$.
Output: the key subtask set $KeyT$.
(1) Determine the $ft()$, $estS()$ and the $lftS()$ of the subtasks in T according to S .
(2) Determine the $slackT()$ of each subtask.
(3) Determine the $\overline{slackT}()$ of the subtasks by recursion.
(4) Determine the set of key subtasks according to $\alpha \rightarrow KeyT$
(5) **return** $KeyT$.

ALGORITHM 2: DetermineKeyTasks().

The pseudocode for function **DetermineKeyTasks()** is shown in Algorithm 2. The input of **DetermineKeyTasks()** is a schedule S of an application T and the threshold $\bar{\alpha}$; the output is the set of key subtasks. First, in Line (1), the $ft()$, $estS()$ and the $lftS()$ of the subtasks are determined according to Sections 3.5.6 and 3.5.7 and Formula (12). Then, the $slackT()$ of each subtask is figured out by Formula (13) in Line (2). Line (3) determines the $\overline{slackT}()$ of the subtasks by recursion. Finally, the set of key subtasks is determined according to the threshold α .

Table 2 shows the process of **DetermineKeyTasks()**. When the $\bar{\alpha} = 0.15$, Figure 3 shows the key subtasks which shall adopt the mirrored task method to improve the performance of fault tolerance.

It should be noted that the threshold $\bar{\alpha}$ is given by the users, which is related to the makespan of the application; i.e., the higher $\bar{\alpha}$ leads to more key subtasks. Then, the makespan of the application is shorter. On the contrary, the smaller $\bar{\alpha}$ will lead to a longer makespan.

4.2.2. Function of DeployKeyTasks(). The function of **DeployKeyTasks()** is to deal with the key subtasks, which minimizes the makespan of the application to the least extent. The main operation of **DeployKeyTasks()** is using mirrored task method; in order to make readers have a better understanding of this function, we need to expound the mirrored task method first.

The mirrored task method is to deploy a copy of a key subtask on another processor; the original subtask is denoted by τ_i^P and the copy of the subtask is denoted by τ_i^C . The τ_i^P and the τ_i^C start at the same time, and the check-point interval of them is $2\varphi_{opt}$ (the φ_{opt} is determined by [22]). The distinction between the τ_i^P and the τ_i^C is that the first check-point interval of τ_i^P is φ_{opt} ; meanwhile, the first check-point interval of τ_i^C is $2\varphi_{opt}$. Obviously, once a failure occurs in one of the processors, the interlaced check-point interval of the two same subtasks makes sure that the time delayed by dealing with the failure is W (the time to read a check-point file).

Figure 4 shows an example of the mirrored task method. The Figure 4(a) displays the ideal situation of the mirrored task method; i.e., there are no failures happen in both P_k and P_l , and the finish time is $4\varphi + 2\delta$. In Figure 4(b), there is only one failure happening on P_l in the second check-point

interval φ^2 . First, the processor P_l reads the latest check-point file named δ_k^1 . Then, with time goes by, the processor P_l will immediately load the latest check-point file δ_k^3 when it generates. Thus, the finish time is $4\varphi + 2\delta + W$. Figure 4(c) illustrates the worst case; both P_k and P_l have a failure in the same interval φ^2 , the two processors will have to load the latest check-point file δ_k^1 . Thus, the finish time is $4\varphi + 2\delta + W + X$.

Obviously, the mirrored task method is far better than the traditional copy task based method, since the copy task will start only when the original task failed, and it will waste a lot of time. Moreover, the mirrored task method is also better than the method in [22], since the probability of the worst case is far less than the probability of the occurrence for one failure in a processor.

The pseudocode for function **DeployKeyTasks()** is displayed in Algorithm 3. The loop in Line (1) makes sure that all the key subtasks can be deploy on the processors. The applied processors should be used first when deploying the key subtasks; the loop in Line (2) illustrates this constraint.

Line (3) makes sure that the key subtasks τ_i^B have no overlapping with other tasks on P_j . Lines (4)-(5) deploy τ_i^B on P_j . If all the applied processors have no idle interval for τ_i^B (Line (8)), we will have to apply a new processor and deploy τ_i^B on it; then, we put the new processor into the set of applied processors (Lines (9)-(11)). At last, we deploy the check-point interval (φ_{opt} or $2\varphi_{opt}$) to the processors (Line (14)) and save these operations in T (Line (15)).

It should be noted that the overlapping in Line (3) is not just the computation time of the τ_i^B ; we also consider the delayed time which may be caused by failures. In order to avoid the overlap caused by the delayed τ_i^B , we use the $1.3w_i$ as the execution time of τ_i^B to determine whether an overlap happen, since the $1.3w_i$ is much greater than a delayed τ_i^B .

4.3. The Feasibility Study of FAUSIT. The operations to the processors in our FAUSIT are complex, such as the different check-point interval and the mirrored subtasks; the readers may doubt the feasibility of FAUSIT. Thus, we will explain the feasibility study of FAUSIT in this subsection.

According to [15], Google Cloud provides the *gcloud* suit for users to operate the processors (virtual machine) remotely. The operations of *gcloud* include (but not limited to) applying processors, releasing processors, check-point,

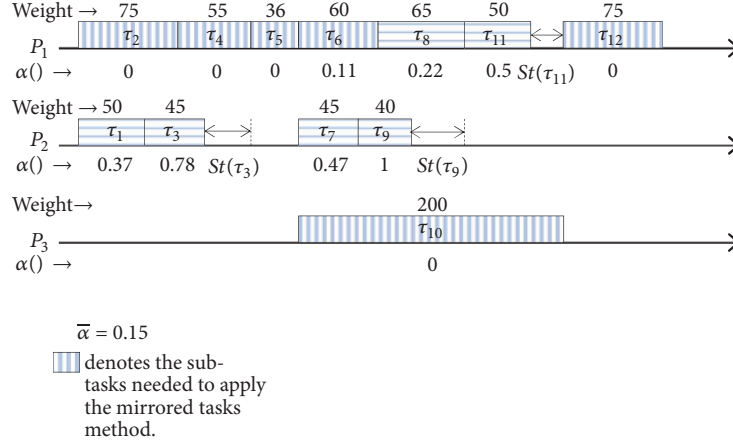
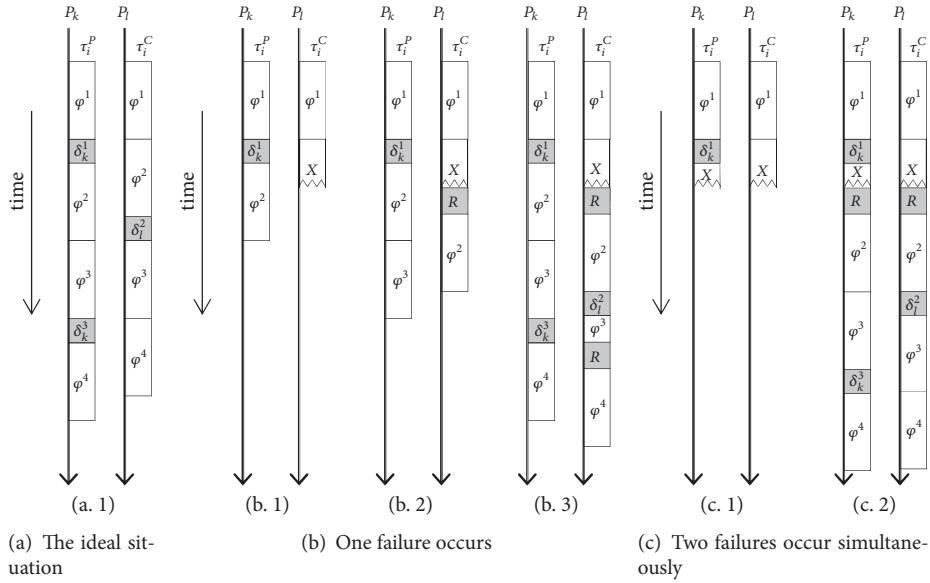
FIGURE 3: An example of $\bar{\alpha} = 0.15$.

FIGURE 4: An example of mirrored tasks method.

TABLE 2: An example of **DetermineKeyTasks()**.

Subtasks	Weight	$lft()$	$est()$	$St()$	$St'()$	$\alpha()$
τ_1	50	0	50	0	18.4	0.37
τ_2	75	0	75	0	0	0
τ_3	45	50	130	35	35	0.78
τ_4	55	75	130	0	0	0
τ_5	36	130	166	0	0	0
τ_6	60	166	226	0	6.8	0.11
τ_7	45	166	211	0	21.2	0.47
τ_8	65	226	291	0	14.1	0.22
τ_9	40	211	291	40	40	1
τ_{10}	200	166	366	0	0	0
τ_{11}	50	291	366	25	25	0.5
τ_{12}	75	366	441	0	0	0

Input: the key subtask set yT .
Output: a fault tolerance operation F

```

(1) for  $\tau_i^B$  in  $KeyT$  do
(2)   for  $P_j$  in the processors which have been applied do
(3)     if  $\tau_i^B$  has no overlapping with other tasks on  $P_j$  then
(4)       Deploy the  $\tau_i^B$  on  $P_j$ 
(5)     End for in Line (2).
(6)   end if
(7) end for
(8) if  $\tau_i^B$  has not be deployed on the processor. then
(9)   Apply a new processor.
(10)  Deploy  $\tau_i^B$  on the new processor.
(11)  Put the new processor in the set of processors.
(12) end if
(13) end for
(14) Deploy the check-point interval to the processors.
(15) Save the operations in  $F$ .
(16) return  $F$ 

```

ALGORITHM 3: **DeployKeyTasks()**.

TABLE 3: The Characteristics of the Big Data Application.

Workflows	Task Count Range	Edge Count	Avg. Run Time of Task (Sec)	Deadlines Range
Epigenomics	100	322	3856.51	1.0CP
	200	644		
		
	1000	3228		

and loading check-point files. These operations in *gcloud* can make user implement the FAUSIT easily.

5. Empirical Evaluations

The purpose of the experiments is to evaluate the performance of the developed FAUSIT algorithm. We evaluate the fault tolerant of FAUSIT by comparing it with two other algorithms published in the literature. They are FASTER algorithm [23] and the method in [22]. The main differences of these algorithms to FAUSIT and their uses are briefly described below.

- (i) FASTER: a novel copy task based fault tolerant scheduling algorithm. On the basis of copy task based method, the FASTER adjust the resources dynamically.
- (ii) The method in [22]: it is a theoretic method which provides an optimal check-point interval φ_{opt} .

5.1. Experiment Settings. The DAG based big data applications we used for the evaluation are obtained from the DAG based applications benchmark provided by Pegasus WorkflowGenerator [28]. We use the largest application from the benchmark, i.e., Epigenomics, since the bigger application

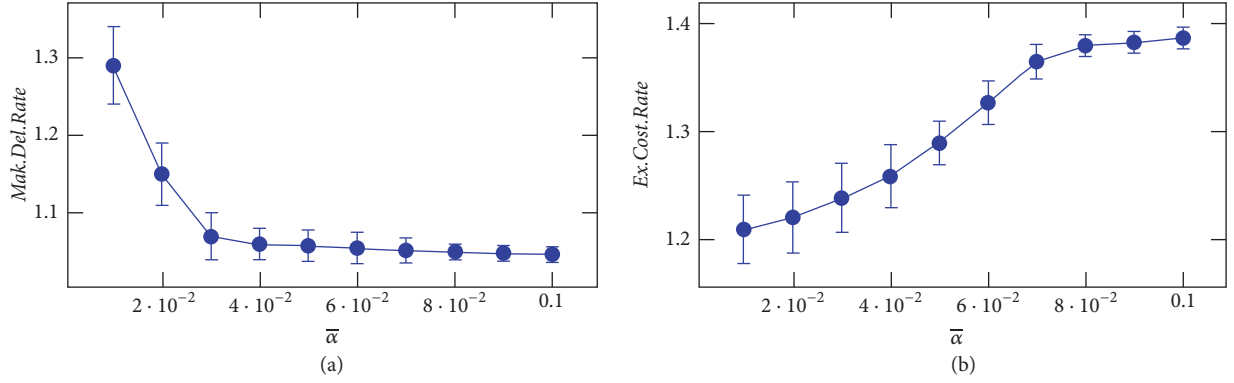
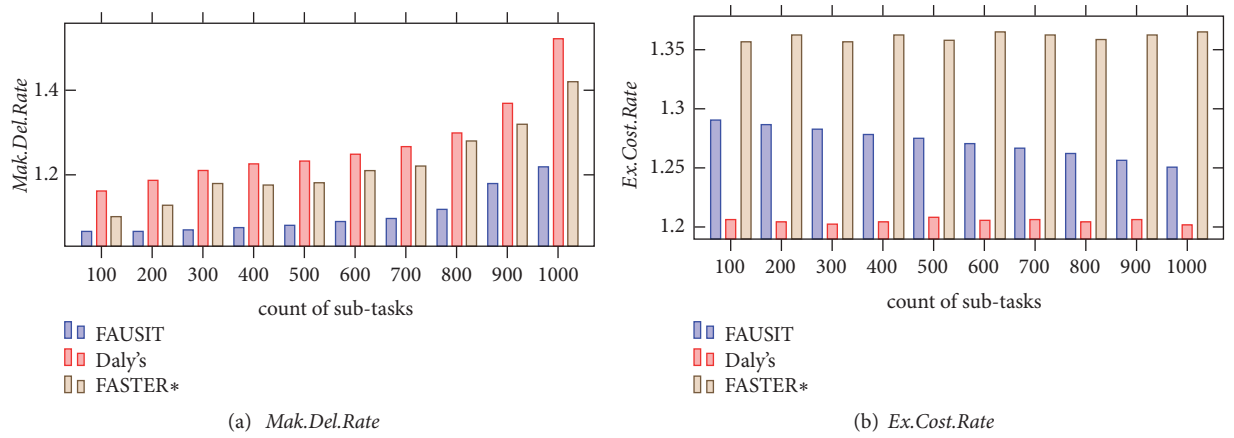
is more sensitive to failures. The detailed characteristics of the benchmark applications can be found in [29, 30]. In our experiments, the number of subtasks in an application is ranging from 100 to 1000. Since the benchmark does not assign deadlines for each application, we need to specify the deadlines; we assign a deadline for the applications: it is 1.0CP. Table 3 gives the characteristics of these applications including the count of tasks and edges, average task execution time and the deadlines.

Because FAUSIT does not schedule the subtasks on the processors, we hire a high-performance schedule algorithm (MSMD) to determine the map from subtasks to the processors. MSMD is a novel static schedule algorithm to reduce the cost and the makespan for an application. On the basis of MSMD, we use FAUSIT to improve the fault tolerant of an application.

5.2. Evaluation Criteria. The main objectives of the FAUSIT are to find the optimal fault tolerant mechanism for a big data application running on cloud. The first criterion to evaluate the performance of a fault tolerant mechanism is how much time is delayed to complete the application, i.e., the makespan to finish the application. We introduce the concept of *makespan delay rate* to indicate how much extra time consumed by the fault tolerant mechanism. It is defined as follows:

$$Mak.Del.Rate = \frac{\text{The practical makespan}}{\text{The ideal makespan}}. \quad (18)$$

where the ideal makespan represents the completion time for running an application without fault tolerant mechanism and any failures, and the practical makespan is the completion time consumed on practical system which has a fault tolerant mechanism and the probability of failures.

FIGURE 5: (a) The *Mak.Del.Rate* and (b) the *Ex.Cost.Rate* of $\bar{\alpha}$.FIGURE 6: The result of *Mak.Del.Rate* and *Ex.Cost.Rate*.

The second goal of the FARSIT is to minimize the extra cost consumed by the fault tolerant mechanism. Undoubtedly, to improve the performance of fault tolerant, any fault tolerant mechanisms will have to consume extra computation time for an application. Thus, the extra computation time is a key criterion to evaluate the performance of fault tolerant mechanism. Therefore, we define extra cost rate for the evaluation:

$$Ex.Cost.Rate = \frac{\text{The practical processors time}}{\text{The ideal processors time}}. \quad (19)$$

where the practical processors time denotes the time consumed by the processors to running an application for a practical system with failures, The ideal processors time is the sum of the w_i in $G(V, E)$ for an application.

5.3. The Optimal α for FAUSIT. Before comparing with the other algorithms, we need to determine the optimal $\bar{\alpha}$ for FAUSIT first. Due to the optimal $\bar{\alpha}$ is an experimental parameter, we have to figure it out by experiments.

We test the $\bar{\alpha}$ by the Epigenomics application with 1000 subtasks for 10 times and make the $T_d = 1.0T_c$; the results are shown in Figures 5(a) and 5(b). In Figure 5(a), the *Mak.Del.Rate* becomes lower along with the increased; i.e., a larger $\bar{\alpha}$ will lead to a shorter makespan for an application.

On the contrary, Figure 5(b) shows that the larger $\bar{\alpha}$ will lead to more cost for an application.

Through a comprehensive analysis of Figures 5(a) and 5(b), we make the $\bar{\alpha} = 0.033$ which can equalize the *Mak.Del.Rate* and the *Ex.Cost.Rate* and make sure that both the *Mak.Del.Rate* and the *Ex.Cost.Rate* are smaller than other situations.

5.4. Comparison with the Other Solutions. Since the proposed FAUSIT algorithm is a heuristic algorithm, the most straightforward way to evaluate its performance is to compare with the optimal solution when possible. We randomly generate 10 different applications for each scale of Epigenomics shown in Table 3, and we make the $T_d = 1.0CP$.

What should be paid attention to is that although the FASTER is a copy tasks based method, but it is developed for multiple DAG based big data applications. As a result, it cannot compare the performance of it with the other two methods directly. In order to make a comprehensive comparison, on the basis of FASTER, we modified the FASTER to make it handle a single DAG based application, and this new FASTER is denoted by FASTER*.

The average of *Mak.Del.Rate* and *Ex.Cost.Rate* are displayed in Figure 6. In Figure 6(a), our FAUSIT has the minimum *Mak.Del.Rate* for any size of application. Meanwhile,

the FASTER* has higher *Mak.Del.Rate* than our FAUSIT, and the Daly's method has the highest *Mak.Del.Rate*. The result in Figure 6(a) shows that our FAUSIT has the best performance for minimizing the makespan of an application among these methods.

In Figure 6(b), due to the fact that the Daly's method only adopts the check-point mechanism, this makes it has the minimum *Ex.Cost.Rate*. Meanwhile, the FASTER* consumes the maximum cost, since the *Ex.Cost.Rate* of it is very large. Interestingly, along with the increase of the count of subtasks in an application, the *Ex.Cost.Rate* of our FAUSIT is becoming smaller, which indicates that our FAUSIT has the ability to handle much bigger scale of application.

In conclusion, compared with the FASTER, our FAUSIT outperforms FASTER on both *Mak.Del.Rate* and *Ex.Cost.Rate*. Compared with the Daly's method, our FAUSIT outperforms it much more on *Mak.Del.Rate* and only consume 9% extra cost, which still makes our FAUSIT have competitiveness in practical system. Besides, the $\bar{\alpha}$ in our FAUSIT can satisfy the requirements from different users; i.e., if the users need a shorter makespan for an application, they can turn the $\bar{\alpha}$ up. On the contrary, if the users care about the cost, they can turn the $\bar{\alpha}$ down. Thus, the $\bar{\alpha}$ make the FAUSIT have strong usability.

6. Conclusion

This paper investigates the problem of improving the fault tolerant for a big data application running on cloud. We first analyze the characteristics of running a task on a processor. Then, we present a new approach called selective mirrored task method to deal with the imbalance between the parallelism and the topology of the DAG based application running on multiple processors. Based on the selective mirrored task method, we proposed an algorithm named FAUSIT to improve the fault tolerant for a big data application running on cloud; meanwhile, the makespan and the computation cost is minimized. To evaluate the effectiveness of FAUSIT, we conduct extensive simulation experiments in the context of randomly generated workflows which are real-world application traces. Experimental results show the superiorities of FAUSIT compared with other related algorithms, such as FASTER and Daly's method. In our future work, due to the superiorities of selective mirrored task method, we will try to apply it to other big data applications processing scenarios, such as improving the fault tolerant of multiple applications in the respect of could providers. Furthermore, we will also investigate the effectiveness of the selective mirrored task method in parallel real-time applications running on cloud.

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 they have no conflicts of interest.

Acknowledgments

This work was supported partly by Doctoral Research Foundation of Liaoning under grant no. 20170520306, Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis Open Fund (HCIC201605), Guangxi Youth Capacity Improvement Project (2018KY0166), and Supercomputing Center of Dalian University of Technology.

References

- [1] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [2] Market Research Media, "Global cloud computing market forecast 2019-2024," <https://www.marketresearchmedia.com/?p=839>.
- [3] L. F. Sikos, "Big data applications," in *Mastering Structured Data on the Semantic Web*, 2015.
- [4] G.-H. Kim, S. Trimi, and J.-H. Chung, "Big-data applications in the government sector," *Communications of the ACM*, vol. 57, no. 3, pp. 78–85, 2014.
- [5] G. Wang, T. S. E. Ng, and A. Shaikh, "Programming your network at run-time for big data applications," in *Proceedings of the 1st ACM International Workshop on Hot Topics in Software Defined Networks (HotSDN '12)*, pp. 103–108, ACM, Helsinki, Finland, August 2012.
- [6] P. Costa, A. Donnelly, A. Rowstron, and G. O'Shea, "Camdoop: Exploiting in-network aggregation for big data applications," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, USENIX Association, 2012.
- [7] H. Liu, H. Ning, Y. Zhang, Q. Xiong, and L. T. Yang, "Role-dependent privacy preservation for secure V2G networks in the smart grid," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 2, pp. 208–220, 2014.
- [8] E. Sindrilariu, A. Costan, and V. Cristea, "Fault tolerance and recovery in grid workflow management systems," in *Proceedings of the 4th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS '10)*, pp. 475–480, IEEE, February 2010.
- [9] Q. Zheng, "Improving MapReduce fault tolerance in the cloud," in *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum (IPDPSW '10)*, IEEE, April 2010.
- [10] L. Peng et al., "Deep convolutional computation model for feature learning on big data in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 790–798, 2018.
- [11] Q. Zhang, L. T. Yang, Z. Yan, Z. Chen, and P. Li, "An efficient deep learning model to predict cloud workload for industry informatics," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3170–3178, 2018.
- [12] L. Man and L. T. Yang, "Hybrid genetic algorithms for scheduling partially ordered tasks in a multi-processor environment," in *Proceedings of the Sixth International Conference on Real-Time Computing Systems and Applications (RTCSA '99)*, IEEE, 1999.
- [13] Q. Zhang, L. T. Yang, and Z. Chen, "Deep computation model for unsupervised feature learning on big data," *IEEE Transactions on Services Computing*, vol. 9, no. 1, pp. 161–171, 2016.

