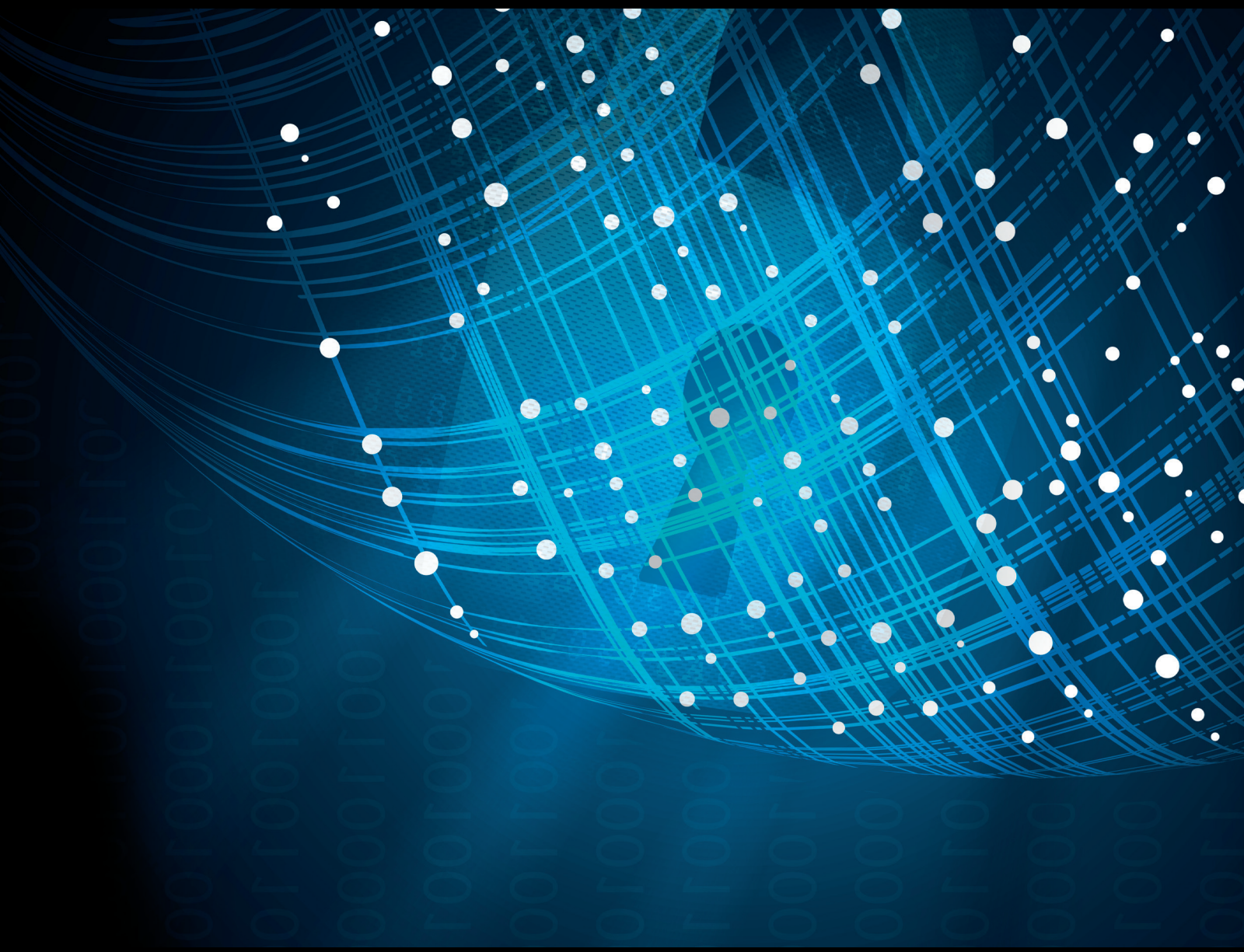
Security and Communication Networks

Secure Deployment of Commercial Services in Mobile Edge Computing 2021

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



Secure Deployment of Commercial Services in Mobile Edge Computing 2021

Security and Communication Networks

Secure Deployment of Commercial Services in Mobile Edge Computing 2021

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

Contents

Contents

Hindawi
Security and Communication Networks
Volume 2022, Article ID 3664986, 11 pages
https://doi.org/10.1155/2022/3664986

Research Article

Secure Analysis for IIOT Systems Using Hyperchaotic Image Encryption

Haini Zeng and Qiping Zou

School of Artificial Intelligence and Smart Manufacturing, Hechi Univesity, Yizhou 546300, China

Correspondence should be addressed to Qiping Zou; 706611232@qq.com

Received 8 October 2021; Revised 8 April 2022; Accepted 29 April 2022; Published 24 June 2022

Academic Editor: Xiaolong Xu

Like edge computing, intelligent cameras and image sensors are widely used in the Industrial Internet of Things (IIOT), including design and finished product quality inspection. However, the images generated by these sensors are constantly at risk of information leakage and privacy violations in the IIOT. Due to the involvement of third parties, traditional encryption algorithms are no longer adapted to image encryption for IIOT. In the context of the IIOT, an image encryption technology based on hyperchaotic systems and dynamic DNA coding is proposed. First, the image pixel position is scrambled by the hyperchaotic mapping index sequence, so that the image pixel matrix is dynamically DNA coded, and the base operation is performed on the given DNA sequence. Then, Keccak is used to calculate the hash value of the given DNA sequence as the initial value of the chaotic system and a certain number of base substitutions are performed on the DNA encoded pixel value according to the quaternary hyperchaotic sequence generated by the hyperchaotic system. Finally, ciphertext feedback and chaotic system iteration are used to further enhance the confusion and diffusion characteristics of the algorithm. The test results show that the algorithm not only has a large key space, strong sensitivity to keys, but also has strong resistance to exhaustive analysis attacks.

1. Introduction

As one of the most promising industries in the world today, the Internet of Things (IOT) is strongly driving the digital transformation and upgrading of traditional industries, enabling human society to enter the era of the Internet of everything [1–6]. The IOT collects, perceives, and analyzes the corresponding data from the surrounding environment through connected nodes and makes specific responses. Edge computing is widely used in IIOT to make up for the deficiencies of cloud computing [7–9]. Wireless multimedia sensor networks (WMSNs) is one of the IOT auxiliary devices, which consists of vision sensors. The WMSNs supervises the surrounding environment by continuously capturing images of the surrounding environment by visual sensors. However, the large amount of visual data obtained has significant redundancy [10]. Researchers of surveillance networks generally agree that multimedia surveillance networks should have visual data collection and record sensitive data for future use, such as anomalous event detection, case management, data analysis, and video abstraction. Due to energy and bandwidth constraints, it is impractical to send unprocessed video data over communication lines. In addition, extracting sensitive data from the large amount of surveillance data is difficult and time-consuming [11]. Therefore, it is necessary to exploit the processing and transmission capabilities of smart vision sensors to autonomously collect important visual data. This facilitates the intelligent selection of the appropriate picture from the multiview surveillance data captured by the connected IOT infrastructure of multiple sensors. It can process the collected data in real time, in order to send relevant data to a central memory. In addition, it enables surveillance experts to grasp the relevance of the original lengthy sequence by simply analyzing representative frames. However, when visualization data from WMSNs are sent wirelessly to a vision processing center (VPH) or a base station (BS), the communication is vulnerable to several security issues.

Therefore, some security mechanisms need to be designed to transmit the visual data safely to the BS because any slight change in the transmitted data can affect the decision of visual data analyst at the base station. In addition, it is comparatively difficult to transmit multimedia data in WMSNs using dedicated lines due to congestion in the bandwidth allocation mechanism.

Chaotic mappings have been widely used in the design of encryption schemes due to their unpredictability, ergodicity, and sensitivity to parameters and initial values [12]. It is generally more common to use chaotic mappings for the generation of pseudorandom number sequences, but studies have shown that the dynamical properties of chaotic mappings degrade to some extent. When calculating chaotic mapping values, the limitation of accuracy makes the chaotic mapping exist with finite and periodic orbits [13], which leads to degradation of all properties of the chaotic system and makes the chaos-based encryption scheme security flawed. Therefore, it is necessary to investigate the dynamical properties of chaotic mappings with finite accuracy. The structure and properties of several chaotic mappings have been studied by some scholars. In 2016, Yoshioka and Kawano analyzed in detail the relationship between the period, initial value, and order of Chebyshev polynomials on the ring of integer power remainders of 2 [14]. In 2019, Li et al. derived a strong correlation between the nodes corresponding to one-dimensional chaotic mappings on the domain of fixed-point operators and proved the number of iterations required for the iteration value of the tent mapping to converge to zero [15].

Due to the large amount of image data and high redundancy, in order to better meet the requirements of image protection, some researchers combine DNA encoding rules and chaotic mapping theory to propose some new image encryption methods [16–19]. For example, Chai et al. [17, 18] established an image encryption algorithm based on chaos by making appropriate improvements. Encryption is realized by combining with DNA code, and it turns out that its encryption performance reaches a higher level and the efficiency is high. Only DNA coding rules are used in the algorithm, which is simple to operate and shows strong applicability. However, in practical applications, most of the DNA encoding rules selected are fixed, which makes the algorithm's ability to resist exhaustive attacks very weak, so this is likely to cause security risks.

Therefore, this paper addresses these issues by employing an intelligent and efficient system that intelligently collects important data and gives appropriate decisions in real time through each sensor node, thus reducing bandwidth consumption and transmission costs. In addition, this paper proposes a security algorithm for secure transmission of sensitive visual data to the fusion center. We combine the hyperchaotic system, DNA calculation, and Keccak function to encrypt the image in chunks. The hash function Keccak processes is used to process the original image to obtain the initial value of the hyperchaotic system. Technically, the system encrypts the visual data using image encryption before transmitting the data, thus improving the security during communication in industrial WMSNs.