- [14] "Cluster networking in ec2," <https://amazonaws-china.com/ec2/instance-types>.
- [15] "Google cloud," <https://cloud.google.com/>.
- [16] "Microsoft azure," <https://azure.microsoft.com/>.
- [17] J. Rao, Y. Wei, J. Gong, and C.-Z. Xu, "QoS guarantees and service differentiation for dynamic cloud applications," *IEEE Transactions on Network and Service Management*, vol. 10, no. 1, pp. 43–55, 2013.
- [18] J. Wentao, Z. Chunyuan, and F. Jian, "Device view redundancy: An adaptive low-overhead fault tolerance mechanism for many-core system," in *Proceedings of the IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC '13)*, IEEE, 2013.
- [19] M. Engelhardt and L. J. Bain, "On the mean time between failures for repairable systems," *IEEE Transactions on Reliability*, vol. 35, no. 4, pp. 419–422, 1986.
- [20] H. Akkary, R. Rajwar, and S. T. Srinivasan, "Checkpoint processing and recovery: towards scalable large instruction window processors," in *Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, pp. 423–434, IEEE Computer Society, San Diego, Calif, USA, 2003.
- [21] J. W. Young, "A first order approximation to the optimum checkpoint interval," *Communications of the ACM*, vol. 17, no. 9, pp. 530–531, 1974.
- [22] J. T. Daly, "A higher order estimate of the optimum checkpoint interval for restart dumps," *Future Generation Computer Systems*, vol. 22, no. 3, pp. 303–312, 2006.
- [23] X. Zhu, J. Wang, H. Guo, D. Zhu, L. T. Yang, and L. Liu, "Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 12, pp. 3501–3517, 2016.
- [24] X. Qin and H. Jiang, "A novel fault-tolerant scheduling algorithm for precedence constrained tasks in real-time heterogeneous systems," *Parallel Computing. Systems & Applications*, vol. 32, no. 5–6, pp. 331–356, 2006.
- [25] S. Mu, M. Su, P. Gao, Y. Wu, K. Li, and A. Y. Zomaya, "Cloud storage over multiple data centers," *Handbook on Data Centers*, pp. 691–725, 2015.
- [26] H. Topcuoglu, S. Hariri, and M. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 3, pp. 260–274, 2002.
- [27] H. Wu, X. Hua, Z. Li, and S. Ren, "Resource and instance hour minimization for deadline constrained DAG applications using computer clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 3, pp. 885–899, 2016.
- [28] G. Juve, "workflowgenerator," <https://confluence.pegasus.isi.edu/display/pegasus/workflowgenerator>, 2014.
- [29] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M. Su, and K. Vahi, "Characterization of scientific workflows," in *Proceedings of the 3rd Workshop on Workflows in Support of Large-Scale Science (WORKS '08)*, pp. 1–10, IEEE, November 2008.
- [30] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, "Characterizing and profiling scientific workflows," *Future Generation Computer Systems*, vol. 29, no. 3, pp. 682–692, 2013.

Research Article

Predicting Fine-Grained Traffic Conditions via Spatio-Temporal LSTM

Xiaojuan Wei , Jinglin Li , Quan Yuan , Kaihui Chen, Ao Zhou , and Fangchun Yang

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China

Correspondence should be addressed to Jinglin Li; jlli@bupt.edu.cn

Received 27 September 2018; Revised 6 December 2018; Accepted 24 December 2018; Published 14 January 2019

Guest Editor: Qingchen Zhang

Copyright © 2019 Xiaojuan Wei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Predicting traffic conditions for road segments is the prelude of working on intelligent transportation. Many existing methods can be used for short-term or long-term traffic prediction, but they focus more on regions than on road segments. The lack of fine-grained traffic predicting approach hinders the development of ITS. Therefore, MapLSTM, a spatio-temporal long short-term memory network preluded by map-matching, is proposed in this paper to predict fine-grained traffic conditions. MapLSTM first obtains the historical and real-time traffic conditions of road segments via map-matching. Then LSTM is used to predict the conditions of the corresponding road segments in the future. Breaking the single-index forecasting, MapLSTM can predict the vehicle speed, traffic volume, and the travel time in different directions of road segments simultaneously. Experiments confirmed MapLSTM can not only achieve prediction for road segments based a large scale of GPS trajectories effectively but also have higher predicting accuracy than GPR and ConvLSTM. Moreover, we demonstrate that MapLSTM can serve various applications in a lightweight way, such as cognizing driving preferences, learning navigation, and inferring traffic emissions.

1. Introduction

Traffic prediction of road segments is a fundamental issue in the Intelligent Transportation Systems (ITS), which can be hopefully used for planning optimal driving routes [1], urban computing [2], balancing traffic control [3, 4], and enhancing driving comfort [5]. It is necessary to explore the traffic dynamics and analyze the evolution pattern of traffic flow. Due to the generation of industrial IoT big data, network infrastructures and computational models have been equipped and applied [6–8]. If the global traffic information is not recognized accurately and timely, ITS will be not successfully deployed or the deployed system will be paralyzed sooner or later.

In general, the power of effectively predicting the future traffic conditions for road segments comes from the historical and real-time traffic information. According to the duration for the future, 3–10 days, 1–3 days, within 1 day, and no more than 15 minutes, traffic flow forecast usually is included long-term, recent-term, short-term and short-time [9]. Most of the existing methods present prediction trend either by using probability and statistics of the time-dependent evolution

of current road, or only using the pure spatial relationships among various road segments. Although available spatiotemporal information is combined to model the traffic network pattern, the information does not play out its full potential.

Traffic network possesses complicated spatio-temporal relationship. The prediction methods should have accuracy, robustness, adaptability and portability as the traffic flow is a high-dynamic, high-dimensional, non-linear and non-stationary random process. Traffic conditions of road segments are influenced inevitably by the spatio-temporal information in the traffic network. Deep learning can be used to model high-level abstractions by using multiple non-linear transformations, while the learning network has rarely taken the overall spatio-temporal dynamic pattern into account. It is not convincing to achieve accurate traffic prediction merely by spatial relations between regions or road segments. Hence, the prediction results perform not well at certain times, which occur especially when there are insufficient GPS trajectories through road segments. Based on this, it is proper to consider more supplementary aspects such as map-matching technology used to recognize traffic conditions for road segments accurately and finely.

In this paper, we propose a fine-grained and lightweight approach for traffic predicting of road segments, named MapLSTM, a spatio-temporal long short-term memory network (LSTM [10]) preluded by map-matching [11]. MapLSTM only requires vehicles GPS, and not need to deploy specialized traffic sensors in urban and not use the unobtainable data from ground loop. MapLSTM first obtains the historical and real-time traffic conditions of road segments via map-matching. Then LSTM is utilized to predict the traffic conditions of the corresponding road segments in the future. Breaking the single-index forecasting, MapLSTM can predict multiple traffic conditions for road segments simultaneously. To summarize, the major contributions of this paper consist of the following aspects:

(1) Breaking through the difficulty of obtaining segment-based traffic data, we perform the cognizing of road-grained traffic conditions via map-matching technology.

(2) Based on a large scale of taxi GPS trajectories, we propose MapLSTM to extract features from the high-dynamic, high-dimensional, non-linear and non-stationary traffic flow. And we confirm that MapLSTM have a higher predicting accuracy than GPR [1] and ConvLSTM [12].

(3) We demonstrate MapLSTM can serve to various pragmatic applications: cognizing driving preferences, learning navigation and inferring traffic emissions.

The remainder of this paper is organized as follows: Section 2 reviews the literature on traffic prediction. Section 3 describes the materials and gives details of our mechanism MapLSTM. The next, Section 4 demonstrates the effectiveness and applications. This paper ends in Section 5 with conclusion on our work.

2. Literature Review

Traffic condition prediction can not only be used as the design basis of signal control of ITS but also provide decision support for dynamic route guidance. Whereas, there are still some bottlenecks in short or long term traffic prediction through a lot of real spatio-temporal data.

Spatio-temporal semi-supervised learning model proposed in [13] can infer the volume of each road with real-world data collected from 155 loop detectors and 6918 taxis over 17 days. There are totally 19165 road segments in the urban area, but only 155 road segments are equipped with loop detectors, which results there in an inherent deviation in acquiring the city-wide traffic volumes. Although the constructed affinity graph can characterize the similarities among roads based similar speed patterns, the factors influencing traffic flow are not only the speed of vehicles, but also the road topology, road structure, the regional characteristics, and so on. Traffic conditions of each road segment cannot be predicted accurately only by the spatial relations on the macro.

A vehicle speed is influenced by many factors: the vehicle type, the traffic conditions and the driver's behaviour. A data driven model is proposed in [14] for vehicle speed prediction where the average traffic speed is estimated based on historical traffic data at first and then the statistical relationship with individual vehicle speed is presented by hidden markov

models. Finally, the individual vehicle speed is predicted by forward-backward algorithm. Another mechanism proposed in [15] is a cooperative method which combines with fuzzy markov model and auto-regressive model. These machine learning approaches for vehicle speed prediction do not care about the basic data sources, and focus more on the accuracy of prediction algorithms rather than the accuracy of segment-based traffic prediction.

DeepSense [16] is a typical deep learning approach for traffic prediction with Taxi GPS traces. DeepSense gains the prediction results based on sufficient dataset by using Restricted Boltzmann Machine. Due to the night data is too sparse, so DeepSense made a prediction based filling according to the data of the same time in history. But this prediction-based prediction approach may lose credibility in deep learning. In addition, DeepSense extract and classify the speed only on 0 ~ 60km/h to reflect the traffic congestion or smooth, which lacks universality in some other regions.

Understanding traffic density from large-scale images is another way to recognize the traffic status. Reference [17] as a related work selects a region of interest in a video stream at first, then counts the number of vehicles in the region for each frame, so the density is calculated by dividing that number by the region length. Reference [18] is another image-based learning to measure traffic density using a deep convolutional neural network. These vision-based cognitive methods mainly play a role in local regions, which can dedicate to the operational control but cannot make an efficient decision in tactical planning with in the long run.

In addition, the predicted object is univocal in the existing methods, more is traffic volume or speed, which can merely infer the traffic state of the road segment is congestion, slow, normal, moderate, and unimpeded. It is necessary to explore fine-grained and accurate perception in a simple way.

3. Materials and MapLSTM

In this section, we first provide materials on GPS trajectory, map-matching, and LSTM. Then we depict MapLSTM designed for traffic prediction.

3.1. Materials

3.1.1. GPS Trajectory. Taxis can be considered as ubiquitous mobile sensors constantly probing a city's rhythm and pulse. Being inherent characteristic, GPS-based taxis have proven to be an extremely useful data source for uncovering the underlying traffic behaviour. So far, the taxi GPS data have been used for urban computing, detecting hot spots, map reconstruction, finding routes, and so on [2, 19].

The GPS records of a large number of taxis in a city are routinely saved to a log file L , resulting in a very large data set $L = \{p_{11}, p_{12}, \dots, p_{1n}; \dots; p_{i1}, p_{i2}, \dots, p_{in}; \dots\}$. Fields for each GPS record generally contains TaxiID, Location (Longitude, Latitude), Speed, Event, GPS_state, Bearing, and 6 commonly used timing units: YYYY-MM-DD HH:MM:SS. Figure 1 shows an example of GPS log and trajectories. A trajectory T is a time series of GPS points with the time interval between any consecutive GPS points not exceeding

id	event	speed	gps_state	time	loc
490911	4	12	1	2012-11-10T00:14:302	[116.4763412, 39.9084358]
174669	4	56	1	2012-11-10T00:14:342	[116.3215103, 39.9749603]
431093	4	45	1	2012-11-10T00:14:342	[116.3878275, 39.8554573]
214981	4	52	1	2012-11-10T00:14:352	[116.4019732, 39.9967346]
194059	4	8	1	2012-11-10T00:14:362	[116.3780594, 39.888031]
102524	4	43	1	2012-11-10T00:14:362	[116.3636475, 40.0007591]
154958	4	52	1	2012-11-10T00:14:392	[116.2795029, 39.8959732]
154778	4	21	1	2012-11-10T00:14:402	[116.5870667, 40.0791435]
489612	4	19	1	2012-11-10T00:14:412	[116.2969513, 39.9593964]
235465	4	0	1	2012-11-10T00:14:412	[116.5154037, 39.9145546]
194713	4	52	1	2012-11-10T00:14:422	[116.3490982, 39.8671875]
194150	4	56	1	2012-11-10T00:14:442	[116.3636017, 39.9786758]
153633	4	12	1	2012-11-10T00:14:462	[116.428978, 39.9323196]
69947	4	58	1	2012-11-10T00:14:462	[116.2947998, 39.9778099]
174669	4	54	1	2012-11-10T00:14:462	[116.3230084, 39.9750824]
.....					

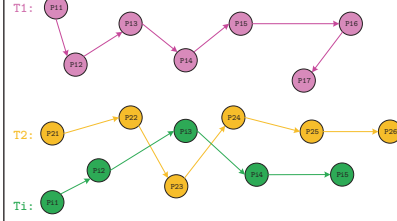


FIGURE 1: An example of GPS log and GPS trajectories.

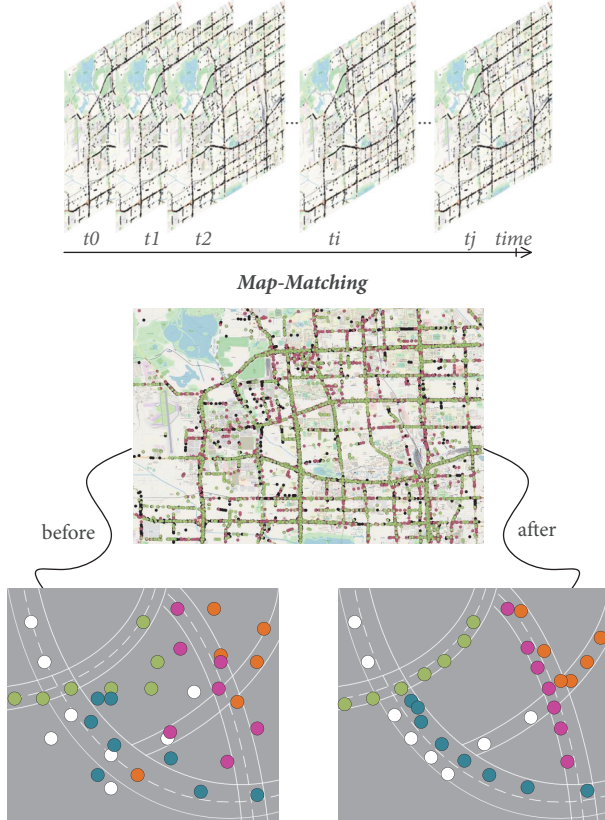


FIGURE 2: Map-matching when before and after.

a certain threshold Δt (usually $\Delta t \geq 1$ min), i.e., $T_i: p_{i1} \rightarrow p_{i2} \rightarrow p_{i3} \rightarrow \dots \rightarrow p_{in}$.

3.1.2. Map-Matching. Map-matching is the process of aligning a sequence of observed GPS positions with the road network on a digital map [20]. As a preprocessing step of MapLSTM, map-matching can effectively improve the existing huge amount of low-sampling-rate GPS trajectories in data set.

As shown in Figure 2, map-matching can be performed with the same or different time interval as the GPS points. The GPS points without map-matching can only be mapped to the road network. Not all GPS points can be mapped to their corresponding segments due to the GPS positioning error. But after map-matching, all GPS points can be corrected to the corresponding road segments.

Input: The network's input in current time x_t ;
the initial weight matrix W ,
and bias units b about gates I , O , F ;
Output: The forget gate, input gate, cell state in different time, output gate and the cell output.

- (1) $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$;
- (2) $\#$: σ represent Sigmoid function.
- (3) $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$;
- (4) $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$;
- (5) $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$;
- (6) $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$;
- (7) $h_t = o_t * \tanh(C_t)$;
- (8) **return** $f_t, i_t, \tilde{C}_t, C_t, o_t, h_t$.

ALGORITHM 1: For calculating each element of LSTM.

3.1.3. LSTM. LSTM [10] is a time recurrent neural network, which is the most widely used method to process and predict events with relatively long intervals in time series. LSTM can learn about long-term reliant information by input gate I , output gate O , and forget gate F , where, I determines how much of the network input at the current time x_t is saved to the cell state c_t . O determines how much of the control unit state c_t is output to the current output value h_t of LSTM. F determines how much of the cell state from the previous time c_{t-1} remains to the current time c_t . In short, the input X at different time determines the cell state C at the corresponding time and the current cell state c_t will be affected by the previous cell c_{t-1} .

The calculation of each element of LSTM is shown in Algorithm 1. At the current time t , f_t denotes forget gate, i_t represents input gate obtained by the previous output h_{t-1} and the current input x_t , C_t denotes the cell state and \tilde{C}_t denotes the cell state at the previous time, o_t denotes the output gate, and h_t denotes the cell output. LSTM can not only save information long ago under the control of F but also avoid the current irrelevant content into memory based the gate I .

3.2. MapLSTM. MapLSTM is fine-grained and lightweight way. It only requires sampled GPS points of vehicles and not need to deploy expensive traffic sensors in urban and not use the unobtainable data from ground loop. In this section, we describe MapLSTM in detail.

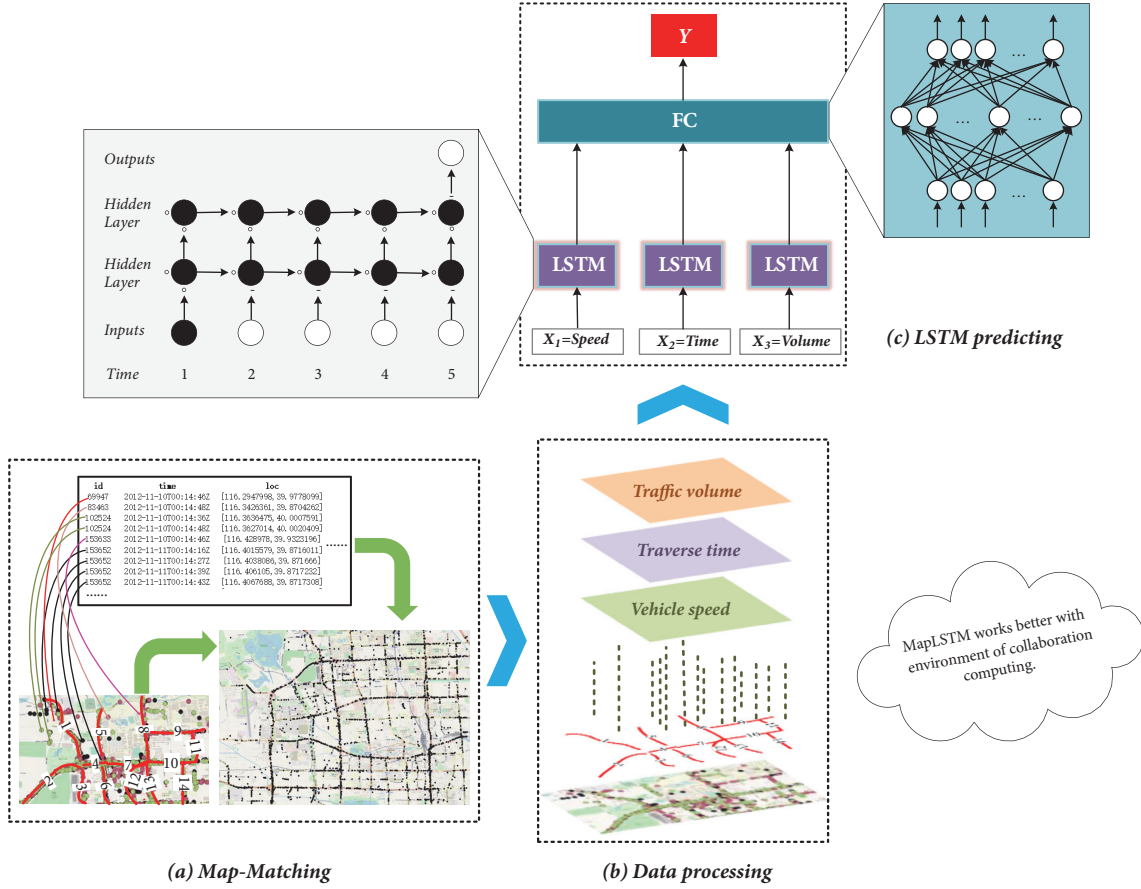


FIGURE 3: MapLSTM framework for traffic prediction. It consists of three processes: Map-matching, data processing and LSTM predicting.

3.2.1. Framework. Figure 3 shows the framework of MapLSTM, which consists of three processes: map-matching, data processing and LSTM predicting.

(a) Map-Matching. A large number of sampled GPS points stored in GPS log need to be matched to road segments. In order to facilitate the operation, it is necessary to manually redivide road segments based on the road network before matching. Generally, the division is based on the intersection, or no redivision, just based on the inherent segments structure in road network, if the calculation resources and road segments information are sufficient and detailed. We do our best to maintain the original topography relationship between the divided road segments. After map-matching, all GPS points can be shifted to the corresponding road segments.

(b) Data Processing. The road segments experienced map-matching also mean the information has been extended, where the road segments and vehicle information are paired off according to their ID and location. Therefore, we can have information statistics including vehicle speed, traverse time, and traffic volume taking one road segment as a unit (i.e., segment-based). The traverse time can be counted in different directions: from west to east, from east to west, from north to south, and from south to north. The processed data are sent to prediction model LSTM as the training set and testing set.

(c) LSTM Predicting. The traffic data of vehicle speed, the traverse time, and traffic volume based road segments are input to LSTM concurrently for predicting task. The hidden layers of LSTM can control the long-term or short-term impact on the current state. After output layer of LSTM, it goes through a full connected network with three layers, in which the purpose is to better explore the implied relationships between states.

MapLSTM enables cognition of road segment-based traffic conditions in a lightweight way. For the real-time cognition of global situations, MapLSTM is still valid by collaboration computing where a groups of cells work together to accomplish a relatively large task. Edge computing after cloud computing is a typical collaborative computing environment and has been widely used [21, 22].

3.2.2. Map-Matching Algorithm. Before map-matching, it is necessary to have a information understanding about roads and vehicles. Table 1 describes an example with a sample of the information. All the information about roads and vehicles can be correlated based the auxiliary information (ID, longitude and latitude).

ST-Matching [20] is a pathway with candidate computation and spatio-temporal analysis for low-sampling-rate GPS trajectories. We follow ST-Matching analysis architecture and make a map-matching work on a real digital map in Beijing. As described in Algorithm 2, for the available

TABLE 1: An example with a sample of the main information about road and vehicle.

Name	The Main Fields								
Road	ID	MapID	PathName	Pathclass	Oneway	Width	Length	Direction	Meters
	59565200918	595652	Xing Fu Xi Jie	4	F	30	0.284	2	3.68
Vehicle	ID	Bearing	Speed	State	Longitude	Latitude	Event	Time	Positioning
	6409	84	46	1	3973633	11633100	1	20160916182046	GPS/BeiDou/Mix

Input: Beijing Road network R , Coordinate axis A , Trajectories T , where, $T = \{t_1, t_2, t_3, \dots, t_n\}$, $t_i = \{p_{i1} \rightarrow p_{i2} \rightarrow p_{i3} \rightarrow \dots \rightarrow p_{in}\}$, p_{ij} is a GPS sampling point, $i, j \in [1, n]$;

Output: The one-to-one results of road segments and vehicles information: M_T' .

- (1) Initialize $CPset = 0$;
- (2) Repeat $i = 1, 2, 3, \dots, n$;
- (3) For $j = 1$ to n
- (4) $S_{i,j} = GetCandidatePoints(p_j, R)$
- (5) $CPset.add(S_{i,j})$
- (6) End for;
- (7) $k = 1$;
- (8) While $k \leq \|CPset\|$ do
- (9) $V_s = GetSpaVal(R, A, CPset(k), dist(p, CPset^p))$
- (10) $V_t = GetTemVal(R, A, Speed, time(p_{i,j}, p_{i,j+1}))$
- (11) $M_T = MatchSeq(V_s, V_t)$
- (12) $k++ = 1$
- (13) End while;
- (14) Visualized $M_T \rightarrow M_T'$;
- (15) **return** M_T' .

ALGORITHM 2: Map-matching algorithm.

historical trajectories, GPS sampling points in the trajectories are traversed to get the candidate point set which waiting to be corrected. For all candidate points, the spatial value can be reached by combining with the information of road network, longitude, latitude, and distance, and the temporal value can be reached by adding the time information. After spatial analysis and temporal analysis, matching results can be accomplished.

After map-matching, roads information where the vehicles are located can be easily obtained, and the traffic data about the roads can also be clearly gained after statistics in turn.

3.2.3. Training Data Generating. The raw trajectory data cannot be used directly for our predicting task. It is necessary to match and statistics at first. If we want to get the traffic status prediction of road segments, we need to make a segment-based statistics about the traverse time in different directions, the vehicle speed and the traffic volume.

The data of traverse time in different directions, the average vehicle speed and traffic volume of road segments can be generated by Algorithm 3. When map-matching is done, more fine-grained data can also be obtained such as the average speed and traffic volume under different directions of road segments. The data after map-matching and statistics

Input: Road Log R^l , GPS Log V^l ,
Output: The traffic data about the road segments.

- (1) Initialize $TT_we, TT_ew, TT_sn, TT_ns, Speed, Count_we, Count_ew, Count_ns, Count_sn$;
- (2) $GetVehicleColumns(ID, Time, V_Speed, V_lon, V_lat)$;
- (3) $GetRoadColumns(RoadID, R_lon, R_lat)$;
- (4) Repeat $RoadID$;
- (5) $Flag = 0$;
- (6) For $i = StartTime$ to $EndTime$
- (7) if
- (8) $t_i, V_lon < R_w_lon; V_lat < R_n_lat$
- (9) $t_{i+1}, V_lon \geq R_w_lon; V_lat \geq R_n_lat$
- (10) $t_{n+i}, V_lon < R_e_lon; V_lat < R_s_lat$
- (11) $t_{n+i+1}, V_lon \geq R_e_lon; V_lat \geq R_s_lat$
- (12) then
- (13) $Flag = 1$
- (14) $Count_we++ = 1$;
- (15) $Count_ns++ = 1$
- (16) $TT_we++ = t_{n+i+1} - t_{i+1}$;
- (17) $TT_ns++ = t_{n+i+1} - t_{i+1}$
- (18) Imitate $GetCountData(Count_ew, Count_sn)$;
- (19) Imitate $GetTTData(TT_ew, TT_sn)$;
- (20) if $Flag = 1$ then $Speed++ = V_Speed$;
- (21) End for;
- (22) $Avg_TT_we = TT_we / Count_we$;
- (23) Imitate $GetData(Avg_TT_ew, Avg_TT_sn, Avg_TT_ns)$;
- (24) $Volume = \sum_{p,q}^{w,e,n,s} Count_pq$;
- (25) $AvgSpeed = Speed / Volume$;
- (26) **return** $Avg_TT_we, Avg_TT_ew, Avg_TT_sn, Avg_TT_ns, AvgSpeed, Volume$.

ALGORITHM 3: To generated the segment-based traffic data.

can be used, which also mean that the training data and the testing data of prediction network are generated.

4. Experiment

We compare the following experiments to verify the performance of MapLSTM.

(1) *Gaussian Process Regression (GPR)* [1]. It is one of the most popular used prediction algorithms and often used to compare performance as a baseline.

(2) *ConvLSTM* [12]. It extends LSTM to have convolutional structures in both the input-to-state and state-to-state transitions, and captures spatiotemporal correlations better.

(3) *ConvLSTM+*. It is ConvLSTM increased epoch numbers.

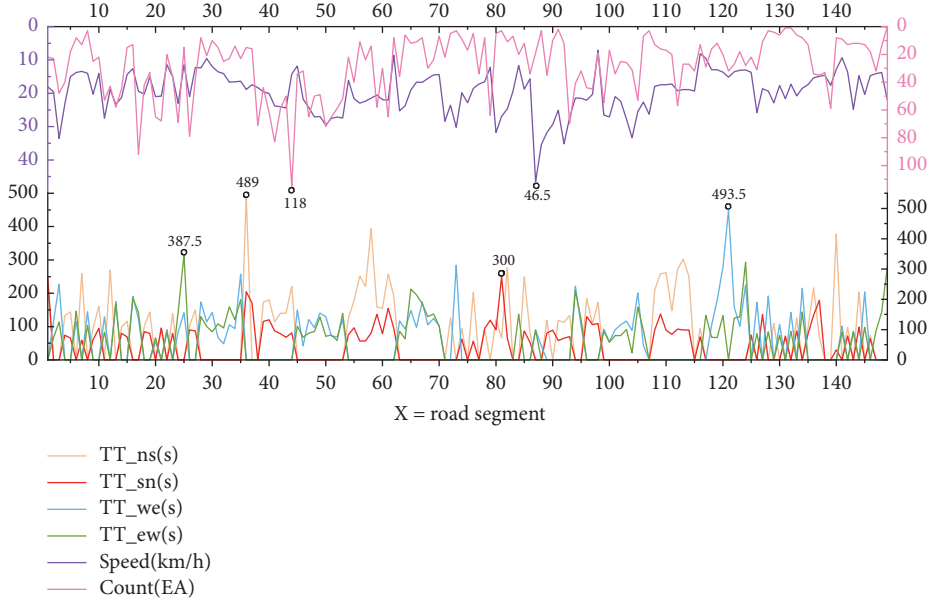


FIGURE 4: Traffic data about road segments at 8:00 on November 1, 2012.

4.1. Datasets. A large scale of real taxi trajectory data are used in our predicting task. The data package of GPS log includes over 400,000 taxicabs' trajectories in November 2012, Beijing. And full-scale entries are contained during 24 hours for each day. We use data between 8:00 ~ 20:00 in weekdays as the traffic pattern can be learned better in the daytime. We can get dataset $22 \times 13 \times 30$ when the time interval is 2 minutes.

There are too many segments in road network R , so we manually redivide the road segments based R to verify the feasibility of MapLSTM. The road segments after redivision is stored to set R_{ss} . Figure 4 depicts the traffic data of R_{ss} on November 1, 2012 at 8 o'clock, including the traverse time in different directions $TT_{W \leftarrow E}$, $TT_{N \leftarrow S}$, the average vehicle speed, and the traffic volume.

4.2. Training. In MapLSTM, the obtained dataset is divided into training set and test set in an 8 : 2 ratio. The prediction model has $batch_size = 20$, $lr_decay = 0.93$, $hidden_size = 250$, and $num_steps = 6$ (the size of window, that means using data from the previous 6 time units to predict the next one). The sizes of the three full connection layers are 180×150 , 250×180 , and 250×250 , which is related to the total number of road segments.

ConvLSTM has the same dataset as MapLSTM, and the model has $input = 21 \times 21$, $batch_size = 8$, $num_steps = 6$, $kernel = 5 \times 5$, $filters = 10$, and $max_epoch = 70$. ConvLSTM+ is iterated 20 times more than ConvLSTM.

4.3. Performance Evaluation. Mean absolute error (MAE) is the most commonly used criteria in predictive algorithms and is employed to evaluate the proposed MapLSTM.

$$MAE = \frac{1}{N} \sum_{i=1}^N |(f_i - y_i)| \quad (1)$$

where f_i is the predicted value and y_i is the observed value. The smaller the MAE, the stronger the predictable ability of algorithms.

As shown in Table 2, whether it is the MAE of vehicle speed, traffic count, or travel time in different direction TT_{WE} (from west to east), TT_{EW} (from east to west), TT_{SN} (from south to north), and TT_{NS} (from north to south), MapLSTM is smaller than GPR and ConvLSTM. For a certain algorithm, the closer the value of "Train" and "Test" of each parameter is, the more robust it is. The results of ConvLSTM are similar to MapLSTM but do not exceed MapLSTM. That is because ConvLSTM with the ability to capture spatiotemporal correlations is good at predicting relatively single spatial pattern, but the spatial patterns of road traffic are complex. In the future, we will focus on complex spatial correlations in traffic environment. Compared to ConvLSTM, some parameters of ConvLSTM+ are slightly better because ConvLSTM+ increased the number of epoch.

It is important to note that MAE is affected by the accuracy of the raw data and it will decline if the dataset is large enough.

4.4. Applications

4.4.1. Cognizing Driving Preference. Different drivers have different preferences about different types of roads, and they also have different impulse to reroute roads due to their different tolerance about the cost expectations of current congestion. For example, the drivers with low tolerance may choose a highway bypass which have a lower congestion cost expectations but have more traffic lights. Tolerance of drivers changes dynamically with various spatial-temporal conditions such as travel distance, congestion time, and arrival time. Therefore, a large deviation between the traffic optimization results and the actual expectation of drivers will lead to failure of traffic scheduling. Quite a few drivers

TABLE 2: MAEs comparison of GPR, ConvLSTM, and MapLSTM.

Algorithms		Speed	Count	TT_WE	TT_EW	TT_SN	TT_NS
GPR		70.79	46.43	66.81	70.7	63.2	66.08
ConvLSTM	Train	18.78	7.18	18.75	18.27	17.77	19.01
	Test	19.27	7.32	18.71	18.11	18.14	19.29
ConvLSTM+	Train	19.44	7.13	18.75	18.59	17.82	19.46
	Test	18.91	6.89	18.71	17.96	17.85	18.93
MapLSTM	Train	18.33	5.59	16.42	16.5	16.94	18.62
	Test	18.53	7.05	16.91	17.21	17	18.57

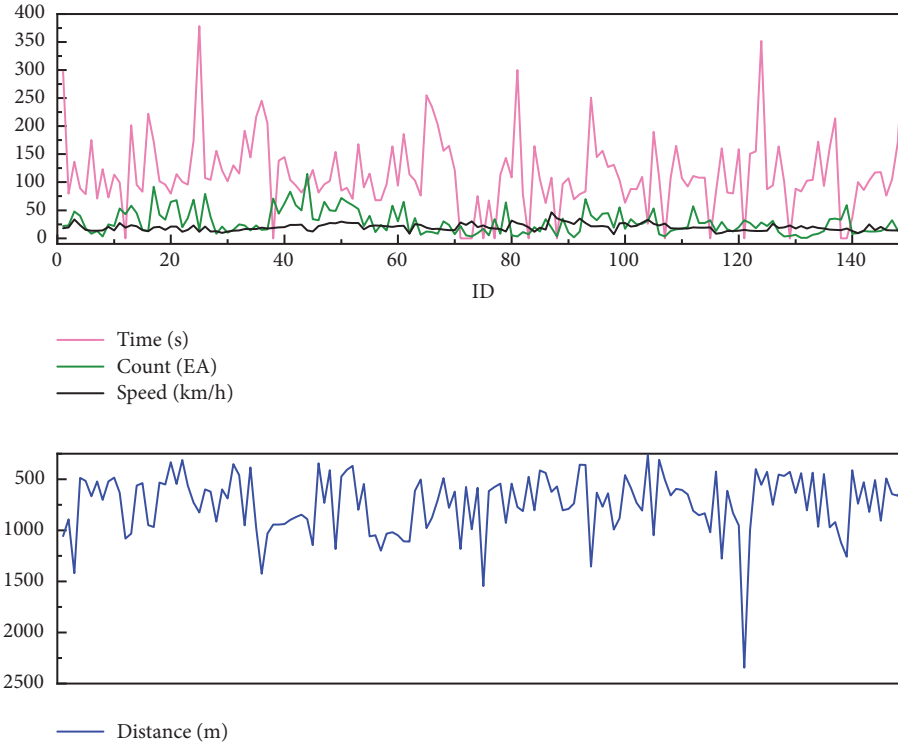


FIGURE 5: Segment-based traffic information at a certain time.

choose a looked like shortest road, only to find the route is congested by many vehicles whose drivers make a similar decision.

The traditional route planning methods are more inclined to train drivers' basic selection tendency and do not have personalized features. The participants in these methods are considered the rational contenders perfectly. The planned result is the purely rational optimal solution and does not express the noncomplete rational decision-making preference for drivers in the actual routing decisions. Although the questionnaire may be a handy pathway for cognizing driving preferences, it lacks efficiency and comprehensiveness.

The premise of learning driving preferences is to obtain an understanding about the roads conditions. The more we aware of road properties, the more satisfied we cognise the personalized preferences. MapLSTM can have a fine-grained cognition of road traffic conditions, so we can learn the driving preferences easily. For drivers of vehicles, there are two preferences getting the most attention: time and distance.

Figure 5 shows the traffic information about the travel time, distance, vehicle density, and speed of each road segments in *Rss* where the vehicle is driving from place A to place B. In order to compare the preference in driving, the full driving routes based different driving preferences including average speed, vehicle count, distance and travel time are shown in Figure 6.

4.4.2. Learning Navigation. Navigating vehicles to their destination is an important service for ITS. In addition to using historical and real-time traffic conditions, the state-of-the-art systems take into account the impact on the future traffic conditions which can be obtained by predicting. For example, the method in [23] has the ability of learning experience-based autonomous navigation based the global traffic dynamic, and the method in [1] is another dynamic planning scheme based on situation awareness where the city sensors are deployed to maintain an up-to-date view of the city's current traffic state.

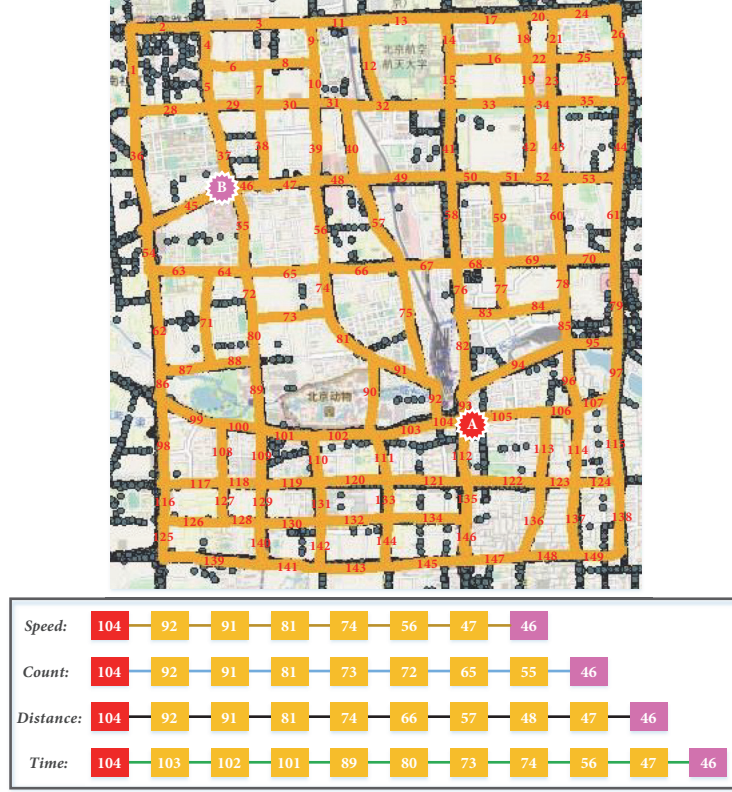


FIGURE 6: Routes with different driving preferences from A to B.

As mentioned above, the existing methods are still laborious for lightweight, fine-grained, and accurate prediction. So we propose MapLSTM to predict traffic conditions effectively. We analyze and compare the use about the predicted traffic conditions in navigation planning, as in Table 3, the lower the computing complexity, the lighter the planning algorithm; the higher the navigation accuracy, the better the navigation performance; perdurability represents the sustainability of a transportation system; the higher the perdurability, the more sustainable the transportation system.

4.4.3. Inferring Traffic Emissions. In the COPERT model [28], hot emissions are one of the key essentials about traffic emissions. Hot emissions occur when the engine of vehicle is at its normal mode. Hot emission factor EF , the amount of pollutant a single vehicle emits per kilometer (g/km), is calculated as a function of travel speed $v(km/h)$ [29].

$$EF = \frac{(a + cv + ev^2)}{(1 + bv + dv^2)} \quad (2)$$

where, a, b, c, d, e are the pollution emission parameters of COPERT model, these values are given in [29] to caluminate different kinds of emissions and gas consumption: CO, Hydrocarbon, Nox, Fuel Consumption (FC).

As in Figure 7, we infer different kinds of traffic emissions and gas consumption of 126th road segment at 10:00 in the next five days, the average of CO is about 0.5, Hydrocarbon is 0.04, Nox is 0.09, FC is 4.24, CO_2 is 4.29, and $PM_{2.5}$ is 0.007.

As for other pollutants like CO_2 and $PM_{2.5}$, their emission factors are proportional to FC.

$$EF_{co_2} = 3.18 * EF_{FC} \quad (3)$$

$$EF_{PM_{2.5}} = 3 * 10^{-5} * EF_{FC}$$

4.4.4. Other Applications. Table 4 compares the applications about traffic prediction in recent two years. It can be seen from Table 4 that the traffic prediction methods is more inclined to use machine learning and deep learning algorithm to achieve more accurate and larger regional prediction; the advance cannot be separated from the rapid development of machine learning and deep learning in recent years.

5. Conclusions

Urban road traffic system is the lifeblood of a city, which ensures its operation. Predicting traffic conditions for road segments is the prelude of working on intelligent transportation. In this paper, we proposed MapLSTM, a traffic predicting mechanism for road segments, to promote the development of ITS. MapLSTM can accelerate the landing of many applications in a lightweight and fine-grained way. In the future, autonomous humanlike driving based on road topography is worth concern, and we will focus on complex spatial correlations in traffic environment.

TABLE 3: Applications and comparisons about the predicted traffic conditions in navigation planning.

Literature	Raw data source	Object-based	Pathway	Core algorithm	Complexity	Accuracy	Save time	Perdurability
[1]	smart sensors	city	self-aware	Gaussian Process Regression	middle	middle	middle	low
[23]	GPS points	region	autonomous	Value Iteration Network	middle	middle	high	middle
[24]	street-view images	intersection	autonomous	CNN+RL+A*	high	middle	high	low
[25]	GPS points	region	agents	Ant Colony+RL	middle	middle	middle	middle
[26]	vehicles sharing	city	RIS	statistics	low	low	middle	low

TABLE 4: Applications and comparisons about the traffic prediction.

Year	Literature	Basic data source	Target	Term	Core algorithm	Complexity	Granularity	Object-based
2018	[18]	web camera	traffic density	short	Convolutional neural network	high	fine-grained	intersection
	[27]	an open dataset	traffic flow	long/short	Generative adversarial network	high	coarse-grained	freeway
2017	[17]	web camera	traffic density	short	Fully convolutional networks	high	fine-grained	restricted area
	[15]	an experimental car	vehicle speed	short	Auto-regressive model	middle	fine-grained	road segment
	[14]	floating car	vehicle speed	short	HMMs+SUMO	middle	coarse-grained	motorway
	[13]	Loop Detector	traffic volume	short	ST semi-supervised learning	low	fine-grained	road segment
	[1]	traffic loops	traffic flow	long	Gaussian process regression	low	coarse-grained	region

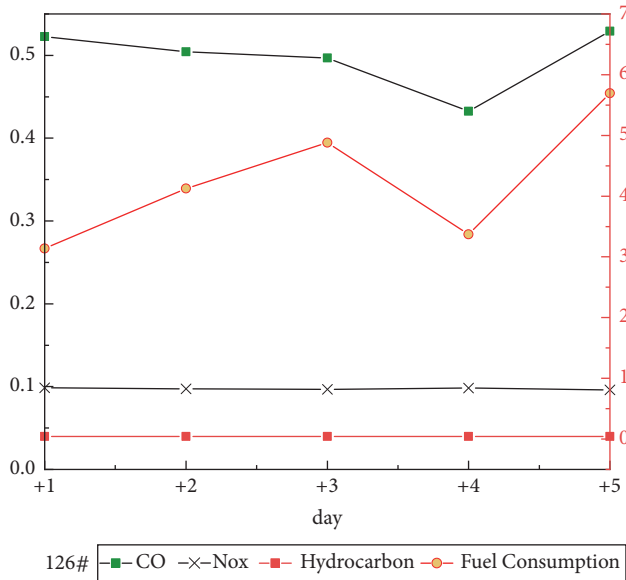


FIGURE 7: Interrupting traffic emissions of 126th road segment.

Data Availability

We used the source code of ConvLSTM in our paper; the URL is: “<https://github.com/carlthome/tensorflow-convlstm-cell>.” Moreover, we used the dataset “T-Drive Taxi Trajectories” released by MSRA; the URL is “<https://www.microsoft.com/en-us/research/project/urban-computing>.” There is just one week of data in released dataset. Although one week of data can also conduct secondary analyses, we used one month of data of “T-Drive Taxi Trajectories” in our experiments for better performance, in which data was from the previous cooperation project.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Natural Science Foundation of Beijing under Grant no. 4181002, and the Natural Science Foundation of China under Grant no. 61876023.