The approach and results show that the proposed encryption algorithm can encrypt different images securely and efficiently, making it more suitable for Industrial Internet of Things.

The rest of the paper is organized as follows: Section 2 shows the proposed system in detail. Section 3 shows encryption algorithm design. Section 4 presents the experimental results, and then the study concludes in Section 5.

2. Hyperchaotic System and Dynamic DNA Coding Algorithm

In the process of obfuscation and diffusion operation, the encryption algorithm in this article uses two kinds of chaotic sequences, DNA sequence library and pixel gray value conversion operation to achieve the purpose of encryption.

2.1. Improved Hyperchaotic Systems. Chaos is a complex phenomenon in nature. In 1963, Lorenz [20–25] used computer numerical experiments to discover the first chaotic attractor. This discovery is an important milestone in the study of chaos. Since then, the study of chaos has permeated almost all fields of natural science and social science. In 1979, Rössler discovered the first four-dimensional hyperchaotic system, i.e., the Rössler hyperchaotic system [21]. Compared with chaotic phenomena, hyperchaos expands in two or more directions and has at least two positive Lyapunov exponents. So the lowest dimension of hyperchaotic system is four, and there is at least one nonlinear term. This makes the hyperchaotic system present more complex dynamics, showing stronger randomness and unpredictability.

Zhang et al. [22] proposed a three-dimensional continuous autonomous chaotic system with the equation of state shown in the following equation:

$$\begin{cases} \dot{x} = \alpha x + yz \\ \dot{y} = -x + cy \\ \dot{z} = dy^2 - bz. \end{cases} \tag{1}$$

In equation (1), x, y, and z are system state variables and α, b, c, and d are system real parameters. When $a = 20, b = 5, c = 10$, $an\ d\ d = 7$, the system can produce chaotic attractors, and the three Lyapunov exponents of the system are $L_1 = 1.2371, L_2 = -0.0291$, and $L_3 = -16.4484$. The structure of the system formed under such conditions is more complex than that of the low-dimensional system, which can produce chaotic sequences of certain combinatorial forms. It makes the design of sequences more flexible and can better meet the application requirements. Given the initial values in the specific application, the corresponding sequences x, y, and z can be determined. The three are arranged in ascending order to determine x', y', and z'. The set of replacement addresses, corresponding to X, Y, and Z, is obtained by performing a certain position comparison. The image pixel position matrix can be scrambled by this sequence in a specific application. The 3 index sequences are determined and some scrambling operations are performed by them.

Based on this system, a new four-dimensional hyperchaotic system is constructed by first modifying three equations, followed by introducing a fourth-dimensional state variable using the state feedback control method and the second equation used for the introduced variable. Its state equation is shown in the following equation:

$$\begin{cases} \dot{x} = a(y - x) + bw \\ \dot{y} = -cx + 2\ dy\ z \\ \dot{z} = h - fy^2 - ksin(z) \\ \dot{w} = gx. \end{cases} \tag{2}$$

In equation (2), c and g are nonzero real numbers. x, y, z, and w are the state variables of the system. a, b, c, d, and r are the control parameters of the system. When $(a, b, c, d, g, f, h,$ and $k)$ is equal to $(15, 3, 8, 8, 2.7, 1, 3,$ and $2)$, respectively, the system behaves as hyperchaotic motion. The corresponding Lyapunov exponents are $L_1 = 0.6307, L_2 = 0.2868, L_3 = 0.0001,$ and $L_4 = -9.0857$. One of the methods and criteria for determining chaos is the Lyapunov exponent. If a positive Lyapunov exponent is obtained, the system is determined to be chaotic. If more than one positive Lyapunov exponent is obtained, the system is determined to be hyperchaotic.

When $(a, b, c, d, g, f, h,$ and $k)$ is equal to $(15, 3, 8, 8, 2.7, 1, 3,$ and $2)$, respectively, the system can generate topologically complex hyperchaotic attractors. The kinetic equations of equation (2) are calculated using the built-in Runge–Kutta function (ode45) in Matlab2020b and solved to obtain four one-dimensional chaotic sequences. Figures 1(a)–1(d) show the four chaotic attractor phase diagrams of *x-y-z*, *x-y-h*, *x-z-h*, and *y-z-h* for the last 40,000 data from each of the four sets of data obtained by using ode45 to calculate the hyperchaotic system in this paper.

From Figure 1, the system has the following characteristics: (a) The system has high dimensionality and can generate four different chaotic sequences. (b) The system structure is complex and the generated chaotic sequences have higher entropy values. (c) The four initial values of the system greatly affect the generated chaotic sequences, and all four initial values can be used as keys, which increase the key capacity and the difficulty of breaking the system.

2.2. Keccak Algorithm. Keccak [26, 27] is a standard one-way hash function algorithm. NIST's evaluation of Keccak is that the algorithm has very good security and implementation. Especially, it is designed in a completely different way compared to SHA-2, avoiding many known attacks and providing some performance that SHA-2 does not have. Keccak can generate hash values of any length, but to match the SHA-2 hash length, the SHA-3 standard specifies four output length versions: 224 bit, 256 bit, 384 bit, and 512 bit. In terms of the maximum length of the input data, SHA-3 is $2^{64} - 1$ bits, SHA-2 is $2^{128} - 1$ bits, and SHA-3 has no length limit. Keccak uses a sponge construction which is completely different from the SHA-1 and SHA-2 algorithms. In sponge construction, after the input data is filled, it goes through an absorbing phase and a squeezing phase to generate the output hash. The hash function can calculate a fixed-length hash value based on a message of any length. Attaching the hash value to the message or storing it together with the message can prevent the message from being modified during storage or transmission. Different messages have different hash values. As long as one bit changes in the message, the hash value will be completely different. Using this feature, by selecting the appropriate message, the hash value generated by the Keccak hash function is used to perform operations on the image to change the pixel value of the image. At the same time, the hash value is modified to set the initial value and system parameters of the chaotic system, so as to further improve the security of encryption. Keccak has no limit on the upper limit of the length of the input data and can generate any degree of hash value.

After the original image is converted with Keccak, a set of 512-bit hash values will be generated: 9caa44db566cfe1-f6a98c4991fffe891bb7d7fdf840449a026e923e9feab60b8b7e-d7a3933a757358c2c9441366976fab4bda222f9b5e4d-f814322e0dc12c13f. The generated hash values are used as input information for the next hash function to generate new hash values. The cycle is generated eight times to obtain a total of 256×8 bit hash values. A DNA encoding rule is chosen to encode the obtained hash values, and every 8-bit group of hash value is encoded to convert the 256×8 bit hash values into a 16×16 DNA encoding matrix. For example, according to the first encoding rule: db $\longrightarrow$ 11011011 $\longrightarrow$ TGCT.

In this article, the hash value K is generated by the Keccak algorithm, and then the initial value of the chaotic system is generated. Dividing K by bytes, it can be expressed as $k_1, k_2, k_3, \cdots k_{64}$. The initial value of the hyperchaotic system is calculated by the following formulae:

$$h_j = \frac{\left(k_{j+1} \oplus k_{j+2} \oplus k_{j+3}\right) + k_{j+4} + k_{j+5} + k_{j+6}}{256}, \tag{3}$$

$$\begin{cases} x_0 = 1 + abs\left(roun\ d\left(h_1\right) - h_1\right), \\ y_0 = 1 + abs\left(roun\ d\left(h_2\right) - h_2\right), \\ z_0 = 1 + abs\left(roun\ d\left(h_3\right) - h_3\right), \\ w_0 = 1 + abs\left(roun\ d\left(h_4\right) - h_4\right). \end{cases} \tag{4}$$

Among them, $j = 6(i - 1)$, where $i = 1$, 2, 3, and 4. The keys generated in this way have the advantages of good randomness, periodicity, and long key space properties. By combining the original image information with the key, the algorithm will effectively resist known plaintext and selected plaintext attacks.

2.3. Dynamic DNA Coding Technology. At present, the scale of nucleic acid databases has increased substantially, and the corresponding growth rate can be described by an exponential law. The corresponding data capacity is very large and can be regarded as a natural code book. DNA sequence [11] is mainly used for ciphertext diffusion and hash value generation. The DNA molecule consists of adenine (A), cytosine (C), guanine (G), and thymine (T). A specific analysis shows that a DNA molecule can be formed by

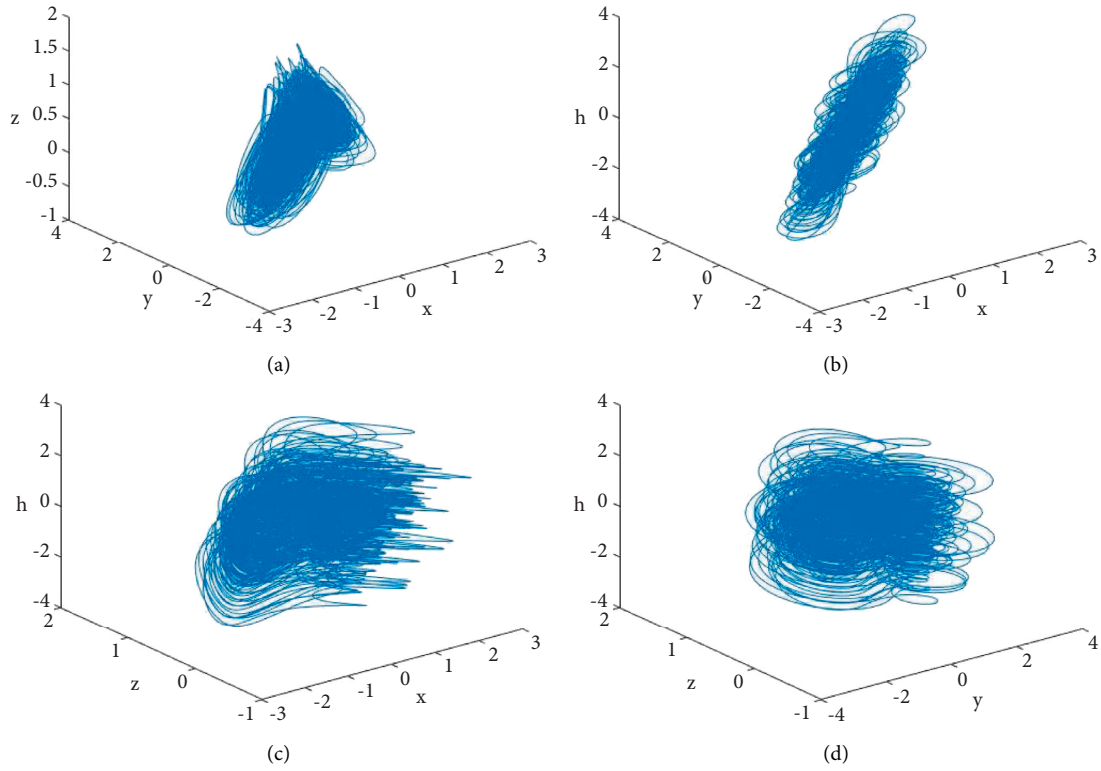


Figure 1: Phase diagram of the new four-dimensional hyperchaotic system. (a) *x-y-z* chaos attractor phase diagram, (b) *x-y-h* chaos attractor phase diagram, (c) *x-z-h* chaos attractor phase diagram, and (d) *z-y-h* chaos attractor phase diagram.

binding two single-stranded DNA molecules under the action of hydrogen bonds. The principle of complementary base pairing in this binding process is related to the base characteristics, and the corresponding pairing rules are expressed as hydrogen bond pairing of A and T and hydrogen bond pairing of G and C [11]. This combination of characteristics is similar to the binary formed by semiconductor pass-throughs. Thus, it is possible to store and process information on the basis of such combinations and meet certain requirements for computational analysis [25, 28].

2.3.1. Coding Rules. If we proceed according to the A ⟶ 00, B ⟶ 01, C ⟶ 10, and T ⟶ 11 rule, we match the complementary pairing A–T and C–G of the base pair. The relevant complementary pairing rules in this case are specified in Table 1.

The grayscale value of each pixel of the grayscale image is described by the corresponding 8-bit binary number. In the case of DNA encoding, a simple 4-base sequence is encoded and then converted into a DNA sequence, and the conversion rules for the DNA sequence are used in the image processing. In the encrypted image, the following base substitution rules are set to meet the interference requirements and to improve the confidentiality.

2.3.2. Base Substitution Rules. Setting a specific mapping function $L(x)$, the following relational rules are determined.

Table 1: Coding rules.

Rule	1	2	3	4	5	6	7	8
00	A	A	C	G	C	G	T	T
01	C	G	A	A	T	T	C	G
10	G	C	T	T	A	A	G	C
11	T	T	G	C	G	C	A	A

$$\begin{cases} x \neq L(x) \neq L(L(x)) \neq L(L(L(x))) \\ x = L(L(L(L(x)))). \end{cases} \tag{5}$$

Here, $x \in \{A, C, G, T\}$, there are six reasonable combinations of base substitutions according to this convention.

The permutation process of the images can be scrambled by randomly selecting any of the permutation combinations in Table 2 according to the application requirements, based on which the encoding is performed.

2.3.3. Base Algebraic Operation Rules. A ⟶ 00, C ⟶ 01, G ⟶ 10, and T ⟶ 11 codes are set according to the complementary pairing rules. Exclusive OR, addition, and subtraction rules of the DNA are shown in Tables 3–5, respectively. For other codes, similar operation rules are determined on these basis.

In this article, Keccak algorithm is used to generate hash value K for DNA sequence, and the length of K is 512 bits. The gray value diffusion is used to process the image pixel gray value and DNA sequence by base operation. During this

Table 2: Base substitution rules.

1	A ⟶ T ⟶ C ⟶ G ⟶ A
2	A ⟶ T ⟶ G ⟶ C ⟶ A
3	A ⟶ C ⟶ T ⟶ G ⟶ A
4	A ⟶ C ⟶ G ⟶ T ⟶ A
5	A ⟶ G ⟶ T ⟶ C ⟶ A
6	A ⟶ G ⟶ C ⟶ T ⟶ A

Table 3: Exclusive OR operation rules.

XOR	A	C	G	T
A	A	C	G	T
C	C	A	T	G
G	G	T	A	C
T	T	G	C	A

Table 4: Addition operation rules.

XOR	A	C	G	T
A	A	C	G	T
C	C	G	T	A
G	G	T	A	C
T	T	A	C	G

Table 5: Subtraction operation rules.

XOR	A	C	G	T
A	A	T	G	C
C	C	A	T	G
G	G	C	A	T
T	T	G	C	A

operation, the starting base position R of the sequence must be set. The dynamic DNA coding technology is mainly based on the analysis of the position in the pixel matrix and the hash value K in the coding process, and then the appropriate coding rules are determined.

The corresponding DNA coding rules are as follows:

$$R_{i,j} = Mo\ d((i-1)*N+j;8) \oplus Bin2\ de\ c\left(k_s k_{s+1} k_{s+2}\right). \quad (6)$$

Among them, $i \in \{1, 2, \ldots, M\}$, $j \in \{1, 2, \ldots, N\}$, and $s = Mo\ d((i-1)*N+j-1, 510)+1$. $K_s K_{s+1} K_{s+2}$ are composed of three binary bits of s bit, $s+1$ bit, and $s+2$ bit, respectively, of the hash value K. Since each pixel value of the image can be represented by 8-bit binary, and each pixel corresponds to 4 bases, it can be determined that the length of the DNA_S after the encoding process is $4 \times M \times N$.

3. Encryption Algorithm Design

The algorithm in this article is divided into two parts. First, pixel position scrambling transformation is performed. In the operation process, a permutation index is constructed based on the Lorentz-chaotic sequence to scramble the image. Each pixel of the original image is converted into DNA sequence information, and ciphertext feedback processing is performed according to a certain sequence to achieve the purpose of replacement. The encryption flowchart is shown in Figure 2, and the details of the encryption process are as follows:

Step 1: input the gray image I and determine the output size as two-dimensional matrix $I_1 = M \times N$.

Step 2: obtain the index sequence X from formula (1) and scramble the matrix I_1 to obtain a new matrix I_2.

Step 3: use dynamic DNA coding to process I_2 and obtain a new DNA coding matrix I_3.

Step 4: download the DNA sequence of the gene bank, and intercept $4 \times M \times N$ base sequences from R to form a matrix I'.

Step 5: XOR the base sequences corresponding to I_3 and I' to obtain a matrix I_4 and scramble it with the index sequence Y obtained by Lorenz mapping to obtain a new matrix I_5.

Step 6: use formula (2) to generate the DNA sequence P, and then determine the number of base substitutions according to the conversion rule. Select the corresponding rules from Table 2 to replace and form the corresponding code matrix I_6.

Step 7: after replacement, select a DNA encoding rule. Then, the binary code is formed by the replacement process, the gray value is formed by the conversion process, and the matrix I_7 is formed by the restoration process. The corresponding replacement expression is as follows:

$$\begin{cases} x_i = x_i, & P_i = 0, \\ x_i = L(x_i), & P_i = 1, \\ x_i = L(L(x_i)), & P_i = 2, \\ x_i = L(L(L(x_i))), & P_i = 3. \end{cases} \quad (7)$$

Step 8: determine the index sequence Z according to the Lorenz mapping equation (1), scramble the matrix I_7, obtain the encrypted matrix I_8, and output the corresponding ciphertext. The decryption algorithm only needs to reverse the above steps.

This algorithm also meets the applicability requirements for color image encryption. During the processing, RGB decomposition is simply performed, and then the same operation is performed.

4. Experimental Simulation Results and Analysis

The algorithm proposed in this article is used to simulate several different images. This algorithm can be used to encrypt images of any size. Figure 3 shows the encrypted and decrypted keyframe images from the visual data monitored in the industrial network. In Figure 3, the experimental results show that after using the encryption algorithm to encrypt the plaintext image, no information about the plaintext image can be obtained from the encrypted image. Thus, our proposed image encryption algorithm can