References

- [1] T. Liebig, N. Piatkowski, C. Bockermann, and K. Morik, “Dynamic route planning with real-time traffic predictions,” *Information Systems*, vol. 64, pp. 258–265, 2017.
- [2] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, “Urban computing with taxicabs,” in *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp '11)*, pp. 89–98, Toulouse, France, September 2011.
- [3] G.-R. Iordanidou, I. Papamichail, C. Roncoli, and M. Papa-georgiou, “Feedback-based integrated motorway traffic flow control with delay balancing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2319–2329, 2017.
- [4] L. Li, K. Ota, and M. Dong, “Humanlike driving: empirical decision-making system for autonomous vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 6814–6823, 2018.
- [5] J. Li, G. Luo, N. Cheng et al., “An end-to-end load balancer based on deep learning for vehicular network traffic control,” *IEEE Internet of Things Journal*, 2018.
- [6] S. Choudhury, “Cellular automata and wireless sensor networks,” in *Emergent Computation*, pp. 321–335, Springer, 2017.
- [7] Q. Zhang, L. T. Yang, Z. Chen, P. Li, and F. Bu, “An adaptive dropout deep computation model for industrial iot big data learning with crowdsourcing to cloud computing,” *IEEE Transactions on Industrial Informatics*, 2018.
- [8] Q. Zhang, L. T. Yang, A. Castiglione, Z. Chen, and P. Li, “Secure weighted possibilistic c-means algorithm on cloud for clustering big data,” *Information Sciences*, 2018.
- [9] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, “Road traffic forecasting: recent advances and new challenges,” *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 93–109, 2018.
- [10] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [11] S. S. Chawathe, “Segment-based map matching,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 1190–1197, Istanbul, Turkey, June 2007.
- [12] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS '15)*, pp. 802–810, December 2015.
- [13] C. Meng, X. Yi, L. Su, J. Gao, and Y. Zheng, “City-wide traffic volume inference with loop detector data and taxi trajectories,” in *Proceedings of the 25th ACM SIGSPATIAL International Conference*, pp. 1–10, Redondo Beach, Calif, USA, November 2017.
- [14] B. Jiang and Y. Fei, “Vehicle speed prediction by two-level data driven models in vehicular networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1793–1801, 2017.
- [15] J. Jing, D. Filev, A. Kurt, E. Ozatay, J. Micheline, and U. Ozguner, “Vehicle speed prediction using a cooperative method of fuzzy Markov model and auto-regressive model,” in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '17)*, pp. 881–886, Los Angeles, Calif, USA, June 2017.
- [16] X. Niu, Y. Zhu, and X. Zhang, “DeepSense: A novel learning mechanism for traffic prediction with taxi GPS traces,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '14)*, pp. 2745–2750, Austin, TX, USA, December 2014.
- [17] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, “Understanding traffic density from large-scale web camera data,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*, pp. 4264–4273, Honolulu, HI, USA, July 2017.
- [18] J. Chung and K. Sohn, “Image-based learning to measure traffic density using a deep convolutional neural network,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1670–1675, 2018.
- [19] P. S. Castro, D. Zhang, C. Chen, S. Li, and G. Pan, “From taxi GPS traces to social and community dynamics,” *ACM Computing Surveys*, vol. 46, no. 2, pp. 1–34, 2013.

- [20] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate GPS trajectories," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 352–361, ACM, Seattle, WA, USA, November 2009.
- [21] 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.
- [22] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, "A double deep q-learning model for energy-efficient edge scheduling," *IEEE Transactions on Services Computing*, 2018.
- [23] S. Yang, J. Li, J. Wang, Z. Liu, and F. Yang, "Learning Urban Navigation via Value Iteration Network" in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '18)*, pp. 800–805, Changshu, Suzhou, China, June 2018.
- [24] S. Brahmabhatt and J. Hays, "DeepNav: Learning to Navigate Large Cities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*, pp. 3087–3096, Honolulu, HI, USA, July 2017.
- [25] A. Eydi, S. Panahi, and I. iNakhai Kamalabadi, "User-based vehicle route guidance in urban networks based on intelligent multi agents systems and the ant-q algorithm," *International Journal of Transportation Engineering*, vol. 4, no. 3, pp. 147–161, 2017.
- [26] T. Yamashita, K. Izumi, and K. Kurumatani, "Car navigation with route information sharing for improvement of traffic efficiency," in *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (ITSC '04)*, pp. 465–470, Yokohama, Japan, October 2004.
- [27] A. Koesdwiady and F. Karray, "New results on multi-step traffic flow prediction," *Artificial Intelligence*, 2018, <https://arxiv.org/abs/1803.01365>.
- [28] L. Ntziachristos, Z. Samaras, S. Eggleston et al., "Copert iii, computer programme to calculate emissions from road transport, methodology and emission factors (version 2.1)," *European Energy Agency*, 2000.
- [29] J. Shang, Y. Zheng, W. Tong, E. Chang, and Y. Yu, "Inferring gas consumption and pollution emission of vehicles throughout a city," in *Proceedings of the 20th ACM SIGKDD International Conference*, pp. 1027–1036, New York, NY, USA, August 2014.

Research Article

Attribute Reduction Based on Genetic Algorithm for the Coevolution of Meteorological Data in the Industrial Internet of Things

Yong Cheng,¹ Zhongren Zheng,² Jun Wang,^{1,2} Ling Yang,³ and Shaohua Wan⁴ 

¹Division of Science and Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China

²Department of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

³Department of Electronics, Binjiang College, Nanjing University of Information Science and Technology, Nanjing 210044, China

⁴School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China

Correspondence should be addressed to Shaohua Wan; shaohua.wan@ieee.org

Received 23 August 2018; Revised 17 November 2018; Accepted 11 December 2018; Published 3 January 2019

Guest Editor: Mianxiong Dong

Copyright © 2019 Yong Cheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the problem of attribute redundancy in meteorological data from the Industrial Internet of Things (IIoT) and the slow efficiency of existing attribute reduction algorithms, attribute reduction based on a genetic algorithm for the coevolution of meteorological data was proposed. The evolutionary population was divided into two subpopulations: one subpopulation used elite individuals to assist crossover operations to increase the convergence speed of the algorithm, and the other subpopulation balanced the population diversity in the evolutionary process by introducing a random population; these two subpopulations completed the evolutionary operations together. With the TSDPSO-AR algorithm and ARAGA algorithm, the attribute reduction operation for precipitation in meteorological data was performed. The results showed that the proposed algorithm maintained the diversity of the population during evolution, improved the reduction performance, and simplified the information system.

1. Introduction

With the development of the Internet of Things technology, a large number of sensors and smart terminals are used in traditional industries, which will lead to a tremendous growth in big data. How to effectively manage large amounts of data in the Industrial Internet of Things (IIoT) to improve industrial production efficiency has become an urgent problem that needs to be solved. Meteorological elements are increasing remarkably, which brings some challenges [1–4]. To address this issue, this paper designs configurable meteorological data acquisition to meet the needs of more application scenarios. The increase in data amount is beneficial to the improved mining of potential meteorological patterns, but there is no clear purpose in the process of collecting meteorological data, and the change in meteorological phenomena is only related to some meteorological elements collected, where the attribute redundancy in the collected meteorological data is large. These redundant attributes not only reduce the mining

efficiency of meteorological data but also reduce the data mining accuracy. Therefore, it is very important to perform attribute reduction on collected weather data. In the rough set theory, which is a data mining method that effectively deals with fuzzy and uncertain information, one of its core contents is deleting redundant attributes in the knowledge base under the condition of keeping the decision ability of the knowledge base unchanged [5]. Therefore, using attribute reduction to delete redundant attributes in meteorological data and improving the mining efficiency of meteorological data has important practical significance [6]. Many scholars nationally and abroad have conducted in-depth research and discussions on this method and have also made remarkable achievements. It has been proven that the minimum attribute reduction used for solving information systems is an NP-hard problem. Therefore, many scholars use heuristic algorithms to improve the reduction efficiency. You Z et al. discuss the attribute kernel and attribute reduction operation of multiple decision tables in the distributed environment and

proposes an information entropy reduction algorithm based on the vertical distribution in the multidimensional table [7]. This method reduces the communication cost in the process of distributed reduction and improves the reduction efficiency through parallel and conditional information entropy and elements in the transmission class. Heuristic-based attribute reduction improves the reduction efficiency to some extent, but there are still some shortcomings. To further improve the reduction efficiency, many scholars combine rough set attribute reduction with other optimization algorithms. The coevolution reduction algorithm, combined with the quantum frog group, was proposed by Ding Weiping et al. in [8]. Using the optimal execution experience of the frog group and elite individuals to guide the model group to the target direction quickly, the convergence efficiency and global search ability of the attribute reduction are improved, but the cooperative coevolution reduction algorithm is suitable for high performance. Dimensional data sets greatly reduce the performance of data reduction with smaller data dimensions. Chen J et al. proposed an efficient rough set clustering algorithm based on a genetic algorithm [9]. The global search ability of the genetic algorithm was used to improve the convergence speed of the algorithm. Zhang Rongguang et al. proposed a particle set based on the rough set attribute reduction algorithm [10]. By introducing the improved tabu search algorithm, the local search strategy of the particle swarm optimization was improved, and the diversity of the population was improved. Based on this background, attribute reduction based on the genetic algorithm for the coevolution of meteorological data (AECMD) was proposed. The algorithm divides the evolutionary population into two subpopulations. One subpopulation quickly guides population evolution through the use of elite-assisted cross-operation, and the other subpopulation maintains population diversity by introducing random populations in the later stages of evolution. This assists in crossover strategies to avoid the impact of random populations on the evolutionary population due to fitness values that are too small. Through the coevolution of the two populations, the entire evolutionary process improves the algorithm's reduction performance.

2. Attribute Reduction in the Rough Set Theory

2.1. Information System. Formally, an information system can be described as follows: Let $S = (U, A, V, f)$ [11–18], where $U = \{x_1, x_2, \dots, x_n\}$ is a nonempty finite set (i.e., the domain $A = C \cup D$ is an attribute set, where C and D represent a conditioned attribute set and decision attribute set, respectively). $V = V_C \cup V_D$, V_C represents the conditioned attribute values, and V_D is a decision attribute value. $f : U \times A \rightarrow V$ is an information function, which gives a value to each object in the information system. In the information system, S represents a cluster of equivalence relations on U . If $B \subseteq S$ and $B \neq \emptyset$, we define $IND(B)$, which is an indiscernibility relation.

$$IND(B) = \bigcap_{b \in B} IND(\{b\}) \quad (1)$$

Obviously, $IND(B)$ is an equivalent relationship.

2.2. Attribute Reduction and Attribute Core. Attribute subsets P and Q are the equivalence relation clusters on the domain U . C and D are the conditional attribute sets and decision attribute sets, respectively, and $P \subseteq C$, $Q \subseteq D$. Q' 's positive region is recorded as $POS_P(Q)$ [7, 19–29]:

$$POS_P(Q) = \bigcup_{X \in U/Q} P_-(X) \quad (2)$$

If $POS_P(Q) = POS_{(P-\{a\})}(Q)$, $a \in P$, then it is said that a can be saved in Q ; otherwise, a is necessary in Q .

In the attribute subset P , all sets of the Q necessary relations in the knowledge base are called Q cores of P , as $CORE_Q(P)$. In the information system, if the attribute subset B is relative to an independent D , and $POS_B(D) = POS_C(D)$, then B is a D relative reduction in C , and the collection of all C' 's D -reductions is denoted as $RED_D(C)$. Because of $CORE_Q(P) = \cap RED_Q(P)$, therefore the attribute core is the intersection of all reductions and cannot be deleted in the attribute reduction process of the information system. Thus, the core of the information system can be regarded as the core of the attribute reduction [30].

2.3. Attribute Independence Degree. Let the information system $S = (U, A, V, f)$, where R is an equivalence relation cluster on U and $P, Q \subseteq R$; then define the dependence degree of Q on P as [31]

$$\gamma(P, Q) = \gamma_P(Q) = \frac{|POS_P(Q)|}{|U|} \quad (3)$$

\parallel denotes the base of the set; $POS_P(Q)$ denotes the positive P of Q in the universe U .

3. Coevolutionary Reduction in the Adaptive Genetic Attribute

The number of populations during evolution is limited. After several iterations, the population is composed of individuals with higher fitness values. At this time, the diversity of the population is low, which makes the selection and crossover operators lose their primary roles. In the process of evolution, different operators have different effects on population diversity. In the process of the selected operator iteration, the population evolution phenomenon of "survival of the fittest" is embodied, but it reduces the diversity of the population, the crossover operator keeps the diversity of the population, and the mutation operator improves the diversity of the species [32].

3.1. Adaptive Genetic Algorithm

3.1.1. Fitness Function. The fitness function is a method that calculates the individual's ability to adapt to the surrounding environment. It is a key step in calculating the degree of an individual's superiority and inferiority. It is also a key process in combining the genetic algorithm with the attribute reduction of the rough set. The purpose of attribute reduction is to remove redundant attributes as much as possible to

obtain an optimal solution. Therefore, the design of the fitness function should meet the two requirements of strong classification ability and deletion of redundant attributes as much as possible. For this reason, the individual degree of attribute dependency and individuals with conditional attributes are introduced as parameters into the fitness function. The fitness function formula is as follows (4):

$$F(x) = (1 - \lambda) \frac{N - R_x}{N} + \lambda * r_x^\alpha(D) \quad (4)$$

where N represents the number of conditional attributes and R_x represents the number of individuals whose gene value is "1" in x ; $r_x^\alpha(D)$ is the attribute dependency of D in the individual s whose gene value is "1"; and $\lambda \in (0, 1)$ is the adjustment parameter.

3.1.2. Selecting the Operator. Selecting operators is a key factor in the reduction in population diversity, which reflects the evolutionary direction of the survival of the fittest and determines the search performance of the algorithm [33]. Roulette strategy determines the selection probability according to the fitness of an individual. At the beginning of the iteration, individual differences are greater, and the diversity of the population is abundant. Through this method, the evolutionary phenomenon of the survival of the fittest can be well represented. However, as the iteration progresses, the individual fitness value of the population decreases, and the performance of the selection operator is also greatly weakened. Therefore, the selected operator is improved in this paper, as shown in

$$P_i = \frac{f_i - f_{\min}}{\sum_{j=1}^N (f_j - f_{\min})} \quad (5)$$

where f_{\min} represents the minimum fitness value of the population and f_i represents the individual fitness value of the probability of the current selection.

After calculating the individual fitness value of the population, the population is sorted in descending order according to the size of the fitness value. When selecting the operation, the individual fitness value is subtracted from the minimum fitness value in the contemporary population, and the roulette selection operation is performed. After the individual fitness value subtracts the minimum fitness value of the population, the degree of difference between individuals increases, which enriches the population diversity and balances the selection pressure.

3.1.3. Cross and Mutation Operators. The traditional genetic algorithm uses fixed crossover probability and mutation probability, which may lead to slow convergence and premature convergence. Algorithm premature convergence affects the evolution of better individuals; the population tends to become static, with limited population diversity, and causes the crossover and mutation operators to become ineffective. The standard adaptive genetic algorithm measures the individual's superiority and inferiority by comparing the individual fitness with the average fitness value. When the

fitness value is greater than the average fitness value, the individual is considered to be a good individual and has a small probability of crossover and mutation. Premature convergence is caused by the individuals in the population. To avoid the multiplication of individuals being slightly larger than the average fitness value, causing the population to be single, this paper proposes using the average fitness value of an individual that is greater than the average fitness of the population as a measure of individual merit. This standard increases the probability of crossover and mutation in this part of the evolution process and avoids overproliferation. At the same time, from the point of view of the entire iterative process of the algorithm, due to the higher diversity of the population at the beginning of the iteration, the population has a greater probability of crossover and mutation. As the iteration proceeds, the population gradually starts to converge, and the population's crossover and mutation probability also gradually decreases. Based on this, the crossover and mutation operators P_c and P_m , respectively, are improved as follows:

$$P_c = \begin{cases} \frac{\exp(f_{\max} - f - 1)}{1 + \exp(b_1 \cdot t)} + C_1, & f \geq f_{t\max} \\ \frac{1}{1 + \exp(b_1 \cdot t)} + C_1, & f < f_{t\max} \end{cases} \quad (6)$$

$$P_m = \begin{cases} \frac{l_1 \cdot \exp(f - f_{\max})}{1 + \exp(b_2 \cdot t)} + M_1, & f \geq f_{t\max} \\ \frac{l_1}{1 + \exp(b_2 \cdot t)} + M_1, & f < f_{t\max} \end{cases} \quad (7)$$

where f_{\max} represents the maximum fitness value; $f_{t\max}$ represents the average fitness value of the individual whose fitness value is greater than the average fitness value; G represents the evolution algebra; b_1 and b_2 represent the changing curvature of the crossover probability and mutation probability with regard to evolution algebra, respectively; C_1 and M_1 represent the convergence limit of the crossover probability and mutation probability, respectively; and l_1 represents the control factor.

3.1.4. Population Diversity. Population diversity is a prerequisite for the evolution of genetic algorithms. Population diversity directly affects the performance of the algorithm. If the population P is binary coded, the population size is n , and the total gene length is L , the population diversity measure is defined as follows:

$$\begin{aligned} \text{div}(P(t)) &= 1 - \frac{1}{L * n} \\ &\cdot \sum_{j=1}^L \left(\max \left\{ \sum_{j=1}^n a_{ij}, \sum_{j=1}^n (1 - a_{ij}) \right\} \right. \\ &\quad \left. - \min \left\{ \sum_{j=1}^n a_{ij}, \sum_{j=1}^n (1 - a_{ij}) \right\} \right) \end{aligned} \quad (8)$$

Input: $S = (U, A, V, f), C_1, M_1, I_1, M_1,$

Output: Attribute reduction red

1. Initialization:

1.1 Calculate the dependency $r_C^\alpha(D)$ of the decision attribute D on the condition attribute C according to formula (3).

1.2 Let $Core(C) = \emptyset$, for any attributes $a_k \in C$, if $r_{C-\{a_k\}}^\alpha(D) - r_C^\alpha(D) \neq 0$, then $Core(C) = Core(C) \cup \{a_k\}$; if $r_{Core(C)}^\alpha(D) = r_C^\alpha(D)$, then $Core(C)$ is the minimum reduction in the base attribute D for the condition attribute C ; otherwise, step 3 is performed.

1.3 For any attributes $a_k \in C$, if $a_k \in Core(C)$, then the corresponding chromosomal gene position is 1; else, the random selection of 0 and 1 as their chromosomal gene is performed.

2. Start the iterative process:

2.1 According to formula (3), calculate the individual attribute dependency value, calculate the individual fitness value by formula (4), and then sort the population in descending order according to the size of the fitness value.

2.2 Select the first M different individuals to compose the elite library $Elite(t)$ and let $t = 0$; from the population $P(t)$, select the $n/4$ individuals to form subpopulations A and B according to formula (5).

2.3 Subpopulation A undergoes evolutionary operations:

(1) The elite algorithm assists in the crossover and randomly selects the elite individuals in the elite bank and the individuals in the child population A to complete the collaborative cross-operation

(2) Perform the mutation operation to obtain subpopulation $A(t+1)$.

2.4 Evolution of subpopulation B: If the number of iterations is higher than $t/4$, generate random populations and perform elite assisted crossover operations; otherwise, perform mutation operations to obtain subpopulation $B(t+1)$.

2.5 Combine the populations $A(t+1)$ and $B(t+1)$ to obtain the population $Temp(t+1)$ and calculate the fitness value of the population $Temp(t+1)$.

2.6 If $Temp(t+1)$ has an individual fitness value greater than that of $P(t)$, replace $P(t)$ with the smallest fitness value to obtain $P(t+1)$ and $P(t+1)$ sorted in descending order. Take the first M different individuals in $P(t+1)$

to update the elite library and get $Elite(t+1)$;

2.7 It is determined whether the termination condition is satisfied. If it is satisfied, it ends; otherwise, start over at 2.1.

ALGORITHM 1: Attribute reduction based on the genetic algorithm for the coevolution of meteorological data (AECMD).

where $\sum_{j=1}^n a_{ij}$ and $\sum_{j=1}^n (1 - a_{ij})$, respectively, represent the number of the 1 and 0 loci of all individuals in the binary coding group; $\max\{\sum_{j=1}^n a_{ij}, \sum_{j=1}^n (1 - a_{ij})\} - \min\{\sum_{j=1}^n a_{ij}, \sum_{j=1}^n (1 - a_{ij})\}$ indicates the distribution of the L gene, and 0 and 1 are indicated in the population.

3.1.5. Elite Individuals Assist in Cross-Operation. This paper is inspired by [34] to design a new crossover algorithm assisted by elite individuals. Reference [34] uses the same elite individuals to cross-operate with crossover individuals. This kind of operation can quickly lead the population towards elite individuals, but it also greatly reduces population diversity. The crossover operation of this paper is to randomly select an elite individual E_i from the elite pool and complete the crossover operation with the cross individuals to avoid the individual elite individuals from guiding to reduce the diversity of the population. The following formula shows:

$$\begin{aligned} x_i &= (x_1, x_2, \dots, x_k, \dots, x_{k+i} \dots x_L) \\ e_j &= (e_{j1}, e_{j2}, \dots, e_{jk}, \dots, e_{jk+i} \dots e_{jL}) \\ &\Downarrow \\ x' &= (x_1, x_2, \dots, e_{jk}, \dots, e_{jk+i} \dots x_L) \end{aligned} \quad (9)$$

3.2. Cooperative Evolution of the Adaptive Genetic Attribute Reduction Algorithm. The attribute dependence degree, as the basis of measuring the importance of the condition attributes in the information system to the decision attribute, provides a standard measure for evaluating the importance of the conditional attributes in the information system. As a self-organized global optimization search algorithm, the genetic algorithm improves the convergence speed and optimization efficiency of the algorithm. The fitness function is used to connect attribute reduction with the genetic algorithm. The number of attributes and the attribute dependency are introduced into the function, which explains the concept of attribute reduction in the rough set. The algorithm for the interval value attribute reduction based on the genetic algorithm is shown in Algorithm 1.

4. Experiments and Results

4.1. Experimental Data. To reduce the influence of the region on precipitation, the experimental data (i.e., annual daily values) were selected from 5 meteorological stations in Huaian (58145), Yancheng (58151), Suqian (58131), Yangzhou (58245), and Lianyungang (58044) from 2005 to 2014, and the amount of effective data was 16750. In addition to the site number, latitude, longitude, time, and other attributes,

TABLE 1: Meteorological attribute identification.

Attribute name	Identification	Attribute name	Identification
air temperature	1	cloudiness	11
air humidity	2	evaporation	12
pressure	3	wind direction	13
wind speed	4	cloud height	14
solar radiation	5	dew point	15
light intensity	6	3 h surface isobaric	16
visibility	7	6 h precipitation	17
surface temperature	8	cloud type	18
soil temperature	9	sunshine	19
soil moisture	10	precipitation	decision attribute

TABLE 2: Experimental variable table.