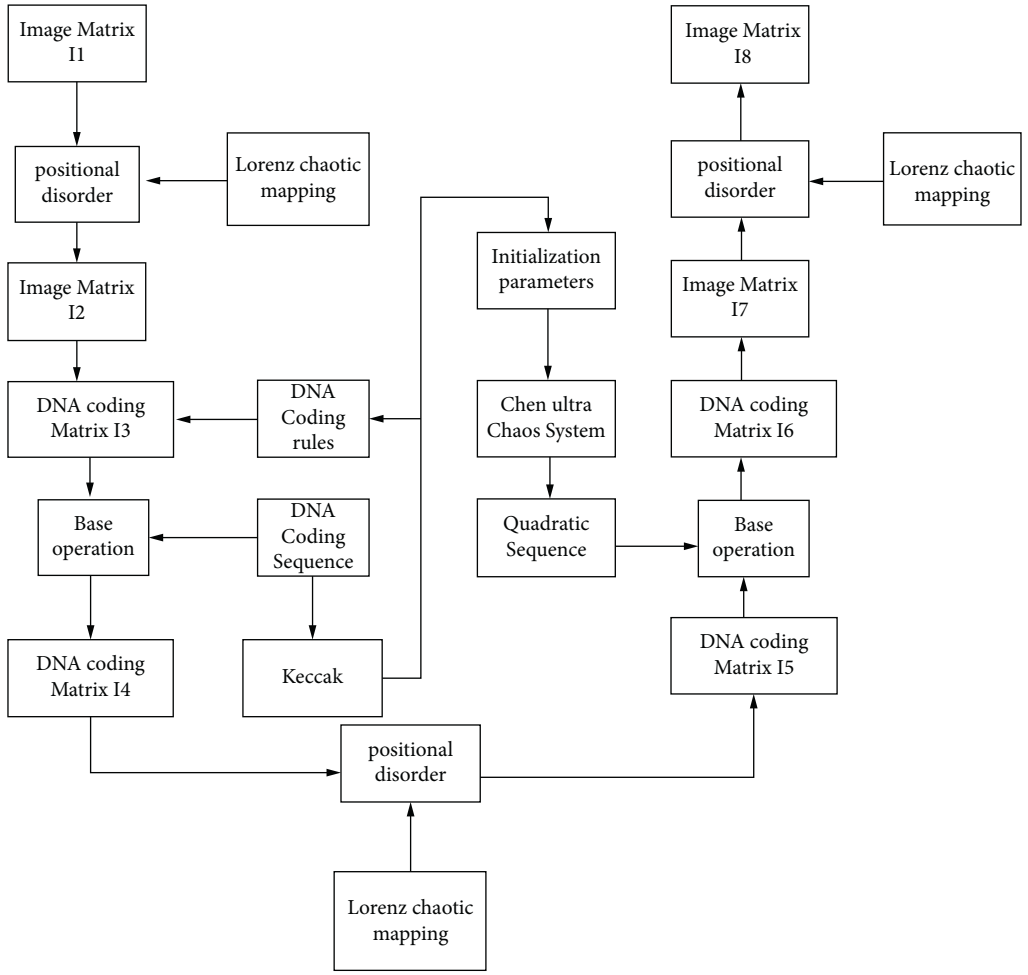


Figure 2: Encryption flowchart.

withstand statistical attacks, and the encryption algorithm has a better effect. Moreover, this algorithm is lossless, and the decrypted image is exactly the same as the original image.

4.1. Key Space and Sensitivity Analysis. The key space refers to the range of the key size used in the encryption and decryption process. The larger the value range of the key is, the larger the corresponding key space is. In theory, if the key space of the encryption system is greater than 2^{128}, it is believed that the encryption system can effectively resist exhaustive attack. In the encryption algorithm of this paper, set $(x_0, y_0, z_0, x_1, y_1, z_1,$ and $w_1)$ as the key. The accuracy of the initial value of the hyperchaotic system of the algorithm in this article can reach 10^{96}, which reaches a very high level with such a large space key. So, it can effectively resist exhaustive attack.

If a small change in the key causes the decrypted image to be completely unrelated to the original image, the key is said to be sensitive, and this property makes it possible to improve the resistance to cracking by increasing the key by a certain number of decimal places. Figure 4(d) shows the decrypted image under the wrong key obtained by changing (0), one of the keys used in encrypting the image, from 0.4985 used in encryption to just 0.4985000000000001 in decryption. It means that the key sensitivity of this algorithm is extremely strong, and even if one of the keys is changed very slightly during decryption, the decryption result is completely different from the original image without any correlation, and the number of keys of this algorithm is very large, which can completely resist the attack by exhaustive keys.

4.2. Analysis of Information Entropy. Information entropy can be used to describe the uncertainty of objects. Its calculation formula is as follows:

$$H(m) = -\sum_{k=0}^{2^N-1} p(m_i)\log_2 p(m_i). \tag{8}$$

A specific analysis shows that for grayscale images, the total number of states of information m is 256. When the information entropy is 8, the information is completely random. According to Table 6, the corresponding information entropy determined by encrypting the test images with the algorithm of this article are close to 8, and the information leakage of the image is very little. Therefore, the image encryption algorithm proposed in this paper generates secure encrypted images with randomness.

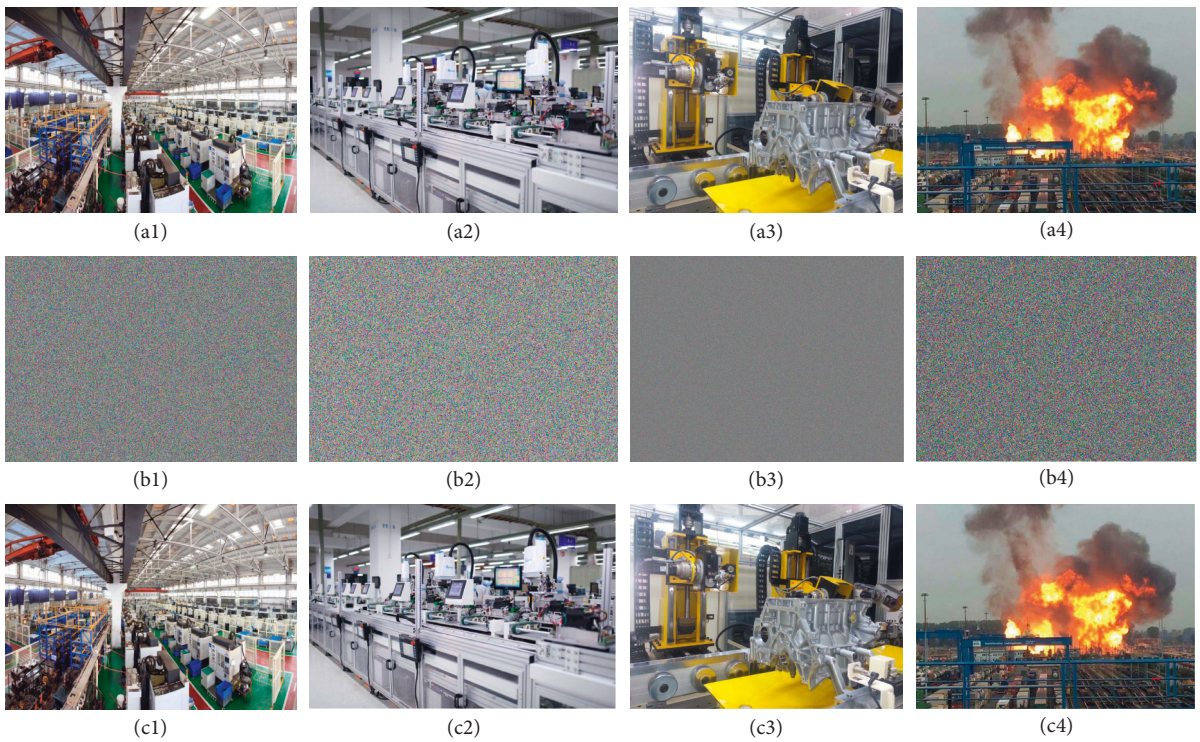


Figure 3: The results of plain, cipher, and decrypted images. (a.i) The plain images, (b.i) the encrypted images, and (c.i) the decrypted images (from left to right, $i \in \{1, 2, 3, 4\}$).

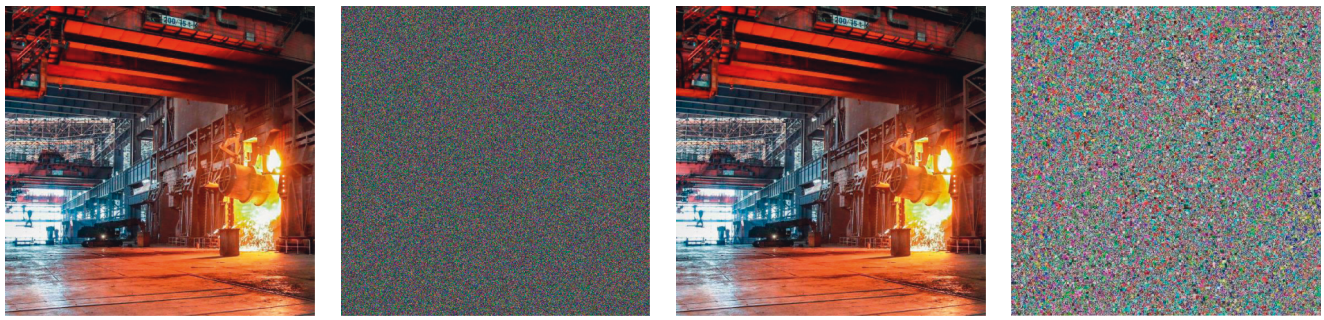


Figure 4: Decrypt image in case of key error. (a) Plain images, (b) encrypted images using the secret key, (c) the decrypted images using the secret key, and (d) small change in key.

Table 6: Information entropy tests.

Name	Plain image			Cipher image		
	R	G	B	R	G	B
Image 1	7.9216	7.9191	7.9257	7.9997	7.9997	7.9998
Image 2	7.7586	7.744	7.749	7.9993	7.9993	7.9993
Image 3	7.8626	7.8017	7.8177	8	8	8
Image 4	7.5443	7.455	7.3033	7.9992	7.9992	7.9991
Image 5	7.9538	7.5324	7.2105	7.9997	7.9998	7.9997

4.3. Analysis of Histogram. An image histogram shows the distribution of the pixel intensity values, and it provides some statistical information of the image. A secure image encryption system can make the encrypted image have a uniform histogram to resist any statistical attacks.

As shown in Figure 5, we can see that the histograms of all three channels of the original image are undulating, while the histograms of all three channels of the ciphertext image are flatly distributed with pseudorandomness, which can hide the statistical properties of the original image, and thus can effectively resist large-scale histogram-based statistical attacks against the image.

4.4. Correlation Analysis. The closer the values between adjacent pixels of an image are, the higher the correlation between adjacent pixels is. The plaintext images have high information redundancy and high correlation of neighboring pixels. In general, the original image has high correlation close to 1. Therefore, image encryption should be

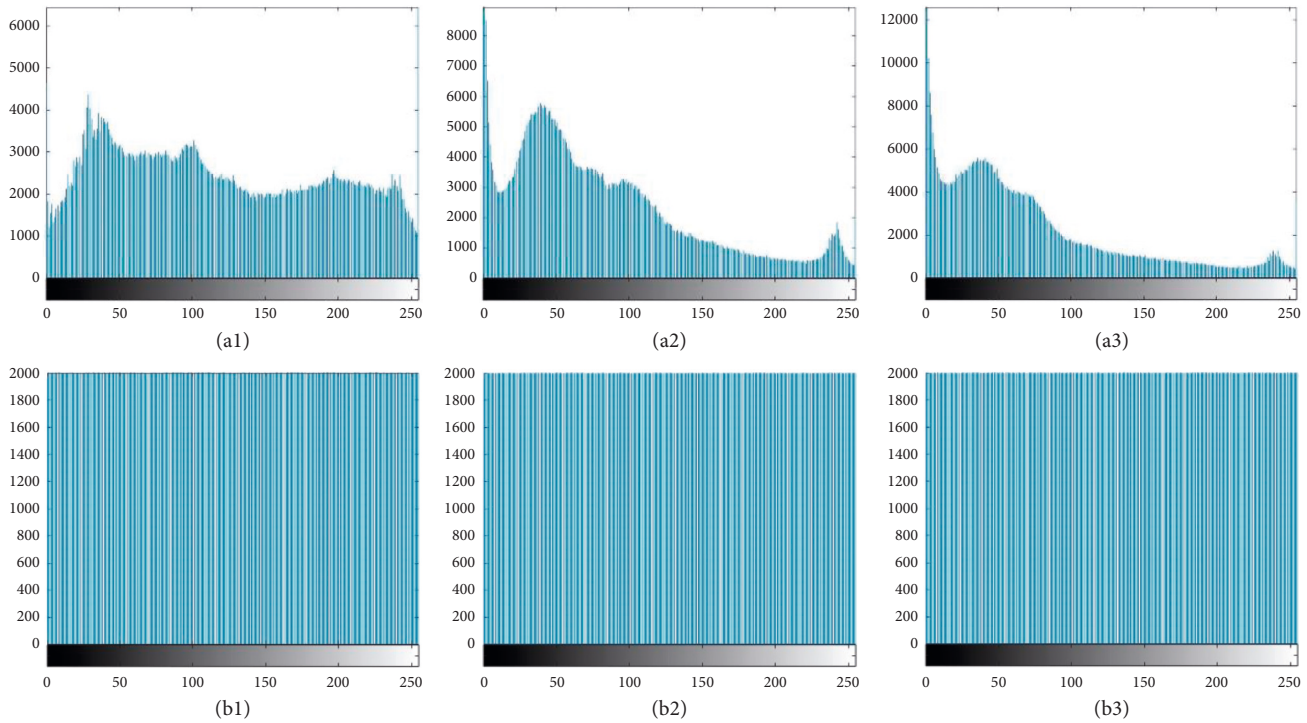


Figure 5: The histogram of the plain and encrypted images. (a.i) The R, G, and B histograms of the plain images, respectively. (b.i) The R, G, and B histograms of the encrypted images, respectively (from left to right, $i \in \{1, 2, 3\}$).

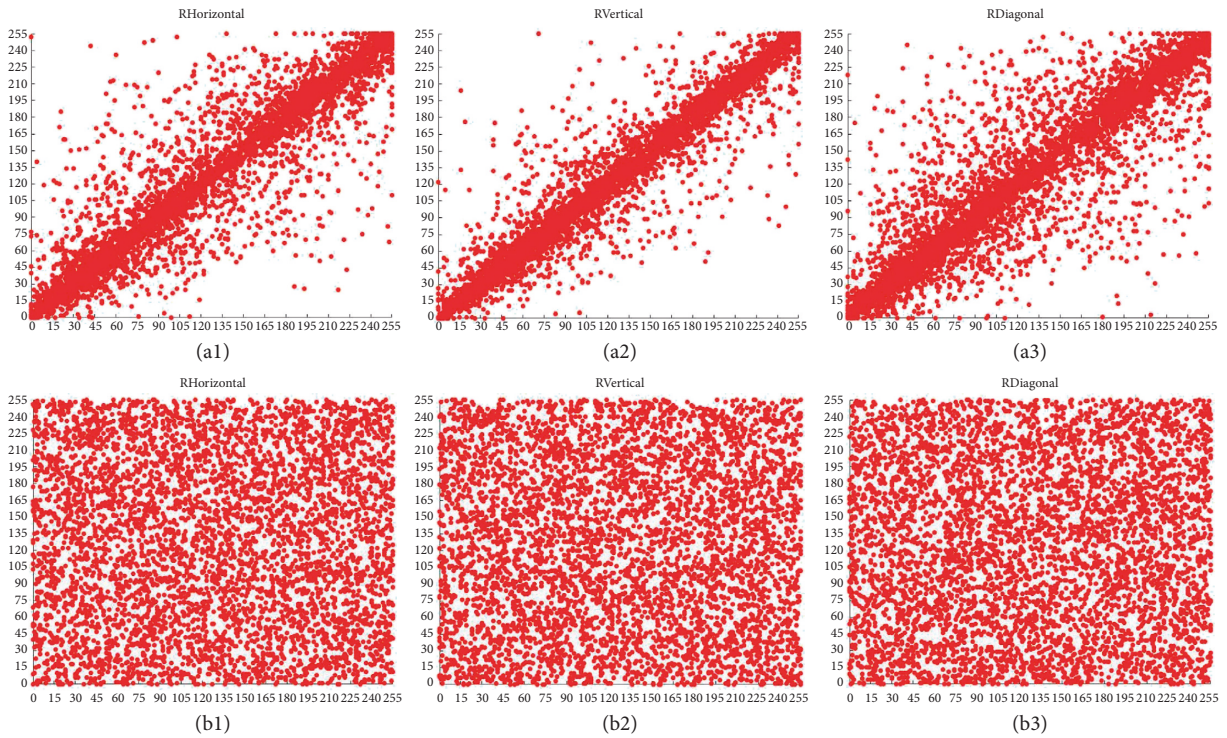


Figure 6: Correlation distribution of adjacent pixels in the plain and encrypted images in the red channel. (a.i) The horizontal, vertical, and diagonal directions of the plain images, respectively. (b.i) The horizontal, vertical, and diagonal directions of the encrypted images, respectively (from left to right, $i \in \{1, 2, 3\}$).

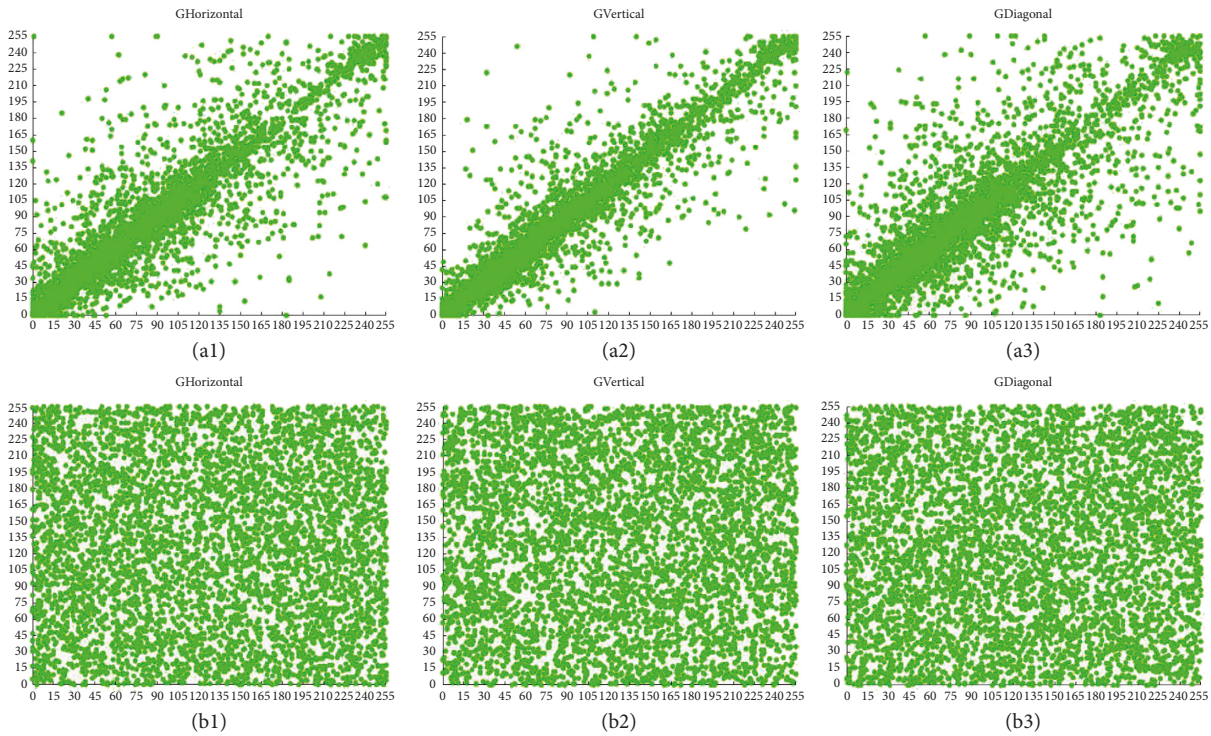


FIGURE 7: Correlation distribution of adjacent pixels in the plain and encrypted images in the green channel. (a.i) The horizontal, vertical, and diagonal directions of the plain images, respectively. (b.i) The horizontal, vertical, and diagonal directions of the encrypted images respectively (from left to right, $i \in \{1, 2, 3\}$).