Variable name	Pc1	Pc2	Pm1	Pm2	α	λ
Variable value	0.9	0.6	0.1	0.001	0.7	0.35

TABLE 3: Precipitation equivalent partition table.

Identification	Precipitation (mm)	Level	Value of decision attribute
R_0	0	No rain	0
R_1	0.1-9.9	Light rain	1
R_2	10.0-24.9	Moderate rain	2
R_3	25.0-49.9	Heavy rain	3
R_4	50.0-99.9	Storm rainfall	4
R_5	100	Heavy rainstorm	5

the data set also included 20 attributes, such as temperature, humidity, and pressure, of which precipitation was a decision attribute, and the rest were conditional attributes. The corresponding classification of conditional attributes, experimental variables, and precipitation levels is shown in Tables 1, 2, and 3, respectively.

4.2. Results Analysis. Since attribute reduction is based on discrete data, the ACIM algorithm is used to discretize the meteorological data and reduce operations.

4.2.1. Performance Analysis of the Improved Genetic Algorithm. The AECMD algorithm improves individual metrics in the evolution process and avoids population diversity imbalance in the evolutionary process by introducing a random population. To analyze the variation in the population diversity in the evolutionary process of the AECMD algorithm, the precipitation attributes are reduced by the AECMD algorithm and the attribute reduction algorithm based on the adaptive genetic algorithm (ARAGA). To directly analyze the diversity of the algorithm, the termination condition is set to meet the maximum number of iterations. A piece is set to meet the maximum number of iterations. Because the initial

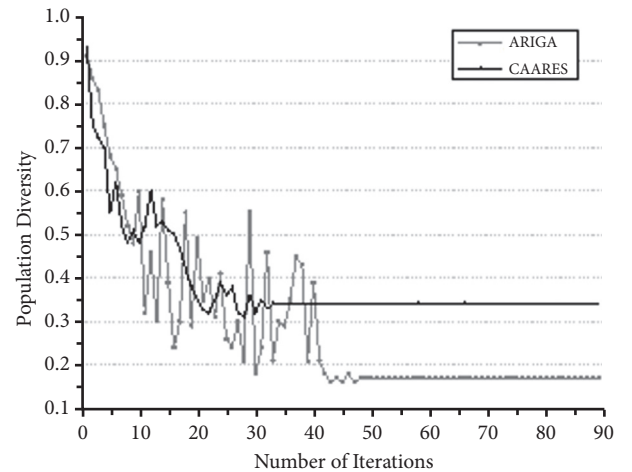


FIGURE 1: Change process of the population diversity.

population is randomly generated, it is not guaranteed that the initial population of the algorithm is the same, and the evolutionary processes of the two attribute reductions are the same. The diversity in the iteration is shown in Figure 1.

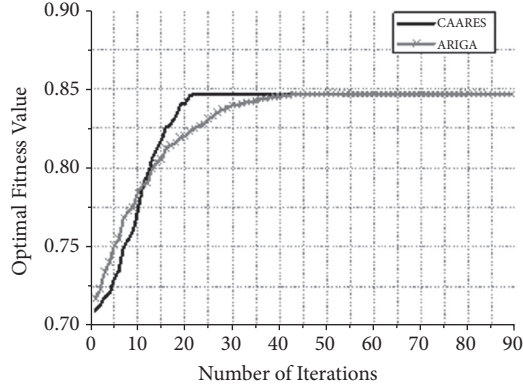


FIGURE 2: Optimal adaptation process.

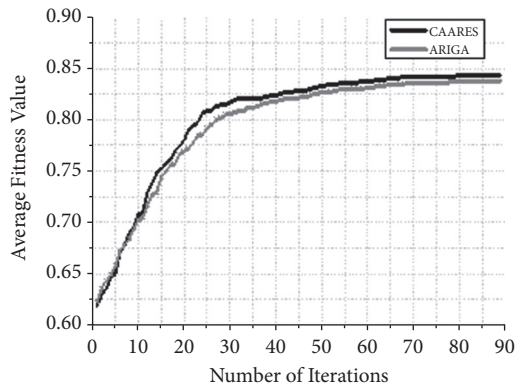


FIGURE 3: Change process of the average fitness.

From Figure 1, we can see that, in the initial stage of the iteration, the population diversity of the two algorithms is more abundant, and then it begins to decrease rapidly. With the evolution, the ARAGA algorithm can improve the diversity of the population, but the diversity of the population is unstable. Finally, the diversity of the population can be maintained at a lower level, and the search performance of the algorithm needs to be improved. In comparison, the diversity of the AECMD algorithm is relatively stable, and the diversity of the population can be maintained at a high level, which effectively avoids premature convergence of the algorithm and ensures the convergence performance of the algorithm. The elite individual plays an important role in the evolution of the population. To analyze the convergence performance of the AECMD algorithm using the genetic algorithm, the changes in the optimal individual and the average fitness value in the iterative process of the AECMD and ARAGA algorithms are recorded. The two changes are shown in Figures 2 and 3, respectively.

Because the initial population is different, the initial optimal individuals are not necessarily the same. From Figure 2, we can see that the AECMD algorithm starts to converge in the twenty-second generation, and the ARAGA algorithm begins to converge after 42 times and then converges slowly. From the change curve of the average fitness

TABLE 4: Average optimal subset number.

Algorithm	TSDPSO-AR	ARAGA	AECMD
Number	5.7	5.3	5

value, it is found that the initial average fitness of the ARAGA algorithm is higher at the beginning but, with the evolution, the advantages of the AECMD algorithm gradually become prominent. The evolution mechanism based on the elitist strategy accelerates the convergence speed of the AECMD algorithm.

4.2.2. Reduction Performance Analysis. To further analyze the effect of the AECMD algorithm on the performance of meteorological data reduction, the precipitation properties of the data set are reduced with the rough set attribute reduction algorithm based on the Tabu Discrete Particle Swarm Optimization [10, 35–39] (TSDPSO-AR) algorithm. To compare the performance of reduction, the classification of every reduction attribute subset is carried out in the KNN (k-Nearest Neighbor, $K=3$) classifier. The training data and test data were carried out at a 4:1 ratio. To avoid contingency, cross-operations are used to calculate the accuracy of the precipitation prediction. The average optimal subset and prediction results are shown in Tables 4 and 5, respectively.

Combined with Tables 3 and 4, we can find that the classification ability of the AECMD algorithm is stronger than that of the other two algorithms. The AECMD algorithm improves the search ability of the algorithm by improving the genetic operators and improves the convergence speed of the algorithm with an elitist strategy. It can also be found that the accuracy of precipitation prediction is high for the no rain and light rain categories, and the accuracy of the prediction decreases with the increase in rainfall level. This is due to the uneven data distribution for different levels of rainfall. With the increase in rainfall grade, the number of corresponding samples is greatly reduced. At this time, a wrong prediction may have a greater impact on the accuracy of the classification; therefore, the corresponding prediction accuracy is also low.

5. Conclusion

In this paper, the crossover operator and mutation operator of the adaptive prediction algorithm are improved, and the evolutionary population is divided into two subgroups. One subpopulation improves the convergence speed by using the elite-assisted cross-operation. The other subpopulation maintains the population diversity in the evolutionary process by introducing a random population, and the two subpopulations are coevolved and complete the iterative operation. Finally, an elitist strategy based on the coevolution mechanism of the genetic algorithm, combined with attribute reduction, is used to complete the precipitation reduction operation and improve the reduction performance of the meteorological data.

TABLE 5: Prediction accuracy of precipitation.

Site	TSDPSO-AR	ARAGA	AECMD
R0	82.10.04 (%)	80.40.05 (%)	84.20.03 (%)
R1	78.40.03 (%)	76.20.07 (%)	81.80.04 (%)
R2	54.70.07 (%)	53.60.03 (%)	56.10.02 (%)
R3	47.50.05 (%)	46.90.06 (%)	48.70.03 (%)
R4	42.30.05 (%)	42.10.04 (%)	43.90.07 (%)
R5	26.20.09 (%)	25.60.08 (%)	27.40.06 (%)

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 they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61402236, 61373064, 61472024, and U1433203), the CERNET Innovation Project (NGII20160318), and the Jiangsu Province “Six talent peaks project in Jiangsu Province” (2015-DZXX-015).

References

- [1] W. Tian, Y. Zheng, R. Yang, S. Ji, and J. Wang, “A Survey on Clustering based Meteorological Data Mining,” *International Journal of Grid and Distributed Computing*, vol. 7, no. 6, pp. 229–240, 2014.
- [2] S. Wan, Y. Zhang, and J. Chen, “On the Construction of Data Aggregation Tree with Maximizing Lifetime in Large-Scale Wireless Sensor Networks,” *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7433–7440, 2016.
- [3] Shaohua Wan, “Energy-Efficient Adaptive Routing and Context-Aware Lifetime Maximization in Wireless Sensor Networks,” *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 321964, 16 pages, 2014.
- [4] S. Wan and Y. Zhang, “Coverage hole bypassing in wireless sensor networks,” *The Computer Journal*, vol. 60, no. 10, pp. 1536–1544, 2017.
- [5] Y. Qian, J. Liang, Y. Yao, and C. Dang, “MGRS: a multi-granulation rough set,” *Information Sciences*, vol. 180, no. 6, pp. 949–970, 2010.
- [6] X. Chen, Y. Di, J. Duan, and D. Li, “Linearized compact ADI schemes for nonlinear time-fractional Schrödinger equations,” *Applied Mathematics Letters*, vol. 84, pp. 160–167, 2018.
- [7] F. Wu, X. Cheng, D. Li, and J. Duan, “A two-level linearized compact ADI scheme for two-dimensional nonlinear reaction-diffusion equations,” *Computers & Mathematics with Applications. An International Journal*, vol. 75, no. 4, pp. 2835–2850, 2018.
- [8] J. Huang, P. Zhang, X. Huangfu, and H. Sun, “A trajectory prediction approach for mobile objects by combining semantic features,” *Journal of Computer Research and Development*, vol. 51, no. 1, pp. 76–87, 2014.
- [9] D. Li, Z. Chen, and J. Liu, “Analysis for Behavioral Economics in Social Networks: An Altruism-Based Dynamic Cooperation Model,” *International Journal of Parallel Programming*, 2018.
- [10] R. Zhang G, H. Xiao Hui U, and Y. Zong S, “Rough Set Attribute Reduction Algorithm Based on Tabu Discrete Particle Swarm Optimization,” *4em Journal of Chinese Computer Systems*, 2017.
- [11] W. B. Deng, G. Y. Wang, and F. Hu, “Self-learning model based on dominance-based rough set approach,” *Chinese Journal of Computers. Jisuanji Xuebao*, vol. 37, no. 12, pp. 2408–2418, 2014.
- [12] H. Li, K. Ota, and M. Dong, “Learning IoT in edge: deep learning for the internet of things with edge computing,” *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.
- [13] M. Tao, K. Ota, and M. Dong, “Locating Compromised Data Sources in IoT-enabled Smart City: A Great-Alternative-Region-based Approach,” in *1em plus 0.5em minus 0.4em IEEE Transactions on Industrial Informatics*, Locating Compromised Data Sources in IoT-enabled Smart City, A Great-Alternative-Region-based Approach. 1em plus 0.5em minus 0.4em IEEE Transactions on Industrial Informatics, 2018.
- [14] L. Li, K. Ota, and M. Dong, “When Weather Matters: IoT-Based Electrical Load Forecasting for Smart Grid,” *IEEE Communications Magazine*, vol. 55, no. 10, pp. 46–51, 2017.
- [15] M. Tao, K. Ota, and M. Dong, “Ontology-based data semantic management and application in IoT- and cloud-enabled smart homes,” *Future Generation Computer Systems*, vol. 76, pp. 528–539, 2017.
- [16] T. Kumrai, K. Ota, M. Dong, J. Kishigami, and D. K. Sung, “Multiobjective Optimization in Cloud Brokering Systems for Connected Internet of Things,” *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 404–413, 2017.
- [17] J. Wu, M. Dong, K. Ota, M. Tariq, and L. Guo, “Cross-domain fine-grained data usage control service for industrial wireless sensor networks,” *IEEE Access*, vol. 3, pp. 2939–2949, 2015.
- [18] D. Li and C. Zhang, “Split Newton iterative algorithm and its application,” *Applied Mathematics and Computation*, vol. 217, no. 5, pp. 2260–2265, 2010.
- [19] A. H. Attia, A. S. Sherif, and G. S. El-Tawel, “Maximal limited similarity-based rough set model,” *Soft Computing*, vol. 20, no. 8, pp. 3153–3161, 2016.
- [20] D. Li, C. Zhang, and J. Wen, “A note on compact finite difference method for reaction-diffusion equations with delay,” *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 39, no. 5–6, pp. 1749–1754, 2015.
- [21] D. Li and J. Zhang, “Efficient implementation to numerically solve the nonlinear time fractional parabolic problems on unbounded spatial domain,” *Journal of Computational Physics*, vol. 322, pp. 415–428, 2016.
- [22] D. Li, J. Zhang, and Z. Zhang, “Unconditionally optimal error estimates of a linearized Galerkin method for nonlinear time

- fractional reaction-subdiffusion equations,” *Journal of Scientific Computing*, vol. 76, no. 2, pp. 848–866, 2018.
- [23] J. Wen, Z. Zhou, Z. Liu, M. Lai, and X. Tang, “Sharp sufficient conditions for stable recovery of block sparse signals by block orthogonal matching pursuit,” *Applied and Computational Harmonic Analysis*, 2018.
- [24] J. Wen, J. Wang, and Q. Zhang, “Nearly optimal bounds for orthogonal least squares,” *IEEE Transactions on Signal Processing*, vol. 65, no. 20, pp. 5347–5356, 2017.
- [25] T. Qiu, K. Zheng, H. Song, M. Han, and B. Kantarci, “A Local-Optimization Emergency Scheduling Scheme with Self-Recovery for a Smart Grid,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3195–3205, 2017.
- [26] C. Wang, H. Lin, and H. Jiang, “CANS: Towards Congestion-Adaptive and Small Stretch Emergency Navigation with Wireless Sensor Networks,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 5, pp. 1077–1089, 2016.
- [27] B. Wang, B. Fang, Y. Wang, H. Liu, and Y. Liu, “Power System Transient Stability Assessment Based on Big Data and the Core Vector Machine,” *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2561–2570, 2016.
- [28] T. Qiu, K. Zheng, M. Han, C. L. Chen, and M. Xu, “A Data-Emergency-Aware Scheduling Scheme for Internet of Things in Smart Cities,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2042–2051, 2018.
- [29] J. Wen, B. Zhou, W. H. Mow, and X.-W. Chang, “An efficient algorithm for optimally solving a shortest vector problem in compute-and-forward design,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 10, pp. 6541–6555, 2016.
- [30] Z. Meng and Z. Shi, “On quick attribute reduction in decision-theoretic rough set models,” *Information Sciences*, vol. 330, pp. 226–244, 2016.
- [31] S. Wan, Y. Zhao, T. Wang, Z. Gu, Q. H. Abbasi, and K. R. Choo, “Multi-dimensional data indexing and range query processing via Voronoi diagram for internet of things,” *Future Generation Computer Systems*, vol. 91, pp. 382–391, 2019.
- [32] N. Metawa, M. K. Hassan, and M. Elhoseny, “Genetic algorithm based model for optimizing bank lending decisions,” *Expert Systems with Applications*, vol. 80, pp. 75–82, 2017.
- [33] Deng Li, Liying Qiu, Jiaqi Liu, and Congwen Xiao, “Analysis of Behavioral Economics in Crowdsensing: A Loss Aversion Cooperation Model,” *Scientific Programming*, vol. 2018, Article ID 4350183, 18 pages, 2018.
- [34] J. Liu, N. Zhong, D. Li, and H. Liu, “BMCGM: A Behavior Economics-Based Message Transmission Cooperation Guarantee Mechanism in Vehicular Ad-hoc NETWORKS,” *Sensors*, vol. 18, no. 10, p. 3316, 2018.
- [35] B. Wang, S. Wan, X. Zhang, and K. R. Choo, “A Novel Index for Assessing the Robustness of Integrated Electrical Network and a Natural Gas Network,” *IEEE Access*, vol. 6, pp. 40400–40410, 2018.
- [36] L. He, C. Chen, T. Zhang, H. Zhu, and S. Wan, “Wearable Depth Camera: Monocular Depth Estimation via Sparse Optimization Under Weak Supervision,” *IEEE Access*, vol. 6, pp. 41337–41345, 2018.
- [37] F. Zhu, W. Quan, Z. Zheng, and S. Wan, “A Bayesian Learning Method for Financial Time-Series Analysis,” *IEEE Access*, vol. 6, pp. 38959–38966, 2018.
- [38] B. Mi, D. Huang, S. Wan, L. Mi, and J. Cao, “Oblivious Transfer Based on NTRUEncrypt,” *IEEE Access*, vol. 6, pp. 35283–35291, 2018.
- [39] Bo Mi, Darong Huang, and Shaohua Wan, “NTRU Implementation of Efficient Privacy-Preserving Location-Based Querying in VANET,” *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7823979, 11 pages, 2018.

Research Article

Homomorphic Evaluation of the Integer Arithmetic Operations for Mobile Edge Computing

Changqing Gong ¹, Mengfei Li ¹, Liang Zhao ¹, Zhenzhou Guo,¹ and Guangjie Han ²

¹School of Computer Science and Technology, Shenyang Aerospace University, Shenyang 110136, China

²Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology, Dalian 116024, China

Correspondence should be addressed to Liang Zhao; lzhaosau@sau.edu.cn

Received 26 September 2018; Accepted 31 October 2018; Published 15 November 2018

Guest Editor: Mianxiong Dong

Copyright © 2018 Changqing Gong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of the 5G network and Internet of Things (IoT), lots of mobile and IoT devices generate massive amounts of multisource heterogeneous data. Effective processing of such data becomes an urgent problem. However, traditional centralised models of cloud computing are challenging to process multisource heterogeneous data effectively. Mobile edge computing (MEC) emerges as a new technology to optimise applications or cloud computing systems. However, the features of MEC such as content perception, real-time computing, and parallel processing make the data security and privacy issues that exist in the cloud computing environment more prominent. Protecting sensitive data through traditional encryption is a very secure method, but this will make it impossible for the MEC to calculate the encrypted data. The fully homomorphic encryption (FHE) overcomes this limitation. FHE can be used to compute ciphertext directly. Therefore, we propose a ciphertext arithmetic operation that implements data with integer homomorphic encryption to ensure data privacy and computability. Our scheme refers to the integer operation rules of complement, addition, subtraction, multiplication, and division. First, we use Boolean polynomials (BP) of containing logical AND, XOR operations to represent the rulers. Second, we convert the BP into homomorphic polynomials (HP) to perform ciphertext operations. Then, we optimise our scheme. We divide the ciphertext vector of integer encryption into subvectors of length 2 and increase the length of private key of FHE to support the 3-multiplication level additional. We test our optimised scheme in DGHV and CMNT. In the number of ciphertext refreshes, the optimised scheme is reduced by 2/3 compared to the original scheme, and the time overhead of our scheme is reduced by 1/3. We also examine our scheme in CNT of without bootstrapping. The time overhead of optimised scheme over DGHV and CMNT is close to the original scheme over CNT.

1. Introduction

With the rapid development of the 5G network and Internet of Things (IoT), mobile devices and IoT devices are more convenient to access the internet and generate massive amounts of data. Since these data come from edge network devices, traditional centralised cloud computing models are difficult to process these multisource heterogeneous data quickly and efficiently. If we migrate some of the features of cloud computing to an edge network as [1–3], it will be beneficial to data collection and calculation. Therefore, mobile edge computing (MEC) emerges as the above requires. Edge computing [4] is a method of optimizing applications or cloud computing systems by taking some portion of an application, its data, or

services away from one or more central nodes (the “core”) to the other logical extreme (the “edge”) of the internet which contacts with the physical world or end users. In one vision of this architecture, specifically for IoT devices, data comes in from the physical world via various sensors, and actions are taken to change physical state via various forms of output and actuators; by performing analytics and knowledge generation at the edge, communications bandwidth between systems under control and the central data centre is reduced. The MEC is to put any computer program that needs low latency nearer to the requests in particular for mobile networks such as 5G. MEC allows terminal devices to migrate storage and computing tasks to network edge nodes. The architecture of edge computing is shown in Figure 1.

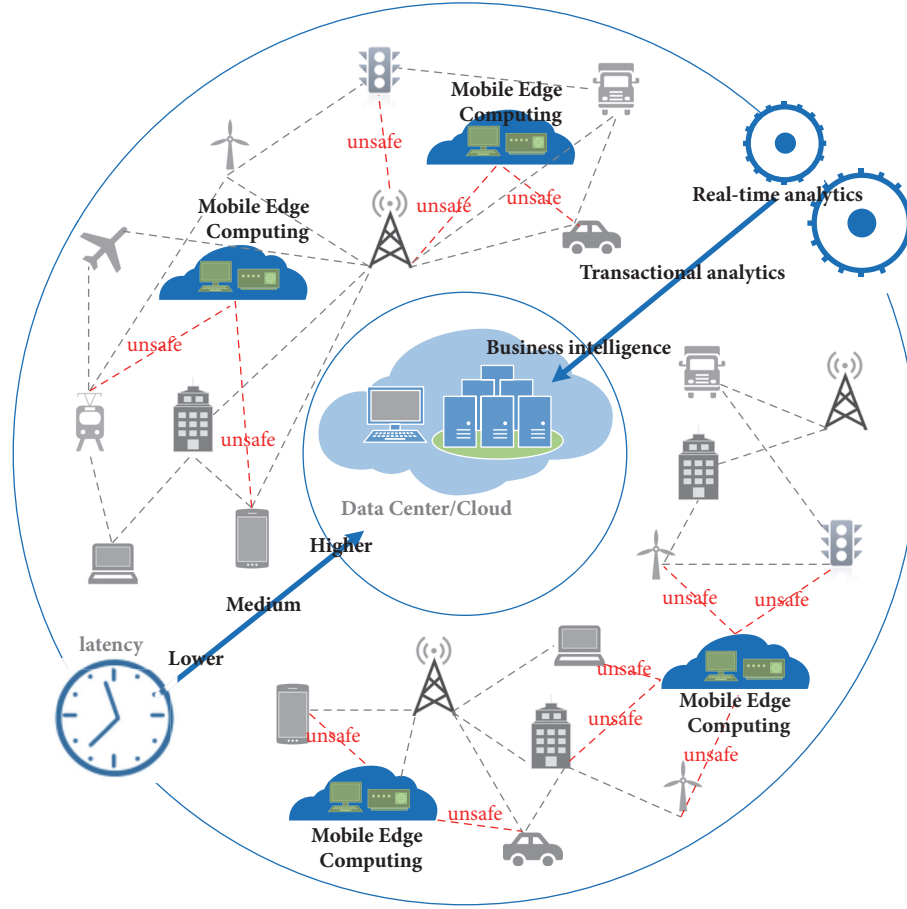


FIGURE 1: The architecture of mobile edge computing.