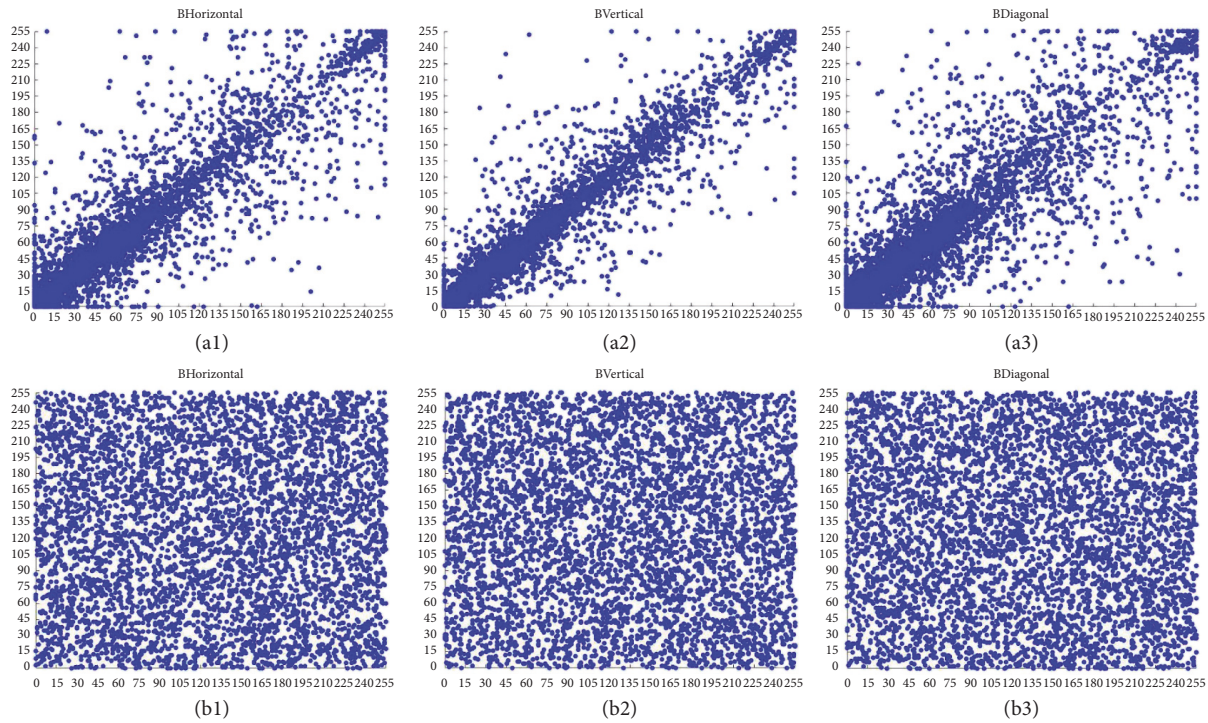


FIGURE 8: Correlation distribution of adjacent pixels in the plain and encrypted images in the blue channel. (a.i) The horizontal, vertical, and diagonal directions of the plain images, respectively. (b.i) The horizontal, vertical, and diagonal directions of the encrypted images, respectively (from left to right, $i \in \{1, 2, 3\}$).

Table 7: Relation of comparison of adjacent pixels.

	Component	Plain image			Cipher image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Image 1	R	0.94851	0.94510	0.90871	0.00193	0.00912	0.01150
	G	0.94898	0.94382	0.90848	−0.01287	−0.01729	−0.00829
	B	0.94954	0.94472	0.91027	0.00068	0.01071	0.01067
Image 2	R	0.95757	0.9382	0.90709	−0.00121	−0.01014	−0.00361
	G	0.95744	0.93797	0.90702	−0.00402	0.00738	0.003608
	B	0.95891	0.94027	0.9108	−0.00642	−0.00624	−0.00439
Image 3	R	0.99273	0.99452	0.98797	0.00875	−0.01873	−0.00453
	G	0.99219	0.99411	0.98713	0.00603	0.00229	0.00609
	B	0.99377	0.99542	0.98713	−0.0107	−0.01102	0.00624
Image 4	R	0.98484	0.96936	0.95865	0.00531	0.017608	0.02200
	G	0.97993	0.96087	0.94654	0.014381	0.00517	−0.03671
	B	0.9758	0.9494	0.93226	0.00894	−0.00344	−0.00179
Image 5	R	0.95749	0.92225	0.90197	0.11934	−0.00692	−0.01505
	G	0.94522	0.91011	0.88341	0.01001	−0.01374	−0.01734
	B	0.94662	0.91398	0.88722	−0.01810	0.00256	−0.01048

able to eliminate these correlations, and the ideal value of correlation for encrypted images should be 0 [29]. The mathematical expressions for the correlation calculation of adjacent pixels (r_{xy}) are shown as follows:

$$\begin{aligned} r_{xy} &= \frac{cov(x, y)}{\sqrt{D(x)D(y)}}, \\ cov(x, y) &= \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)\left(y_i - E(y)\right), \\ E(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i, \\ D(x) &= \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)^2, \end{aligned} \tag{9}$$

where x and y are the data values at adjacent positions, N is the log 5000 of the taken pixel points, $E(x)$ is the mean of the taken pixels, $D(x)$ is the variance, $cov(x, y)$ denotes the correlation function, and r_{xy} is the correlation coefficient. And the larger its absolute value, the stronger the correlation.

In order to resist statistical analysis attacks, it is necessary to break the strong correlation between pixels. We use a statistical test of the correlation of two adjacent pixels in an encrypted keyframe industrial image. We randomly select 5000 pixels in the keyframe. The correlation of the corresponding adjacent pixels in each channel of the RGB space of the color image are tested in the horizontal, vertical, and diagonal directions. Figures 6–8 give the visual results of the correlation distribution of two adjacent pixels in the horizontal, vertical, and diagonal directions of the keyframe image as well as the corresponding encrypted image frame image. The first row is the original image, while the second row is the encrypted image. It can be noticed that the correlations of the original image and the encrypted image are very different. The points in the plot of the correlation of the encrypted image have a good uniform probability distribution while those in the original image are concentrated on the diagonal line in the plot.

In this paper, correlations are calculated in three channels of industrial images along horizontal, vertical, and diagonal directions. Table 7 shows the correlation results of this experiment with different original images and their encrypted images. The statistical results show that the pixel correlation of the original image is very strong, while the correlation coefficient between adjacent pixels of the ciphertext image is close to zero. The algorithm in this article disrupts the correlation between pixels and resists statistical analysis attacks.

5. Conclusion

Video surveillance networks in industrial environments are growing rapidly due to the complementary role of the IOT, but at the same time, a large amount of redundant video data is being generated. This makes the transmission, analysis, and management of images difficult and challenging. This article proposes a four-dimensional hyperchaos and DNA genetics calculation to improve the image encryption method of the chaotic system. This method uses the hash value K generated by the Keccak algorithm as the initial value of the hyperchaotic system, so that the pseudorandom chaotic sequence generated by it can scramble the position of the pixel. Then, make the pixels change dynamically with the DNA code, so that the algorithm has better confusion and diffusion characteristics. Finally, a security analysis is carried out with simulation experiments, which further confirms that the image encryption algorithm proposed in this article is highly secure and can resist various types of attacks.

Data Availability

The data used to support the findings of this study are included in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Key Technology Research Project of Complex Multi-User Wireless Routing Management System (no. 2018XJZD005), basic scientific research ability enhancement project for Young and Middle-Aged Teachers in Guangxi Universities (no. 2021KY0621), and basic scientific research ability enhancement project for Young and Middle-Aged Teachers in Guangxi Universities (no. 2022KY0605).

References

[1] N. Magaia, R. Fonseca, K. Muhammad, A. H. F. N. Segundo, A. V. Lira Neto, and V. H. C. de Albuquerque, "Industrial internet-of-things security enhanced with deep learning approaches for smart cities," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6393–6405, 2021.

[2] Z. Lv, L. Qiao, J. Li, and H. Song, "Deep-learning-enabled security issues in the internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9531–9538, 2021.

[3] S. Nižetić, P. Šolić, D. López-de-Ipiña González-de-Artaza, and L. Patrono, "Internet of Things (IoT): opportunities, issues and challenges towards a smart and sustainable future," *Journal of Cleaner Production*, vol. 274, p. 122877, 2020.

[4] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali, and R. Jain, "Recent advances in the internet-of-medical-things (IoMT) systems security," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8707–8718, 2021.

[5] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, "Challenges and opportunities in securing the industrial internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 2985–2996, 2021.

[6] H. Tran-Dang, N. Krommenacker, P. Charpentier, and D.-S. Kim, "Toward the internet of Things for physical internet: perspectives and challenges," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4711–4736, 2020.

[7] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for IoV in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[8] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He, and X. Zhou, "TripRes," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 2, pp. 1–21, 2021.

[9] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[10] M. Sajjad, I. Mehmood, and S. Baik, "Sparse representations-based super-resolution of key-frames extracted from frames-sequences generated by a visual sensor network," *Sensors*, vol. 14, no. 2, pp. 3652–3674, 2014.