MEC covers a wide range of technologies including wireless sensor networks, mobile data acquisition, mobile signature analysis, cooperative distributed peer-to-peer ad hoc networking, and processing. The foundation of MEC is traditional network, wireless network, mobile network, and Internet of Vehicles (IoV), in which a large number of network infrastructure technologies and services are applied. Include energy efficiency and spectral efficiency tradeoff in device-to-device [5], QoS-aware interdomain multicast [6], efficient cross-layer relay node selection model [7], mobile anchors assisted localization [8], and vehicular communications [9]. Mobile edge computing (see Figure 1) is a service for multiple entities (mobile devices and IoT devices). Many entities put their real-time data (outsourced data in cloud computing) on mobile edge computing for analysis or storage. The extensive features of mobile edge computing such as data collection, real-time analytics, and parallel processing make the privacy issues that exist in the cloud computing environment become more prominent. Therefore, the security of outsourced data remains a fundamental issue for data security of MEC. The red dotted lines (see Figure 1) represent safety risk between the different entities and mobile edge computing, such as IoV and smart home services provided by mobile edge computing. In IoV, the location information of the vehicle and the online browsing records generated by

the owner are collected by the mobile edge calculation, so as to feed back the precise navigation information of the vehicle, and the push information for the owner's preferences. In this process, the data of the owner and the car are exposed to the mobile edge computing, which is extremely unsafe. In smart home, many sensors monitor environment changes in the room, such as temperature, humidity, surveillance video, etc., through mobile edge calculation storage and analysis, giving the best home environment settings. This process is also unsafe. Traditional methods, including intrusion detection, access control, and virtual isolation, can only protect data from being stolen by external attackers. For internal attacks (honest and curious mobile edge computing), data security is still not guaranteed. The encrypted data are relatively considered a safe storage status. However, it is unable to satisfy the computability of ciphertext data. If we want the edge computing to be able to do nontrivial computations with the ciphertext data, and therefore the problem is severe to solve. The nontrivial computations include deep learning for the IoV with Edge Computing [10], offloading computation for MEC [11]. The fully homomorphic encryption (FHE) overcomes this limitation. Gentry described the first encryption scheme that supports both addition and multiplication base on ciphertexts, i.e., FHE scheme [12]. The FHE scheme allows anyone to perform arbitrary computations

on encrypted data, despite not having the secret decryption key.

The development of FHE can be partitioned into three generations [13]. The first generation includes Gentry's original scheme using ideal lattices [12], the somewhat simpler scheme of van Dijk et al. [14], and some optimisations for first generation in public key size [15–17]. All these schemes have a problem of rapidly growing noise, which affected both efficiency and security. The second generation begins with Brakerski-Vaikuntanathan [18, 19] and Brakerski et al. [20] and is characterized by modulus-switching and key-switching techniques for controlling the noise, resulting in improved efficiency, including LWE [18], Ring-LWE [19], and NTRU [21, 22] hardness assumption. The third generation begins with the scheme of Gentry et al. [23]. Third-generation schemes are usually slightly less efficient than second-generation ones, but they can be based on somewhat weaker hardness assumptions.

In the homomorphic evaluation of FHE, different circuits in the real world are realized by multiplication homomorphic, addition homomorphic, and decryption homomorphic (ciphertext refresh). The application of FHE is ciphertext arithmetic operation and ciphertext retrieval. Gentry, Halevi, and Smart propose the first evaluation of a complex circuit, i.e., a full AES-128 block evaluation [24] by using a BGV [20] style scheme. The scheme makes use of batching [25, 26], key switching, and modulus switching techniques to obtain an efficient levelled implementation. Chen Y. et al. [27] propose the integer arithmetic over ciphertext and homomorphic data aggregation by using a BGV style scheme. The scheme uses the HELib library to implement homomorphic evaluation for addition, subtraction, multiplication, and division of unsigned integers. Gai K. et al. [28] propose the blend arithmetic operations on tensor-based FHE over real numbers and proposes a novel tensor-based FHE solution [29]. Yang J. et al. [30] propose the secure tensor decomposition using FHE scheme.

In [24], AES-128 block evaluation cannot implement carry operation. In [27], the integer arithmetic over ciphertext just implement 2-4 bits arithmetic operation of unsigned integer. In [28], the blend arithmetic operations do not implement division operation over tensor-based FHE over real numbers. Therefore, the [24, 27, 28] cannot be used as a complete homomorphic evaluation of a signed integer arithmetic operations scheme by using addition and multiplication homomorphism in MEC. The homomorphic evaluation of integer arithmetic operations is an important foundation for nontrivial computations with the ciphertext data. Therefore, it is significant to construct homomorphic evaluation of integer arithmetic operations scheme.

Contribution. We propose the homomorphic evaluations of integer arithmetic operations base on DGHV [14] and its variants [15, 31, 32] in mobile edge computing. And we use the features of homomorphic encryption to prevent sensitive data from being stolen. At the same time, we can also calculate ciphertext data in mobile edge computing.

- (i) Our scheme refers to the integer operation rules which are expressed in Boolean polynomials (BP) that only contains logical AND, XOR operations. Then, we convert BP into homomorphic polynomials (HP) by using addition and multiplication on ciphertexts.
- (ii) We propose judgment choose Boolean polynomials (JCBP) to solve the constantly choose problem in the multiplication and division. Then we convert JCBP into judgment choose homomorphic polynomials (JCHP) by using addition and multiplication on ciphertexts.
- (iii) We optimise the process of homomorphic evaluation of integer arithmetic operations. We divide the ciphertext vector of integer encryption into subvectors of length 2 and add the length of the private key of FHE to support the 3-multiplication level additional, except for the 15-multiplication level required by bootstrapping.
- (iv) We test the optimized scheme in DGHV [14] and CMNT [17]. In the number of ciphertext refreshes, the optimized scheme is reduced by 2/3 compared to the original scheme, and the homomorphic evaluation time overhead of integer arithmetic operations is reduced by 1/3. We also test our scheme in CNT [31] of without bootstrapping. The time overhead of optimized scheme over DGHV [14] and CMNT [17] is close to the original scheme over CNT [31].

Organization. In Section 2, we will introduce DGHV [14], the variants of DGHV [17, 31, 32] and some homomorphic evaluation in detail. In Section 3, we modify the polynomials of integer arithmetic operation according to the computation process of complement, addition, subtraction, multiplication, and division of integer in the computer. We also propose the BP of integer arithmetic operation and convert the BP into the HP of integer arithmetic operation. In Section 4, we analyse the noise ceiling and optimize the process of homomorphic evaluation of integer arithmetic operations. In Section 5, we show our implementation and experimental result. We show the efficiency and conclusion of our scheme.

2. Related Work

Fully homomorphic encryption scheme over the integers and its variants are an essential branch of homomorphic encryption research. Also, we propose the homomorphic evaluations of integer arithmetic operations base on DGHV and its variants. We use DGHV and its variants to encrypt data and upload ciphertext data to a MEC data center. The server of MEC can process ciphertext data by using features of homomorphic encryption. Our scheme involves homomorphic encryption and ciphertext computing. Therefore, we will introduce the original homomorphic encryption scheme DGHV [14] and its variants on integer, including shorter public keys CMNT [17], public keys compress and modulus switching CNT [31], batch encryption CCKL+

[32]. Below we will replace [14, 17, 31, 32] with DGHV, CMNT, CNT, and CCKL+. At the same time, we will also introduce some ciphertext computing techniques related to our scheme, including full AES-128 block evaluation [24] by using a BGV style scheme, integer arithmetic over ciphertext and homomorphic data aggregation [27], the blend arithmetic operations on tensor-based FHE over real numbers [28].

2.1. DGHV and Variants Scheme. The DGHV scheme is described by van Dijk et al. based on integer and to simplify [12]. The advantages of DGHV include simple encryption process and being easy to understand. However, it has a weakness that the noise in ciphertext increases rapidly with the multiplicative level increases. The scheme is based on a set of public integers: $x_i = q_i p + r_i$, $0 \leq i \leq \tau$, where the integer p is secret. We use the same notation as in DGHV. The DGHV use the following parameters (all polynomial in the security parameter λ):

- (i) γ is the bit-length of the x_i 's.
- (ii) η is the bit-length of private key p .
- (iii) ρ is the bit-length of the noise r_i .
- (iv) τ is the number of x_i 's in the public key.
- (v) ρ' is used for encryption.

For a specific η -bit odd integer p , DGHV use the following distribution over γ -bit integers:

$$\mathcal{D}_{\gamma, \rho(p)} = \left\{ \text{Choose } q \leftarrow \mathbb{Z} \cap \left[0, \frac{2^\gamma}{p}\right), r \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho) : \text{Output } x = q \cdot p + r \right\} \quad (1)$$

KeyGen(λ). Generate a random odd integer p of size η bits as a sk. For the public key, sample $x_i \leftarrow \mathcal{D}_{\gamma, \rho(p)}$ for $0 \leq i \leq \tau$. Relabel so that x_0 is the largest. Let x_0 be odd and $[x_0]_p$ even. Let $pk = \langle x_0, x_1, \dots, x_\tau \rangle$ and $sk = p$.

Encrypt($pk, m \in \{0, 1\}$). Choose a random subset $S \subseteq \{1, 2, \dots, \tau\}$ and a random integer $r \in (-2^\rho, 2^\rho)$, and output $c \leftarrow [m + 2r + 2 \sum_{i \in S} x_i]_{x_0}$.

Decrypt(sk, c). Output $m' = ((c) \bmod p) \bmod 2 = (c \bmod 2)(\lfloor c/p \rfloor \bmod 2)$.

Evaluate($pk, C, c_1, c_2, \dots, c_t$). Given the function F with t input, and t ciphertexts c_i , convert logic AND and logic XOR of F into addition and multiplication, performing all the addition and multiplication, and return the resulting integer.

The following parameter set is suggested in DGHV: $\rho = \lambda$, $\rho' = 2\lambda$, $\eta = \tilde{\mathcal{O}}(\lambda^2)$, $\gamma = \tilde{\mathcal{O}}(\lambda^5)$, $\tau = \gamma + \lambda$. The public key size is then $\tilde{\mathcal{O}}(\lambda^{10})$.

The CMNT scheme is described by Coron et al. to reduce the public key size of the DGHV scheme from $\tilde{\mathcal{O}}(\lambda^{10})$ down to $\tilde{\mathcal{O}}(\lambda^7)$. CMNT scheme applies a new parameter β by the form $x'_{ij} = x_{i,0} \cdot x_{j,1} \bmod x_0$, $1 \leq i, j \leq \beta$ to

generate the $\tau = \beta^2$ integers x'_{ij} used for encryption. CMNT scheme enables reducing the public key size from τ down to roughly $2\sqrt{\tau}$ integers of bits. CMNT scheme uses an error-free x_0 , that is, $x_0 = q_0 p$, since otherwise, the error would grow too large. Additionally, for encryption CMNT scheme consider a linear combination of the x'_{ij} with a coefficient vector $\mathbf{b} = (b_{i,j})$ instead of bits; this enables reducing the public key size further. The vector \mathbf{b} with components in $[0, 2^\alpha]$.

The CMNT scheme takes $\rho = \lambda$, $\eta = \tilde{\mathcal{O}}(\lambda^2)$, and $\gamma = \tilde{\mathcal{O}}(\lambda^5)$ as in the DGHV scheme. However, it takes $\alpha = \lambda$, $\beta^2 = \tilde{\mathcal{O}}(\lambda^2)$, and $\rho' = 4\lambda$. The main difference is that instead of having $\tau = \tilde{\mathcal{O}}(\lambda^5)$ integers x_i 's, CMNT scheme has only $2\beta = \tilde{\mathcal{O}}(\lambda^2)$ integers x_i . Hence the public key size becomes $\tilde{\mathcal{O}}(\lambda^7)$ instead of $\tilde{\mathcal{O}}(\lambda^{10})$.

Coron and Naccache et al. describe the CNT scheme. The CMT describes a method that can compress the public key size of the DGHV scheme and optimise the noise management technique by modifying the modulus switching technology [20]. The noise ceiling of CNT scheme increases only linearly with the multiplicative level instead of exponentially. So, a levelled DGHV variant was implemented. This scheme gives two optimisations for homomorphic encryption on DGHV as below.

The first optimisation is public keys compression. First generate the secret key p of size η bits and use a pseudo-random function f with random seed se to generate a set of $\chi_i \in [0, 2^\gamma]$, $1 \leq i \leq \tau$. Finally, compute δ_i that $x_i = \chi_i - \delta_i$ is small modulo p and store δ_i in the public key, instead of the full x_i 's.

The second optimisation is Modulus-Switching Technique. The CMNT show how to adapt Brakerski, Gentry, and Vaikuntanathan's (BGV) FHE framework [20] to the DGHV scheme over the integers. Under the [20] framework, the noise ceiling increases only linearly with multiplicative depth, instead of exponentially.

The CNT scheme takes $\rho = \lambda$, $\eta = \tilde{\mathcal{O}}(\lambda^2)$, $\gamma = \tilde{\mathcal{O}}(\lambda^5)$, $\alpha = \tilde{\mathcal{O}}(\lambda^2)$, $\tau = \tilde{\mathcal{O}}(\lambda^3)$, and $\rho' = \tilde{\mathcal{O}}(\lambda^2)$. The new public key of CNT scheme has size $\gamma + \tau \cdot (\eta + \lambda) = \tilde{\mathcal{O}}(\lambda^5)$ instead of $\tilde{\mathcal{O}}(\lambda^{10})$ of DGHV.

J.H. Cheon et al. describe the CCKL+ scheme. It extends DGHV to support the same batching capability as in RLWE-based schemes [20, 25], and to homomorphically evaluate a full AES circuit with roughly the same level of efficiency as [24]. The CCKL+ scheme is a merger of two independent works [33, 34] built on the same basic idea but with different contributions. Its security under the (stronger) Error-Free Approximate-GCD assumption already DGHV, CMNT, and CNT.

The CCKL+ scheme extends the DGHV scheme by packing ℓ plaintexts $m_0, \dots, m_{\ell-1}$ into a single ciphertext, using the Chinese Remainder Theorem (CRT). For somewhat homomorphic encryption, this allows us to encrypt not only bits but elements from rings of form \mathbb{Z}_Q . The CKLL+ scheme takes $\rho = 2\lambda$, $\eta = \tilde{\mathcal{O}}(\lambda^2)$, $\gamma = \tilde{\mathcal{O}}(\lambda^5)$, $\alpha = \tilde{\mathcal{O}}(\lambda^2)$, and $\tau = \tilde{\mathcal{O}}(\lambda^3)$ as in CNT scheme, with $\rho' = \tilde{\mathcal{O}}(\lambda)$, $\alpha' = \tilde{\mathcal{O}}(\lambda^2)$, and $\ell = \tilde{\mathcal{O}}(\lambda^2)$.

TABLE 1: The parameters of DGHV, CMNT, CNT and CCKL+, and public key storage size.

FHE scheme	λ	ρ'	η	γ	pk length	pk size
DGHV	λ	2λ	$\bar{\mathcal{O}}(\lambda^2)$	$\bar{\mathcal{O}}(\lambda^5)$	$\bar{\mathcal{O}}(\lambda^{10})$	41 GB
CMNT	λ	4λ	$\bar{\mathcal{O}}(\lambda^2)$	$\bar{\mathcal{O}}(\lambda^5)$	$\bar{\mathcal{O}}(\lambda^7)$	800 MB
CNT	λ	$\bar{\mathcal{O}}(\lambda^2)$	$\bar{\mathcal{O}}(\lambda^2)$	$\bar{\mathcal{O}}(\lambda^5)$	$\bar{\mathcal{O}}(\lambda^5)$	10.1 MB
CCKL+	λ	3λ	$\bar{\mathcal{O}}(\lambda^2)$	$\bar{\mathcal{O}}(\lambda^5)$	$\bar{\mathcal{O}}(\lambda^8)$	5.6 GB

We can conclude (see Table 1) that the CNT scheme performs best in public key storage and only 10.1 MB. We test public key size of DGHV by using “large” parameters of CMNT, and up to 41 GB. The CNT scheme has better noise management technique, in which the noise ceiling increases only linearly with the multiplicative level. Therefore, the CNT scheme supports more multiplication level than other schemes. The CCKL+ scheme implements batch FHE scheme to encrypt a plaintext vector to a ciphertext by using the CRT. However, each one of the components in the plaintext vector is required to be independent. If we encrypt a plaintext vector, the encryption result is a plaintext vector using DGHV, CMNT, and CNT scheme, and the encryption result is a ciphertext by using CCKL+. Therefore, when we do arithmetic operations on the ciphertext of a plaintext vector, we can perform carry operation by using DGHV, CMNT, and CNT scheme, instead of using CCKL+.

2.2. Homomorphic Evaluation. The advantage of homomorphic evaluation is implementing various operations in the real world on ciphertext and is not only multiplication homomorphic, addition homomorphic, and decryption homomorphic (ciphertext refresh). The FHE abstracts various operations of the real world as a collection of circuits consisting of logical XOR and logical AND. The homomorphic evaluation is to implement different circuits in this collection of circuits. Below we will introduce some relevant schemes for homomorphic evaluation of arithmetic operations.

Gentry, Halevi, and Smart propose the first evaluation of a complex circuit, i.e., a full AES-128 block evaluation [24] by using a BGV [20] style scheme. The scheme makes use of batching [25, 26], key switching, and modulus switching techniques to obtain an efficient levelled implementation. After that, Gentry, Smart, and Halevi publish significantly improved runtime results. Compared to the earlier implementation [21], Gentry et al. use the latest version of the HELib library [35]. Two variations of the implementation are reported: one with bootstrapping and one without bootstrapping.

Chen Y. et al. [27] propose the integer arithmetic over ciphertext and homomorphic data aggregation by using a BGV [20] style scheme. The scheme uses the HELib library [35] to implement homomorphic evaluation for addition, subtraction, multiplication, and division of unsigned integers. However, the scheme report time overhead of integer arithmetic over ciphertext without bootstrapping and modulus switching (somewhat homomorphic encryption). The length of integer ciphertext only sets 2, 3, 4 bits and sets 128 security level to guarantee right result. The

scheme does not optimise the operations of integer arithmetic over ciphertext with bootstrapping and modulus switching.

Gai K. et al. [28] propose the blend arithmetic operations on tensor-based FHE over real numbers and propose a novel tensor-based FHE solution [29]. The scheme uses tensor laws to carry the computations of blend arithmetic operations over real numbers. However, blend arithmetic operations only include addition and multiplication. The scheme does not implement blend arithmetic operations with the division.

3. Homomorphic Evaluation of the Integer Arithmetic Operations

Integer addition, subtraction, multiplication, and division are operated by complement addition and shift. One bit full-adder uses XOR (\oplus) gate to get sum and uses AND (\wedge) gate to get carry. Below we will explain our notation. The integer arithmetic operations will take $\mathcal{A} = a_{n-1} \cdots a_0$, $\mathcal{B} = b_{n-1} \cdots b_0$ and $\mathcal{A}^* = a_{n-1}^* \cdots a_0^*$, $\mathcal{B}^* = b_{n-1}^* \cdots b_0^*$ as inputs. \mathcal{A} and \mathcal{B} are the complements. \mathcal{A}^* and \mathcal{B}^* are two's complement of \mathcal{A} and \mathcal{B} . However, in complement operation, \mathcal{A} is original code, \mathcal{A}^* is the complement of the \mathcal{A} . The $\mathfrak{A} = \langle a_{n-1}, \dots, a_0 \rangle$ represents ciphertext vector of \mathcal{A} . Let $a_i = \text{Enc}(a_i)$, $0 \leq i \leq n-1$. The $\mathfrak{A}^* = \langle a_{n-1}^*, \dots, a_0^* \rangle$ represents ciphertext vector of \mathcal{A}^* , $a_i^* = \text{Enc}(a_i^*)$, $0 \leq i \leq n-1$. The \mathfrak{B} represents the ciphertext vector of \mathcal{B} , $b_i = \text{Enc}(b_i)$, $0 \leq i \leq n-1$. The $\mathfrak{B}^* = \langle b_{n-1}^*, \dots, b_0^* \rangle$ represents ciphertext vector of \mathcal{B}^* , $b_i^* = \text{Enc}(b_i^*)$, $0 \leq i \leq n-1$. The \mathfrak{A} , \mathfrak{B} , \mathfrak{A}^* , and \mathfrak{B}^* are inputs of homomorphic evaluation of the integer arithmetic operations. The n is big enough. We do not consider the overflow about integer arithmetic operations.

3.1. Homomorphic Evaluation of the Complement Operations. Fixed-point number use the complement to finish arithmetic operations. The rules of converting original code into complement:

- (i) Positive number: a positive complement and the same original code.
- (ii) Negative number: negative complement is the symbol for the numerical bit reverse and then at the bottom (LSB) plus 1.

Given the original code \mathcal{A} , apply the XOR and AND to get complement \mathcal{A}^* . The MSB a_{n-1} is the signed bit of \mathcal{A} , and the size of the remaining bits $a_{n-2} \cdots a_0$ represent the value.

Given an initialised carry $c_{-1} = 0$, we output the \mathcal{A}^* . The BP of complement:

$$\begin{aligned} c_{-1} &= 0 \\ c_i &= a_i \vee c_{i-1} \\ a_i^* &= (a_i \oplus a_n c_{i-1}) \end{aligned} \quad (2)$$

$$0 \leq i \leq n-2$$

We convert c_i into a BP by using XOR gate and AND gate. The BP: $c_i = a_i c_{i-1} \oplus a_i \oplus c_{i-1}$, $0 \leq i \leq n-2$. Due to initialised carry $c_{-1} = 0$, we can convert c_i into a polynomial without c_{-1} :

$$c_i = \sum_{k=1}^{i+1} \sum_{|\mathcal{S}|=i+1} \prod_{j \in \mathcal{S}} a_j \bmod 2 \quad (3)$$

where $\mathcal{S} = \{a_i, \dots, a_0\}$, $0 \leq i \leq n-3$, $|\mathcal{S}|$ is the Hamming weight of the \mathcal{S} . We can convert above polynomials into HP of complement by using addition and multiplication on ciphertexts. The HP of complement:

$$\begin{aligned} c_i &= \left(\sum_{k=1}^{i+1} \sum_{|\mathcal{S}|=i+1} \prod_{j \in \mathcal{S}} a_j \right) \bmod x_0 \\ a_i^* &= (a_i + a_n^* \cdot c_{i-1}) \bmod x_0 \end{aligned} \quad (4)$$

$$0 \leq i \leq n-2$$

where c_i represents the ciphertext result of c_i , and $Dec(c_i) = c_i$. Let $a_{n-1}^* = a_{n-1}$. The x_0 is the largest odd public key in FHE. We can get ciphertext complement $\mathbf{21}^*$ by using above HP.