[11] I. Mehmood, M. Sajjad, W. Ejaz, and S. W. Baik, "Saliency-directed prioritization of visual data in wireless surveillance networks," *Information Fusion*, vol. 24, pp. 16–30, 2015.

[12] C. Li, Y. Zhang, and E. Y. Xie, "When an attacker meets a cipher-image in 2018: a year in review," *Journal of Information Security and Applications*, vol. 48, p. 102361, 2019.

[13] K. J. Persohn and R. J. Povinelli, "Analyzing logistic map pseudorandom number generators for periodicity induced by finite precision floating-point representation," *Chaos, Solitons & Fractals*, vol. 45, no. 3, pp. 238–245, 2012.

[14] D. Yoshioka and K. Kawano, "Periodic properties of Chebyshev polynomial sequences over the residue ring $\mathbb{Z}/2{k}\mathbb{Z}$," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 8, pp. 778–782, 2016.

[15] C. Li, B. Feng, S. Li, J. Kurths, and G. Chen, "Dynamic analysis of digital chaotic maps via state-mapping networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 6, pp. 2322–2335, 2019.

[16] J. Wu, X. Liao, and B. Yang, "Image encryption using 2D Hénon-Sine map and DNA approach," *Signal Processing*, vol. 153, pp. 11–23, 2018.

[17] X. Chai, Z. Gan, K. Yuan, Y. Chen, and X. Liu, "A novel image encryption scheme based on DNA sequence operations and chaotic systems," *Neural Computing & Applications*, vol. 31, no. 1, pp. 219–237, 2017.

[18] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Optics and Lasers in Engineering*, vol. 88, pp. 197–213, 2017.

[19] H. Wen, S. Yu, and J. Lü, "Breaking an image encryption algorithm based on DNA encoding and spatiotemporal chaos," *Entropy*, vol. 21, no. 3, p. 246, 2019.

[20] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[21] O. E. Rössler, "An equation for hyper chaos," *Physics Letters A*, vol. 71, no. 2-3, pp. 155–157, 1979.

[22] C.-X. Zhang, S.-M. Yu, and Y. Zhang, "Design and realization of multi-wing chaotic attractors via switching control," *International Journal of Modern Physics B*, vol. 25, no. 16, pp. 2183–2194, 2011.

[23] W. J. Ruan and Q. G. Yang, "Research on complex dynamics of a new four-dimensional hyperchaotic system with finite and infinite isolated singularities," *Journal of Guangxi Normal University*, vol. 03, 2021.

[24] J. X. Zhao and X. F. Zhang, "Spatiotemporal color image encryption method based on combined chaotic systems," *Computer Engineering and Design*, vol. 37, no. 9, 2016.

[25] Y. G. Huang, Y. X. Du, and W. Shi, "Image encryption algorithm based on a novel combinatorial chaotic mapping," *Microelectronics & Computer*, vol. 36, no. 5, pp. 47–52, 2019.

[26] G. Bertoni and J. Daement, "The Keccak sponge function family [EB/OL]," 2010, https://Keccak.noekeon.org.

[27] P. Morawiecki, J. Pieprzykand, and M. Srebrny, "Rotational crypt analysis of round-reduced Keccak [EB/OL]," 2018, https://eprint.iacr.org.

[28] K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H. Wang, and S. W. Baik, "Secure surveillance framework for IoT systems using probabilistic image encryption," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3679–3689, 2018.

[29] B. Norouzi, S. Mirzakuchaki, S. M. Seyedzadeh, and M. R. Mosavi, "A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process," *Multimedia Tools and Applications*, vol. 71, no. 3, pp. 1469–1497, 2014.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 4490757, 14 pages
https://doi.org/10.1155/2022/4490757

Research Article

Research on IoT Forensics System Based on Blockchain Technology

Guangjun Liang,[1,2,3] Jianfang Xin,[4] Qun Wang,[1,2] Xueli Ni,[1,2,3] and Xiangmin Guo[1,2,3]

[1]*Department of Computer Information and Cyber Security, Jiangsu Police Institute, Nanjing, China*
[2]*Engineering Research Center of Electronic Data Forensics Analysis, Jiangsu Province, Nanjing, China*
[3]*Key Laboratory of Digital Forensics, Department of Public Security of Jiangsu Province, Nanjing, China*
[4]*School of Intelligent Engineering, Nanjing Institute of Railway Technology, Nanjing, China*

Correspondence should be addressed to Jianfang Xin; xinjfang@163.com

Received 20 August 2021; Accepted 4 May 2022; Published 15 June 2022

Academic Editor: Gautam Srivastava

In recent years, mobile edge computing (MEC) has become a research hotspot in academia. The Internet of Things (IoT) is an excellent way to build the infrastructure required for a MEC environment. Its rich digital tracking repository can provide insights into people's daily activities at home and elsewhere. Meanwhile, due to the open connectivity of the Internet of things devices, they can easily become the target of network attacks and be used by criminals as criminal tools. As a result, civil and criminal cases have increased year by year. This article conducts in-depth research on IoT forensics. By comparing its difference with traditional digital forensics (DF), the definition of IoT forensics is given. We have systematically sorted out the research results since the concept of IoT forensics was proposed in 2013 and proposed a generalized IoT forensics model. By studying blockchain technology and introducing it into the IoT forensics framework, a blockchain-based IoT forensics architecture is further proposed. Further, an alliance chain IoT forensics system is proposed. From the perspective of the data provider and the data visitor, the process of evidence storage and forensics of the IoT system is discussed. Finally, taking Unmanned Aerial Vehicle (UAV) forensics as an example, we give an experiment of IoT forensics analysis.

1. Introduction

The use of the IoT in our daily life leads to two phenomena. First, the use of intelligent Internet of things related devices makes people leave digital traces on these devices, and the data of users' daily activities can be obtained by tracing the personal information stored in the database [1]. Secondly, the number of cases of online fraud is increasing year by year, and various Internet of things devices may become network attack objects or criminal tools. The security vulnerabilities in the IoT system can easily be used by criminals as a means of remote control. In short, public security organs urgently need Internet of things forensics workers to help determine the key information of the case.

The IoT technology can send and receive information between two or more interconnected devices through the internet. As IoT is widely used in various industries such as industry, commerce, and agriculture, IoT devices such as smart sensors have brought huge security risks to users. Liang and Kim [2] conducted research on IoT security, discussed applications in edge computing and blockchain scenarios, and pointed out that machine learning may be a better solution. Regarding IoT platforms and systems, Zhou [3] discussed the lessons learned from these bugs. Miloslavskaya and Tolstoy [4] were concerned about the typical attack problems of IoT assets, and they proposed to find possible security vulnerabilities through intelligent security protection of the Internet of things.

The necessity for scholars to study data forensics technology is that the increase of Internet of things devices increases computer-related crimes. Al-Masri et al. [5] introduced a Fog-based IoT Forensics Framework (FoBI).

Some key problems in digital forensics (DF) are studied, and corresponding solutions are given. The data forensics investigation process includes three steps: data collection, investigation, and investigation results. Investigators further compared digital evidence on different devices. Silvarajoo et al. [6] use appropriate case management tools hosted on the Web to simplify information collection and consolidate data. Because the data formats of different platforms are quite different, the lack of a unified standard has brought great trouble to the follow-up investigation and evidence collection. A management strategy for unified format and shared data is proposed. To assist the FBI in identifying suspects, Elhoseny et al. [7] proposed an optimal deep learning-based convolutional neural network (ODL-CNN).

Due to the distributed storage, decentralized management, and nontampering characteristics of blockchain, this emerging technology can be widely used in important industries such as medical treatment, commerce, information technology sector, and agriculture. Su et al. [8] discussed the sharing scheme in the financial field and pointed out that the most difficult to solve is the security of data-sharing. A data sharing model based on blockchain is proposed, and the technical scheme in the process of establishing the model is given in detail. Sathya et al. [9] focus on the blockchain-based food supply chain field, introduce smart contracts, and propose a supply chain management architecture based on Ethereum. In the food supply chain, there are fewer external attacks, and more research should be done on food traceability, prevention of forged data, server tampering, and other malicious behaviors. Agyekum et al. [10] proposed a blockchain-based proxy method to protect cloud data sharing through encryption. The data owner can use identity encryption to send the data to the cloud, and the agent can regrant the access rights of legal users. The blockchain-based system model is conducive to the decentralization of data sharing, relieving the pressure of big data processing in centralized systems, and is also conducive to the privacy protection of personal data.

Similarly, blockchain also has many applications in the DF of IoT. Kumar et al. [11] studied the issue of cross-border cloud forensics and proposed a blockchain customized IoT framework for DF, which is called Internet of Forensics (IoF). A transparent forensic investigation process was disclosed, taking into account the equipment involving multistakeholders. Existing digital forensics blockchain models tend to have weak security and less consideration for the privacy protection of stakeholders. Li et al. [12] conducted research on the legality of blockchain forensics. The research involves such links as evidence acquisition, evidence fixation, evidence analysis, and evidence presentation, as well as evidence supplementation and evidence circulation. The problem of weak security does not only appear in digital forensics but also in other blockchain systems. Li et al. [13] further study the security issues of blockchain. For each link in the blockchain system, the security risks and security solutions for the hidden risks are discussed separately.

This paper summarizes the research background and significance of IoTF. A blockchain-based IoT architecture is proposed, including an interface layer that can interact with applications. Then, the research background of blockchain technology is discussed, and an IoTF system based on an alliance chain is proposed. The previous research results of this paper were published in the ICAIS 2021 conference collection Advances in Artificial Intelligence and Security [14]. On this basis, we have an in-depth discussion of blockchain technology and further, propose an IoT forensics framework based on blockchain technology. By using the consortium chain idea, IoT terminal, IoT centralized devices, Regulatory department, Judicial department, and Insurance company are integrated into the forensics framework. For a more detailed explanation, we give application examples and flowcharts. Finally, we give a common example of IoTF, drone forensics, to help readers better understand our ideas.