3.2. Homomorphic Evaluation of the Addition and Subtraction Operations. Integer complement addition operation needs to calculate results in order from low to high. Every result bit requires an addend bit, an augend bit, and a carry bit from the low. Every carry bit requires an addend bit, an augend bit, and a carry bit from low as well. By iterating above operations, we can get the result of complement addition. We set integer complement \mathcal{A} and \mathcal{B} as inputs to calculate $\mathcal{S} = \mathcal{A} + \mathcal{B}$, where $\mathcal{S} = s_{n-1} \dots s_0$. The n is big enough. We do not consider the overflow of the \mathcal{S} . The BP of addition:

$$\begin{aligned} c_{-1} &= 0 \\ s_i &= a_i \oplus b_i \oplus c_{i-1} \\ c_i &= a_i b_i \oplus c_{i-1} (a_i \oplus b_i) \end{aligned} \quad (5)$$

$$0 \leq i \leq n-1$$

We can convert above BP into HP of addition by using addition and multiplication on ciphertexts. The HP of addition:

$$\begin{aligned} c_{-1} &= Enc(0) \\ s_i &= (a_i + b_i + c_{i-1}) \bmod x_0 \\ c_i &= (a_i b_i + c_{i-1} a_i + c_{i-1} b_i) \bmod x_0 \end{aligned} \quad (6)$$

$$0 \leq i \leq n-1$$

where the s_i is i -th ciphertext of result vector $\mathcal{S} = \langle s_{n-1}, \dots, s_0 \rangle$, and $Dec(\mathbf{21}) + Dec(\mathbf{23}) = Dec(\mathcal{S})$.

The addition operation can do integer subtraction operation. If we calculate $\mathcal{A} - \mathcal{B}$, we can convert \mathcal{B} to \mathcal{B}^* , and calculate $\mathcal{A} + \mathcal{B}^*$ by using integer addition operation. In order to get \mathcal{B}^* , we need to use the complement operation to get $\mathcal{B}' = b'_{n-1} \dots b'_0$, and then let $b'_{n-1} = b'_{n-1} \oplus 1$, $b'_i = b'_i$, $0 \leq i \leq n-2$. The HP of subtraction contains two parts, including HP of complement and HP of addition.

3.3. Homomorphic Evaluation of the Multiplication Operations. Integer multiplication operation is based on Booth's multiplication algorithm [36]. It is a multiplication algorithm that multiplies two signed numbers in two's complement notation. We set multiplicand \mathcal{A} , and multiplier \mathcal{B} . Booth's algorithm examines adjacent pairs of bits of the multiplier \mathcal{B} in signed two's complement representation, including an implicit bit below the least significant bit, $b_{-1} = 0$. Also, we denote by \mathcal{P} the product accumulator. The steps of the basic algorithm for multiplication operations:

- (1) We reinitialise the value of \mathcal{A} , \mathcal{A}^* , and \mathcal{P} .
 - (i) \mathcal{A} : $\mathcal{A} = \mathcal{A} \ll n$, arithmetic left shift $(n+1)$ bits. $\mathcal{A} = a_n a_{n-1} \dots a_0 0 \dots 0$.
 - (ii) \mathcal{A}^* : $\mathcal{A}^* = \mathcal{A}^* \ll n$, arithmetic left shift $(n+1)$ bits. $\mathcal{A}^* = a_n^* a_{n-1}^* \dots a_0^* 0 \dots 0$.
 - (iii) \mathcal{P} : fill the most significant n bits with 0. To the right of this, append the value of \mathcal{B} . Fill the LSB with a 0. $\mathcal{P} = 0 \dots 0 b_{n-1} \dots b_0 0$.
- (2) Determine the two least significant (rightmost) bits of \mathcal{P} .
 - (i) If $b_{-1} = b_0$, do nothing. Use \mathcal{P} directly in the next step. Arithmetic right shift 1 bit.
 - (ii) If $b_{-1} b_0 = 01$, find the value of $\mathcal{P} = \mathcal{P} + \mathcal{A}$. Ignore any overflow. Arithmetic right shift 1 bit.
 - (iii) If $b_{-1} b_0 = 10$, find the value of $\mathcal{P} = \mathcal{P} + \mathcal{A}^*$. Ignore any overflow. Arithmetic right shift 1 bit.

Repeat above second steps until they have been done $n-1$ times. Drop the LSB from \mathcal{P} . According to second steps mentioned technique, we can summarise a judgment choice Boolean polynomial (JCBP):

$$\begin{aligned} JCBP_{mult}(\mathcal{B}_0, \mathcal{B}_{-1}, \mathcal{A}^*, \mathcal{A}) \\ = (\mathcal{B}_0 \oplus \mathcal{B}_{-1}) [\mathcal{B}_0 \mathcal{A}^* + \mathcal{B}_{-1} \mathcal{A}] \end{aligned} \quad (7)$$

We use \mathcal{P}_i to represent i -th times iteration. The BP of multiplication operation:

$$\begin{aligned} \mathcal{P}_i &= \mathcal{P}_{i-1} + JCBP_{mult}(\mathcal{B}_0, \mathcal{B}_{-1}, \mathcal{A}^*, \mathcal{A}) \\ \mathcal{P}_i &= \mathcal{P}_{i-1} \gg 1 \end{aligned} \quad (8)$$

$$0 \leq i < n-1$$

where \gg represent the arithmetic right shift. We can convert formula (7) into HP of addition by using addition and multiplication on ciphertexts. The HP of $\text{JCBP}_{\text{mult}}(\mathcal{C}_0, \mathcal{C}_{-1}, \mathcal{A}^*, \mathcal{A})$:

$$\begin{aligned} & \text{JCHP}_{\text{mult}, x_0, \rho'}(\mathbf{b}_0, \mathbf{b}_{-1}, \mathbf{A}^*, \mathbf{A}, \mathbf{r}, \text{switch}(\oplus, \wedge)) \\ &= [(\mathbf{b}_0 + \mathbf{b}_{-1})(\mathbf{b}_0 \mathbf{A}^* + \mathbf{b}_{-1} \mathbf{A}) + 2\mathbf{r}] \bmod x_0 \end{aligned} \quad (9)$$

The \mathbf{A} and \mathbf{A}^* represent ciphertext vector of reinitialising \mathcal{A} and \mathcal{A}^* . The $\mathbf{r} = \langle r_0, \dots, r_{n-1} \rangle$ is a noise vector, and $r_i \leftarrow \mathbb{Z} \cap (-2^{\rho'}, 2^{\rho'})$, $0 \leq i \leq n-1$. The HP of multiplication:

$$\begin{aligned} \mathbf{P}_i &= \mathbf{P}_{i-1} \\ &+ \text{JCHP}_{\text{mult}, x_0, \rho'}(\mathbf{b}_0, \mathbf{b}_{-1}, \mathbf{A}^*, \mathbf{A}, \mathbf{r}, \text{switch}(\oplus, \wedge)) \end{aligned} \quad (10)$$

$$\mathbf{P}_i = \mathbf{P}_{i-1} \sim \gg 1$$

$$0 \leq i < n-1$$

where $\sim \gg$ represent the right shift of ciphertext vector \mathbf{P}_{n-1} . When the right shift of \mathbf{P}_{n-1} by 1 ciphertext slot, the most significant component of \mathbf{P}_{n-1} is filled with a copy of the original most significant component. The final value of \mathbf{P}_{n-1} is the signed ciphertext product.

3.4. Homomorphic Evaluation of the Division Operation. Division is the most complex of the basic arithmetic operations. For a simple computer that operate with an adder circuit for its arithmetic operations, a variant using traditional long division, called nonrestoring division, provides a simpler and faster speed. This method only needs one decision and addition/subtraction per quotient bit, and need not restoring step after the subtraction. We set dividend \mathcal{A} and divisor \mathcal{B} . The \mathcal{B}^* is two's complement of \mathcal{B} . The \mathcal{R} is the partial remainder, and the \mathcal{Q} is quotient. The basic algorithm for binary (radix 2) nonrestoring division is as follows:

(1) Reinitialize value of \mathcal{B} , \mathcal{B}^* , \mathcal{R} , and \mathcal{Q} .

- (i) \mathcal{B} : $\mathcal{B} = \mathcal{B} \gg n$, do arithmetic left shift n bits. $\mathcal{B} = \mathcal{C}_{n-1} \dots \mathcal{C}_0 0 \dots 0$.
- (ii) \mathcal{B}^* : $\mathcal{B}^* = \mathcal{B}^* \gg n$, do arithmetic left shift n bits. $\mathcal{B}^* = \mathcal{C}_{n-1}^* \dots \mathcal{C}_0^* 0 \dots 0$.
- (iii) \mathcal{R} : $\mathcal{R} = \mathcal{A} \gg n$, do arithmetic right shift n bits. $\mathcal{R} = \mathbf{r}_{2n-1} \dots \mathbf{r}_0$.
- (iv) \mathcal{Q} : $\mathcal{Q} = \mathbf{q}_{n-1} \dots \mathbf{q}_0$, fill it n bits with 0.

(2) Determine the one most significant (a signed bit) bit of \mathcal{R} .

- (i) If $\mathbf{r}_{2n-1} = 0$, fill the LSB of \mathcal{Q} with 1 digit, do logical left shift 1 bit. Find the value of $\mathcal{R} = 2 * \mathcal{R} + \mathcal{B}^*$.
- (ii) If $\mathbf{r}_{2n-1} = 1$, fill the LSB of \mathcal{Q} with 0 digit, do logical left shift 1 bit. Find the value of $\mathcal{R} = 2 * \mathcal{R} + \mathcal{B}$.

(3) Repeat above second steps until they have been done $n-1$ times.

(4) Convert the quotient \mathcal{Q} . We suppose original $\mathcal{Q} = 11101010$.

(i) Start: $\mathcal{Q} = 11101010$.

(ii) Mask the zero term (Signed binary notation with one's complement): $\overline{\mathcal{Q}} = 00010101$.

(iii) Subtract $\mathcal{Q} = \mathcal{Q} - \overline{\mathcal{Q}}$: $\mathcal{Q} = 11010101$.

(5) The actual remainder is $\mathcal{R} = \mathcal{R} \gg n$. Final result of quotient is always odd, and the remainder \mathcal{R} is in the range $-\mathcal{B} < \mathcal{R} < \mathcal{B}$. To convert to a positive remainder, do a single restoring step after \mathcal{Q} is converted from a nonstandard form to standard form. If $\mathcal{R} < 0$, find the value of $\mathcal{Q} = \mathcal{Q} - 1$ and $\mathcal{R} = \mathcal{R} + \mathcal{B}$.

According to the second step mentioned technique, we can summarise a judgment choice Boolean polynomial (JCBP):

$$\text{JCBP}_{\text{div}}(\mathbf{r}_{2n-1}, \mathcal{B}, \mathcal{B}^*) = \mathbf{r}_{2n-1} \mathcal{B} + (\mathbf{r}_{2n-1} \oplus 1) \mathcal{B}^* \quad (11)$$

We use \mathcal{R}_i to represent i -th times iteration. The BP of division operation:

$$\mathcal{R}_i = 2\mathcal{R}_{i-1} + \text{JCBP}_{\text{div}}(\mathbf{r}_{2n-1}, \mathcal{B}, \mathcal{B}^*)$$

$$0 \leq i < n-1$$

$$\mathcal{Q}_{n-1-i} = \mathbf{r}_{i,2n-1} \oplus 1 \quad 0 \leq i < n-1 \quad (12)$$

$$\mathcal{Q} = \mathcal{Q} - \overline{\mathcal{Q}}$$

$$\mathcal{R}_{n-1} = \mathcal{R}_{n-1} \gg n$$

where $\mathbf{r}_{i,2n-1}$ represent the MSB of \mathcal{R}_i . Finally, doing the fifth step corrects \mathcal{Q} and \mathcal{R}_{n-1} . We can convert above BP into HP of addition by using addition and multiplication on ciphertexts. The HP of $\text{JCBP}_{\text{div}}(\mathbf{r}_{2n-1}, \mathcal{B}, \mathcal{B}^*)$:

$$\begin{aligned} & \text{JCBP}_{\text{div}, x_0, \rho'}(\mathbf{r}_{2n-1}, \mathbf{B}^*, \mathbf{B}, \mathbf{r}, \text{switch}(\oplus, \wedge)) \\ &= [\mathbf{r}_{2n-1} \mathbf{B} + (\mathbf{r}_{2n-1} + \text{Enc}(1)) \mathbf{B} + 2\mathbf{r}] \bmod x_0 \end{aligned} \quad (13)$$

where \mathbf{B} and \mathbf{B}^* represent ciphertext vector of reinitialising \mathcal{B} and \mathcal{B}^* , respectively. The HP of division:

$$\mathbf{R}_i$$

$$= 2\mathbf{R}_{i-1}$$

$$+ \text{JCBP}_{\text{div}, x_0, \rho'}(\mathbf{r}_{2n-1}, \mathbf{B}^*, \mathbf{B}, \mathbf{r}, \text{switch}(\oplus, \wedge))$$

$$0 \leq i < n-1 \quad (14)$$

$$\mathbf{q}_{n-1-i} = (\mathbf{r}_{i,2n-1} + \text{Enc}(1)) \bmod x_0 \quad 0 \leq i < n-1$$

$$\mathbf{Q} = \mathbf{Q} - \overline{\mathbf{Q}}$$

$$\mathbf{R}_{n-1} = \mathbf{R}_{n-1} \sim \gg n$$

where $\mathbf{R}_i = \langle \mathbf{r}_{i,2n-1}, \dots, \mathbf{r}_{i,0} \rangle$ represents ciphertext vector of \mathcal{R}_i and $\mathbf{Q} = \langle \mathbf{q}_{n-1}, \dots, \mathbf{q}_0 \rangle$ represents ciphertext vector of \mathcal{Q} . $\overline{\mathbf{Q}}$

represent $q_i = (q_i + \text{Enc}(1)) \bmod x_0$, $0 \leq i < n$. Finally, we need to correct \mathfrak{Q} and \mathfrak{R}_{n-1} by the following operations:

$$\begin{aligned}\mathfrak{Q} &= [\mathbf{r}_{n-1,n-1} (\mathfrak{Q} - \text{Enc}(1))] \bmod x_0 \\ \mathfrak{R} &= [\mathbf{r}_{n-1,n-1} (\mathfrak{R}_{n-1} + \mathfrak{B})] \bmod x_0\end{aligned}\quad (15)$$

The final value of \mathfrak{Q} and \mathfrak{R} is the result of HP of division.

4. Noise Analysis and Optimization

In Section 3, we describe the homomorphic evaluation of the integer arithmetic operations including complement, addition, subtraction, multiplication, and division. In this section, we will analyse the noise ceiling and optimisation for our scheme. Under the DGHV scheme, the noise ceiling increases exponentially with the multiplicative degree. When ciphertexts have noise at most $2^{\eta-2} < p/2$, the ciphertexts cannot be decrypted correctly. Therefore, we need bootstrapping to control noise of ciphertexts, but using bootstrapping to refresh ciphertext will reduce the efficiency of DGHV. We will show the noise ceiling about the homomorphic evaluation of the integer arithmetic operations in this section. Moreover, we will describe an optimisation for the process of integer arithmetic operations to reduce time overhead in FHE with bootstrapping.

4.1. Noise Analysis of Our Scheme. According to Section 3, we show the noise ceiling and items of homomorphic evaluation of the n bits signed integer arithmetic operations. We denote by HE-IAO the homomorphic evaluation of the integer arithmetic operations and use HE-com, HE-add, HE-sub, HE-mul, HE-div to represent five operations of HE-IAO. The details are shown in Table 2.

Proof. According to homomorphic evaluation of the complement operations formula (3),

$$c_i = \sum_{k=1}^{i+1} \sum_{|\mathcal{S}|=i+1} \prod_{j \in \mathcal{S}} a_j \bmod 2, \quad 0 \leq i \leq n-2 \quad (16)$$

$\sum_{|\mathcal{S}|=i} \prod_{j \in \mathcal{S}} a_j$ has $\binom{n-2}{i+1}$ terms, and degree amounts to $i+1$. According to binomial theorem, the term of formula (3) can be represented as $(1+x)^{n-2} - 1 = \sum_{i=0}^{n-2} \binom{n-2}{i} x^i - 1$, when $x = 1$. Therefore, the polynomial c_i of term is up to $2^{n-2} - 1$, and degree amounts to $n-2$. Because of $a_i^* = (a_i \oplus a_n c_{i-1})$, $0 \leq i \leq n-1$, a_i^* of term amounts to 2^{n-2} , and degree amounts to $n-1$.

According to homomorphic evaluation of the addition operations formula (5), the degree of carry formula $c_i = a_i \oplus b_i \oplus c_{i-1} (a_i \oplus b_i)$ is higher 1 than c_{i-1} , and the term of c_i is higher $2 * \text{term}(c_{i-1}) + 1$ than c_{i-1} , where $\text{term}(c_{i-1})$ represents the term of c_{i-1} . The degree of $c_0 = a_0 \oplus b_0$ amounts to 2, and the term of $c_0 = a_0 \oplus b_0$ amounts to 1. The degree of c_{n-2} amounts to n , and terms of c_{n-2} amount to $2^{n-1} - 1$. The MSB of $\mathcal{S} \mathfrak{J}_{n-1} = a_{n-1} \oplus b_{n-1} \oplus c_{n-2}$ where the degree of \mathfrak{J}_{n-1} amounts to n , and the term of \mathfrak{J}_{n-1} amounts to $2^{n-1} + 1$. If we do not consider complement operation, subtraction and

TABLE 2: The degree ceiling and items of homomorphic evaluation of the n bits signed integer arithmetic operations. Above showing the HP of integer arithmetic operations n -th iterations ciphertext results' degree and term.

HE-IAO	degree	term
HE-com	$n-1$	2^{n-2}
HE-add	n	$2^{n-1} + 1$
HE-sub	n	$2^{n-1} + 1$
HE-mul	$\psi \cdot 2^{2n-4}$	—
HE-div	$\varphi \cdot 2^{2n-4}$	—

TABLE 3: The noise ceiling of homomorphic evaluation of the n bits signed integer arithmetic operations. The base of the log is 2.

HE-IAO	noise ceiling
HE-com	$\rho'^{\log(n-1)} + \log 2^{n-3}$
HE-add	$\rho'^{\log(n)} + \log(2^{n-1} + 1)$
HE-sub	$\rho'^{\log(n)} + \log(2^{n-1} + 1)$
HE-mul	$\rho'^{\log \psi \cdot 2^{2n-4}} + -$
HE-div	$\rho'^{\log \varphi \cdot 2^{2n-4}} + -$

addition have the same process. Therefore, the degree and term are the same as the addition. Multiplication and division require iteration $n-1$ times addition and shift. The processes of multiplication and division are particularly complicated, we can't find the formula to express the degree. According to our calculations, the degree of multiplication and division is close to the 2 to the power of $2n-4$. Therefore, the degree of multiplication is not more than $\psi \cdot 2^{2n-4}$, and the degree of division is not more than $\varphi \cdot 2^{2n-4}$. The term of multiplication and division is too high, and the degree has made noise more than the limitation of correct decryption. \square

We can conclude (see Table 2) the degree of homomorphic evaluation of addition and subtraction is $\bar{\mathcal{O}}(n)$, and the depth of the homomorphic evaluation of multiplication and division is $\bar{\mathcal{O}}(\psi \cdot 2^{2n-4})$. We use items that represent the l_1 norm of HP (the coefficient vector of HP). The noise ceiling of homomorphic evaluation of the n bits integer arithmetic operations can be calculated by the following polynomial:

$$\text{Noise} = \rho'^{\log d} + \log \left| \vec{f} \right| \quad (17)$$

where d represents the degree of HP and $\log d$ represents multiplication level of HP. $|\vec{f}|$ is the l_1 norm of HP. ρ' is the noise of length in every ciphertext. The noise ceiling of homomorphic evaluation of the n bits integer arithmetic operations is shown in Table 3.

We can conclude (see Table 3) the noise of homomorphic evaluation of the n bits signed integer arithmetic operations. The noise increases very rapidly. In the homomorphic evaluation of the complement, addition, and subtraction, noise increases up to $\bar{\mathcal{O}}(\rho' n)$. In the homomorphic evaluation of the multiplication and division, noise is more than $\bar{\mathcal{O}}(\rho' \psi \cdot 2^{2n-4})$.

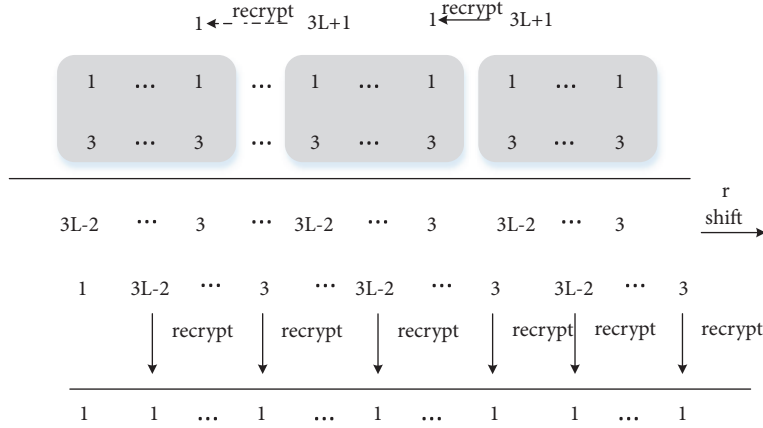


FIGURE 2: The optimised addition operations; each colour block represents same position subvectors of \mathbf{P}_i and \mathbf{S} , the length of each colour block is L , “ r shift” represents ciphertext vector right shift one position, “recrypt” represents ciphertext refresh. This figure shows the degree change of each component of \mathbf{P}_i and \mathbf{S} .

4.2. Optimization of Our Scheme. From the analysis of noise ceiling in Section 4.1, we need to control the noise increases by using ciphertext refresh, or modulus-switching technique in each multiplication level of homomorphic evaluation of the n bits signed integer arithmetic operations. However, the ciphertext refresh or modulus-switching technique will reduce our scheme efficiency. Therefore, we will describe an optimisation for our scheme in this section. The optimisation can reduce the number of ciphertext refresh and improve efficiency.

Because the integer arithmetic operation is completed based on the addition operation, we optimise the homomorphic evaluation of the integer addition. We divide the ciphertext vector of integer encryption into subvectors of length L . The homomorphic evaluation of arithmetic operations between the subvectors in the same position do not need ciphertext refresh, and the subvectors in different positions need to refresh ciphertext-carry once. Therefore, the optimised scheme need not ciphertext refresh in each multiplication level and reduces the number of ciphertext refresh. We take addition operations of the product accumulator to explain our optimisation. In the homomorphic evaluation of the multiplication operations, the product accumulator \mathbf{P}_i is as formula (9). The formula (9) is a polynomial of degree three, and \mathbf{P}_i is a ciphertext vector of degree one. We denote by \mathbf{S} vector of $\text{JCHP}_{mult, x_0, \rho'}$. We divide the ciphertext vector of \mathbf{P}_i and \mathbf{S} into ciphertext subvectors of length L . The ciphertext operations of subvectors of the same position are the same as homomorphic evaluation of the addition operations (without ciphertext refresh). The ciphertext operations of subvectors of different position need to refresh ciphertext-carry once (see Figure 2).

The subvectors of \mathbf{P}_i and \mathbf{S} generate ciphertext-result block and ciphertext-carry. In every ciphertext-result block, the maximum degree is $3L - 2$ (leftmost), and the minimum degree is 3 (rightmost). The ciphertext-carry degree is $3L + 1$ in each subvector of the different position. Each ciphertext of ciphertext-result block and ciphertext-carry can use

ciphertext refresh (recrypt) to reduce degree up to one. The number of ciphertext refresh of once product accumulator ($\mathbf{P}_i + \mathbf{S}$) for ciphertext-carry:

- (i) If n is divisible by L , it will generate $n/L - 1$ ciphertext-carry, and the number of ciphertext refresh is $cnt = n/L - 1$ times.
- (ii) If n is inalienable by L , it will generate $\lfloor n/L \rfloor$ ciphertext-carry, and the number of ciphertext refresh is $cnt = \lfloor n/L \rfloor$ times.

Homomorphic evaluation of the multiplication operations needs $n - 1$ times product accumulator, and whole operations need $(n - 1) \cdot cnt$ times ciphertext refresh for ciphertext-carry, and $(n - 1) \cdot n$ times ciphertext refresh for ciphertext-result block. Total times of ciphertext refresh for homomorphic evaluation of the multiplication operations:

$$CNT = (n - 1) \cdot n + (n - 1) \cdot cnt \quad (18)$$

In the original homomorphic evaluation of the multiplication operations, every multiplication level needs once ciphertext refresh. Whole operations need $(n - 1)(3n - 3) + n(n - 1) + 2n - 3$ times ciphertext refresh. We apply above optimisation to the homomorphic evaluation of the complement, addition, subtraction, multiplication, and division operations, and set $n = 16$, $L = 1, 2, 3, 4$. We show the number of ciphertext refresh of the original scheme ($L = 0$) and an optimised scheme ($L = 1, 2, 3, 4$) as Table 4.

In order to implement our optimisation, we need to adjust the parameters of homomorphic encryption and make FHE support the $\log(3L + 1)$ multiplication level additional, except for the 15-multiplication level required by bootstrapping. Therefore, we reset the length of the private key: $\eta \geq \rho' \cdot \Theta(\lambda \log^2 \lambda) + \rho' \cdot 2^{\log(3L+1)} + \log(2^{L-1} + 1)$. We set security parameter $\lambda = 52$, the length of noise $\rho' = 24$, and $\theta = 15$, $\Theta = 500$. Parameters are set as Table 5.

TABLE 4: The number of ciphertext refresh of homomorphic evaluation of the integer arithmetic operations.

HE-IAO	L = 0	L = 1	L = 2	L = 3	L = 4
HE-com	29	29	22	20	18
HE-add	45	31	23	21	19
HE-sub	74	60	45	41	37
HE-mul	944	494	367	335	303
HE-div	1095	585	437	397	359

TABLE 5: Concrete parameters, based on the “small” parameters of CMNT scheme, and reset the length of public key η and γ . Fixed λ , ρ' and the number of public keys τ , only changed η and γ according to L.

parameters	λ	ρ'	η	γ	τ
L = 0	52	24	1632	$2.0 \cdot 10^6$	1000
L = 1	52	24	1728	$2.5 \cdot 10^6$	1000
L = 2	52	24	1801	$3.0 \cdot 10^6$	1000
L = 3	52	24	1874	$3.5 \cdot 10^6$	1000
L = 4	52	24	1993	$4.2 \cdot 10^6$	1000

5. Experimental Result

Our optimisations described in our scheme were incorporated in our code, which is built on top of GnuMP. We tested our implementation on a desktop computer with Intel Core i5-3470 running at 3.2 GHz, on which we run an Ubuntu 18.04 with 8 GB of RAM and with the gcc compiler version 6.2. We regard this desktop computer as an edge data centre to test our scheme efficiency in edge computing. We choose the ciphertext vector of 16 and 8 bits signed integer as input. Due to our optimisations for the FHE with bootstrapping, we test our original scheme ($L = 0$) and optimised scheme ($L = 1, 2, 3, 4$) in DGHV and CMNT scheme (see Figures 3(a), 3(b), 3(c), 3(d), and 3(e)). In CNT scheme, we only use modulus-switching technique to control noise. So, we test the original scheme in CNT scheme (see Figures 4(a) and 4(b)). In our figures, “HE-com (CMNT 16)” represent the homomorphic evaluation of the complement operations and calculate the ciphertext vector of 16 bits signed integer in the CMNT scheme. “HE-com (DGHV 8)” represent the homomorphic evaluation of the complement operations and calculate the ciphertext vector of 8 bits signed integer in DGHV scheme. The explanation of other legends is the same as above. We show the following experimental results based on the Section 4.2 parameter settings.

We can conclude (see Figures 3(b), 3(c), 3(d), and 3(e)) that $L = 2$ is best parameters setting for homomorphic evaluation of the addition, subtraction, multiplication, and division. However, the time overhead of homomorphic evaluation of the complement operations is monotone increasing at L (see Figure 3(a)). When $L = 0$, the time overhead is the absolute minimum. When $L = 0$ and $L = 1$, the number of ciphertext refresh is the same. Also, the degree of complement operations is $2n - 3$. When $n = 16$, the degree is 29 in $L = 0$. It is the same as $L = 1$. The reduced time overhead of the optimised complement operations cannot offset the computation cost caused by the increase in ciphertext length. However, the relative minimum value appears at $L = 2$

(see Figures 3(a), 3(b), 3(c), 3(d), and 3(e)). It shows that our optimisation is useful and can reduce the time overhead for our scheme, except homomorphic evaluation of the complement operations. Namely, we divide the ciphertext vector of integer encryption into subvectors of length 2 and make DGHV and CMNT scheme to support the $\log(3 \cdot 2 + 1) \approx 3$ multiplication level additional. The optimisation of our scheme can achieve the best results. In the number of ciphertext refreshes, the optimised scheme is reduced by 2/3 compared to the original scheme over DGHV and CMNT, and the homomorphic evaluation time overhead of integer arithmetic operation is reduced by 1/3.

The DGHV and CMNT schemes are different with CNT scheme in noise management technology. The DGHV and CMNT schemes use bootstrapping to control noise, and the CNT scheme uses the modulus switching technology to control noise. Our optimization is used for homomorphic evaluation of integer arithmetic operations which are implemented by DGHV and CMNT schemes (with bootstrapping), rather than by CNT scheme (modulus-switching). Therefore, we just compare the best result ($L = 2$) of our scheme based on DGHV and CMNT schemes with CNT scheme ($L = 0$). We can draw a conclusion (see Figures 4(a) and 4(b)) that the time overhead of our optimised scheme based on DGHV and CMNT schemes (bootstrapping) with $L = 2$ close to the time overhead of basing on CNT (modulus-switching) with $L = 0$. It also shows that our optimisation is effective. And our optimized scheme can be applied to mobile edge computing to solve privacy data computing problems in ciphertext.

6. Conclusion

We implement the homomorphic evaluation scheme of integer arithmetic operations under DGHV and its variants in edge computing. We use the features of homomorphic encryption to prevent sensitive data from being stolen in edge computing. At the same time, we can also calculate ciphertext data in edge computing and improve the QoS

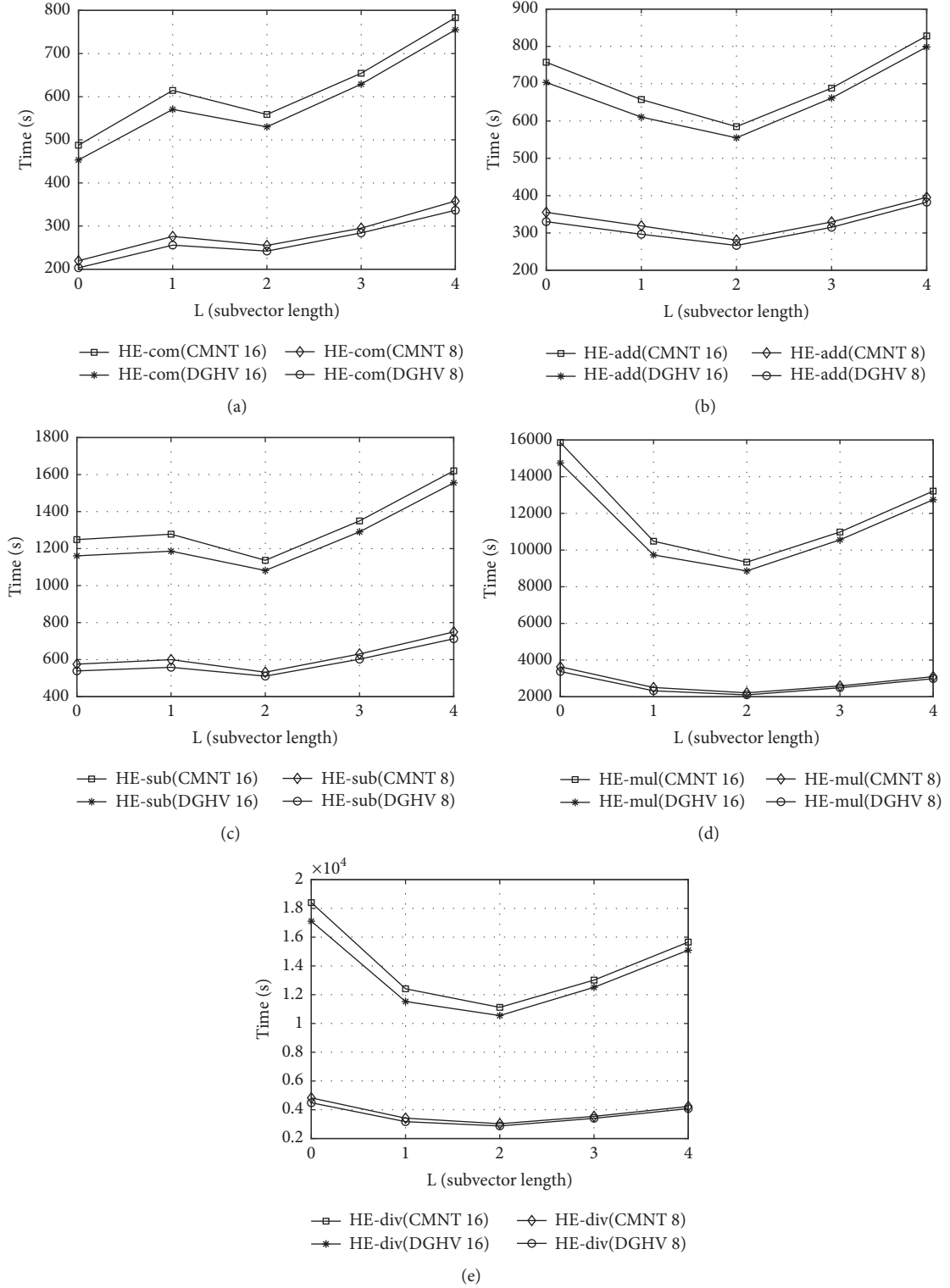


FIGURE 3: Tested (a) HE-com, (b) HE-add, (c) HE-sub, (d) HE-mul, (e) HE-div CPU time.

of edge computing. Through scheme design, noise analysis, scheme optimisation, and experimental comparison, the different performance of homomorphic evaluation of the integer arithmetic operations is obtained under different FHE schemes. Although we optimise our scheme in Section 4.2,

the time overhead of our scheme is still high. The primary open problem is to improve the efficiency of the FHE scheme. This requires us to work together to find a natural FHE scheme or to continue to optimise the FHE architecture to achieve more efficient noise management techniques.

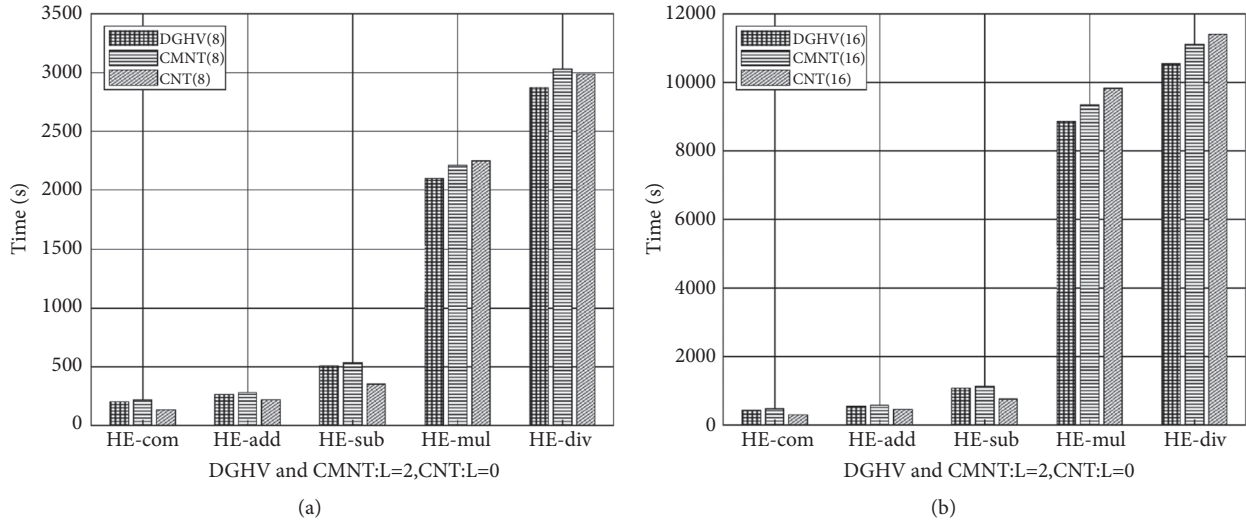


FIGURE 4: (a) Tested HE-IAO CPU time of ciphertext vector of length 8. (b) Tested HE-IAO CPU time of ciphertext vector of length 16.

Data Availability

The data and code used to support the findings of this study have been deposited in the GitHub repository (<https://github.com/limengfeil187/Homomorphic-Encryption.git>).

Conflicts of Interest

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

Acknowledgments

This work is partly supported by the National Science Foundation of China (61701322), the Key Projects of Liaoning Natural Science Foundation (20170540700), and the Liaoning Provincial Department of Education Science Foundation (L201630).

References

- [1] L. He, O. Kaoru, and D. Mianxiong, "ECCN: Orchestration of Edge-Centric Computing and Content-Centric Networking in the 5G Radio Access Network," *IEEE Wireless Commun*, vol. 25, no. 3, pp. 88–93, 2018.
- [2] M. Tao, K. Ota, and M. Dong, "Foud: Integrating Fog and Cloud for 5G-Enabled V2G Networks," *IEEE Network*, vol. 31, no. 2, pp. 8–13, 2017.
- [3] J. Xu, K. Ota, and M. Dong, "Real-Time Awareness Scheduling for Multimedia Big Data Oriented In-Memory Computing," *IEEE Internet of Things Journal*, 2018.
- [4] P. G. Lopez, A. Montresor, D. Epema et al., "Edge-centric computing: vision and challenges," *Computer Communication Review*, vol. 45, no. 5, pp. 37–42, 2015.
- [5] Z. Zhou, M. Dong, K. Ota, J. Wu, and T. Sato, "Energy efficiency and spectral efficiency tradeoff in device-to-device (D2D) communications," *IEEE Wireless Communications Letters*, vol. 3, no. 5, pp. 485–488, 2014.
- [6] A. Y. Al-Dubai, L. Zhao, A. Y. Zomaya, and G. Min, "QoS-Aware Inter-Domain Multicast for Scalable Wireless Community Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 11, pp. 3136–3148, 2015.
- [7] L. Zhao, A. Al-Dubai, X. Li, and G. Chen, "A new efficient cross-layer relay node selection model for Wireless Community Mesh Networks," *Computers Electrical Engineering*, vol. 61, pp. 361–372, 2017.
- [8] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. K. Karagiannis, "A Survey on Mobile Anchor Node Assisted Localization in Wireless Sensor Networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2220–2243, 2016.
- [9] L. Zhao et al., "Vehicular Communications: Standardization and Open Issues," *IEEE Communications Standards Magazine*, 2019.
- [10] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.
- [11] X. Tao, K. Ota, M. Dong, H. Qi, and K. Li, "Performance guaranteed computation offloading for mobile-edge cloud computing," *IEEE Wireless Communications Letters*, vol. 6, no. 6, pp. 774–777, 2017.
- [12] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st annual ACM symposium on Theory of Computing (STOC '09)*, pp. 169–178, ACM, Bethesda, Md, USA, 2009.
- [13] S. Halevi, "Homomorphic Encryption," in *Tutorials on the Foundations of Cryptography*, Information Security and Cryptography, pp. 219–276, Springer International Publishing, Cham, 2017.
- [14] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Advances in cryptology—EUROCRYPT 2010*, vol. 6110, pp. 24–43, Springer, Berlin, Germany, 2010.
- [15] C. Gentry and S. Halevi, "Implementing Gentry's fully—homomorphic encryption scheme," in *Advances in cryptology—EUROCRYPT 2011*, vol. 6632 of *Lecture Notes in Computer Science*, pp. 129–148, Springer, Heidelberg, Germany, 2011.
- [16] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," *Lecture Notes in*

- Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6056, pp. 420–443, 2010.
- [17] J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, “Fully homomorphic encryption over the integers with shorter public keys,” in *Proceedings of the 31st Annual International Cryptology Conference (CRYPTO ’11)*, vol. 6841 of *Lecture Notes in Computer Science*, pp. 487–504, Springer, Santa Barbara, Calif, USA.
 - [18] Z. Brakerski and V. Vaikuntanathan, “Efficient fully homomorphic encryption from (standard) LWE,” in *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS ’11)*, pp. 97–106, Palm Springs, Calif, USA, October 2011.
 - [19] Z. Brakerski and V. Vaikuntanathan, “Fully homomorphic encryption from ring-LWE and security for key dependent messages,” in *Advances in Cryptology—CRYPTO 2011*, R. Phillip, Ed., vol. 6841, pp. 505–524, Springer, Berlin, Germany, 2011.
 - [20] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “(Leveled) fully homomorphic encryption without bootstrapping,” in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pp. 309–325, ACM, 2012.
 - [21] A. López-Alt, E. Tromer, and V. Vaikuntanathan, “On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption,” in *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC ’12)*, pp. 1219–1234, ACM, May 2012.
 - [22] J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig, “Improved security for a ring-based fully homomorphic encryption scheme,” in *Cryptography and coding*, vol. 8308 of *Lecture Notes in Comput. Sci.*, pp. 45–64, Springer, Heidelberg, 2013.
 - [23] C. Gentry, A. Sahai, and B. Waters, “Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based,” *Proceedings of CRYPTO 2013*, vol. 8042, no. 1, pp. 75–92, 2013.
 - [24] C. Gentry, S. Halevi, and N. P. Smart, “Homomorphic evaluation of the AES circuit,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7417, pp. 850–867, 2012.
 - [25] C. Gentry, S. Halevi, and N. P. Smart, “Fully homomorphic encryption with polylog overhead,” in *Advances in cryptology—EUROCRYPT 2012*, vol. 7237 of *Lecture Notes in Comput. Sci.*, pp. 465–482, Springer, Heidelberg, 2012.
 - [26] N. P. Smart and F. Vercauteren, “Fully homomorphic SIMD operations,” *Designs, Codes and Cryptography*, vol. 71, no. 1, pp. 57–81, 2014.
 - [27] Y. Chen and G. Gong, “Integer arithmetic over ciphertext and homomorphic data aggregation,” in *Proceedings of the 3rd IEEE International Conference on Communications and Network Security, CNS 2015*, pp. 628–632, Italy, September 2015.
 - [28] K. Gai and M. Qiu, “Blend Arithmetic Operations on Tensor-Based Fully Homomorphic Encryption Over Real Numbers,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3590–3598, 2018.
 - [29] K. Gai, M. Qiu, Y. Li, and X.-Y. Liu, “Advanced Fully Homomorphic Encryption Scheme over Real Numbers,” in *Proceedings of the 4th IEEE International Conference on Cyber Security and Cloud Computing, CSCloud 2017 and 3rd IEEE International Conference of Scalable and Smart Cloud, SSC 2017*, pp. 64–69, USA, June 2017.
 - [30] L. Kuang, L. T. Yang, J. Feng, and M. Dong, “Secure Tensor Decomposition Using Fully Homomorphic Encryption Scheme,” *IEEE Transactions on Cloud Computing*, vol. 6, no. 3, pp. 868–878, 2018.
 - [31] J.-S. Coron, D. Naccache, and M. Tibouchi, “Public key compression and modulus switching for fully homomorphic encryption over the integers,” in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2012)*, vol. 7237 of *Lecture Notes in Comput. Sci.*, pp. 446–464, Springer.
 - [32] J. H. Cheon, J. S. Coron, J. Kim et al., “Batch fully homomorphic encryption over the integers,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 7881 of *Lecture Notes in Computer Science*, pp. 315–335, Springer, Berlin, Germany, 2013.
 - [33] J.-S. Coron, T. Lepoint, M. Tibouchi, J. H. Cheon, and J. Kim, “Batch fully homomorphic encryption over the integers,” in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 315–335, Springer, Berlin, Germany, 2013.
 - [34] J. Kim, M. S. Lee, A. Yun et al., CRT-based fully homomorphic encryption over the integers, *Cryptology ePrint Archive*, <https://eprint.iacr.org/057.pdf>.
 - [35] S. Halevi and V. Shoup, “Algorithms in helib,” in *Lecture Notes in Computer Science*, vol. 8616 of *Lecture Notes in Comput. Sci.*, pp. 554–571, Springer, Heidelberg, 2014.
 - [36] A. D. Booth, “A signed binary multiplication technique,” *The Quarterly Journal of Mechanics and Applied Mathematics*, vol. 4, pp. 236–240, 1951.