The remaining part of the paper is organized as follows. Section 2 describes IoTF and DF. Section 3 is the research status of IoTF. Blockchain infrastructure and data structure are given in Section 4. The fifth part discusses the design of the blockchain-based IoTF system. Section 6 concludes the paper.

2. IoT Forensics and Digital Forensics

A table summarizing the acronym used in the paper is presented in Table 1.

2.1. What Is DF. At the first International Conference of Computer Investigation Experts (ICCIE) held in the United States in 1991, the concept of "computer evidence" was first put forward. Computer evidence is information stored in electronic form that can be identified, restored, extracted, saved, reported, and made into legal evidence.

The National People's Congress deliberated and approved the draft amendment to the Criminal Procedure Law in 2012, and the new Criminal Procedure Law was formally implemented in 2013. This is the first time that my country's law has included "electronic data" in the types of evidence. At the 28th meeting of the Standing Committee of the Eleventh National People's Congress on August 31 of the same year, it was decided to make the following amendments to the "Civil Procedure Law of the People's Republic of China," adding the type of evidence "electronic data," and it came into effect on January 1, 2015.

The Supreme People's Court issued the newly amended "Several Provisions of the Supreme People's Court on Evidence in Civil Litigation" in 2019, which will come into effect on May 1, 2020. Among them, the types of electronic data are detailed, including five types of various forms:

(1) Information published on web platforms, such as webpages, blogs, and microblogs.

(2) Application communication information of mobile phones, such as SMS, video communication, email, and so on.

(3) Electronic personal information of users, such as identity authentication information, electronic transaction information, communication records, etc.

Table 1: Table summarizing the acronym used in the paper.

Acronym	Explanation
MEC	Mobile edge computing
IoT	Internet of things
FoBI	Fog-based IoT forensics framework
ODL-CNN	Optimal deep learning-based convolutional neural network
IoF	Internet of forensics
ICCIE	International conference of computer investigation experts
DF	Digital forensics
USB	Universal serial bus
JPG	Joint picture group
MP3	Moving picture experts group audio layer-3
MP4	Mobile pentium 4
RFID	Radio frequency identification
EDFIM	Enhanced digital forensic investigation model
DFIM	Digital forensic investigation model
FSAC	Forensic state acquisition controller
FSIoT	Forensic status of the Internet of things
IoA	Internet of everything
PoW	Proof of work
PoS	Proof of stake
DPoS	Delegated proof of stake
PBFT	Practical byzantine fault tolerance
UAV	Unmanned aerial vehicle

(4) Electronic documents, pictures, films, and other electronic documents.

(5) Other information that can prove the facts of the case that is stored, processed, and transmitted in digital form.

Definition 1. Digital forensics is also called electronic data forensics, and its scope includes computer forensics, mobile phone forensics, network forensics, server forensics, etc. It is a process in which public security organs and judicial organs use computer-related technologies to identify, collect, fix, analyze, present, and preserve digital evidence extracted from electronic devices, thereby helping to reconstruct, reproduce, and prove criminal facts.

With the use of digital evidence in criminal law, civil law, and criminal procedure law more and more in-depth, the importance of digital forensics has gradually become prominent. In September 2016, the Supreme People's Court, the Supreme People's Procuratorate, and the Ministry of Public Security issued the "Regulations on Several Issues Concerning the Collection, Extraction, Review, and Judgment of Electronic Data in Criminal Cases" notice, which was officially implemented on October 1. The "Regulations" pointed out that Electronic data includes but is not limited to the following information and electronic files. The Regulations divide electronic data into four categories, basically following the aforementioned "Several Provisions of the Supreme People's Court on Evidence in Civil Litigation," which will not be repeated here.

In January 2019, the Ministry of Public Security issued a notice on the "Rules for Public Security Organs' Handling of Criminal Cases Electronic Data Collection Rules," which came into effect on February 1. The "Rules" point out that digital forensics includes but is not limited to the following:

(1) Collect and extract electronic data

(2) Electronic data inspection and investigation experiments

(3) Electronic data inspection and appraisal

Simply put, the digital forensics process is the process of converting digital evidence into a report form.

The main steps of the digital forensics process are shown in Figure 1.

By considering the concepts related to digital forensics, there are also computer forensics and IoT forensics. Comparing these three concepts, the concept of digital forensics has the largest category. Digital forensics mainly faces the forensics of digital devices. A computer, also known as a digital computer, is a typical digital device. Basically, IoT devices are also digital devices. Therefore, the scope of digital forensics includes computer forensics and IoTF.

2.2. IoT Forensics VS Traditional Digital Forensics. The IoT is designed as a network of intelligent, decision-making, and self-management systems, which has a great impact on DF. Because from the perspective of criminal liability caused by smart things in the IoT, IoT proposes many dimensions. These dimensions will affect the conventional practice of DF. IoT forensics may be different from traditional DF in the following aspects. Table 2 highlights these differences.

2.3. IoT Forensic. Internet of things forensics can collect, analyze, and find digital evidence of data in IoT devices on the premise of legal binding. However, IoT devices with limited cache capacity may be difficult to achieve these goals. Moreover, some devices that can only be connected locally cannot quickly transfer evidence to researchers. Finally, technology law enforcement agencies can confiscate computers, servers, and other equipment, but it is not so simple to set up the amount of Internet of things equipment required for a case investigation.

With the rapid development of Internet of things technology, data evidence extends from personal assets such as notebooks to broad Internet of things devices such as wearable devices. This brings new opportunities and challenges to researchers. Only some of the previous forensics methods and tools are available in the IoTF. There is an urgent need for new tools and regulations to innovate the forensics technology in the era of the IoT [15]. As IoT communication needs to be carried out under the support of protocols and standards, there are high requirements for equipment and evidence collection materials. In 2015, [16] first proposed the concept of the Internet of things forensics and improved Edewede Oriwoh's 1-2-3 regional method model. [15] discusses equipment level forensics, network forensics, and cloud forensics (see Figure 2).

2.4. Generalized IoT Forensics Model. This section proposes a generalized IoT forensics model which consists of three independent components: forensic scenarios, forensic

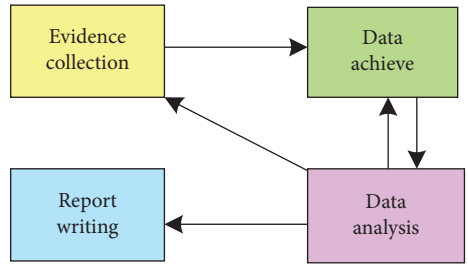


Figure 1: The main steps of the digital forensics process.

Table 2: IoT forensics VS traditional digital forensics.

	Traditional digital forensics	IoT forensics
Source of evidence	Traditional storage media such as computers, mobile phones, USB flash drives, cameras, and servers such as switches and routers	Smart terminals such as cars, drones, smartwatches, smart bracelets, smart sensors, smart industrial equipment, smart appliances, smart wearable devices
Equipment quantity	Tens of billions of magnitudes	Trillions of magnitude
Type of evidence	Electronic documents, standard format files (JPG, MP3, MP4, etc.)	Added a large number of nonstandard data files for IoT smart terminals
Evidence data size	Megabyte	Exabytes
Network type	Wired network, WIFI, Bluetooth, wireless network, internet, mobile communication network	Added RFID, wireless sensor network, Internet of things (Internet of vehicles, industrial Internet of things, etc.)
Protocol	Ethernet, wireless (802.11a/b/g/n), bluetooth, IPv4, IPv6, TCP/IP, etc.	RFID,TCP/IP,B/S和C/S,HTTP, Ajax,Websocket, MQTT,CoAP, etc.
Owner of the evidence (equipment)	Victims, suspects, related contacts	Anyone
Judicial	The relevant legislation is basically complete	The relevant legislation is not yet complete
Privacy	Infringement of citizens' privacy is less problematic	Legislation and borders are not clear, and privacy issues are involved

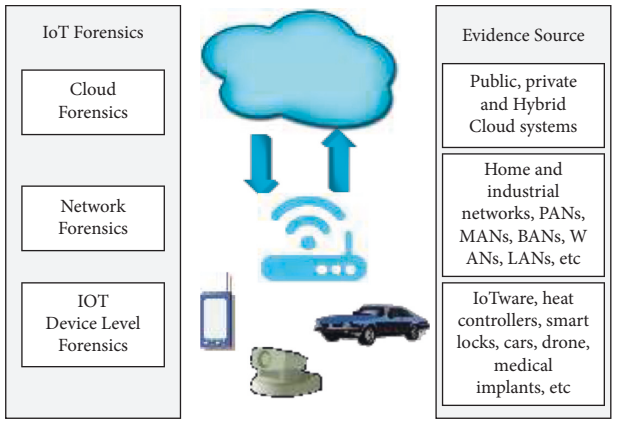


Figure 2: Three-tier IoT forensics model.

objects, and forensic processes (see Figure 3). The IoT forensics scene is very broad, and it can even be said to cover all aspects of our work and life. This confirms the importance of IoT forensics from another perspective. People will always leave traces in the IoT unknowingly, which will be an important starting point and breakthrough for digital forensics investigators. Although this will involve user privacy issues, we will discuss them in detail in the follow-up content.

Here, we use smart home, smart wear, industrial internet, and Internet of Vehicles as typical IoT forensics scenarios. All kinds of smart appliances in the smart home scene can be obtained for evidence, such as smart TVs, smart door locks, smart rice cookers, smart refrigerators, routers, and so on. Through the forensics of smart door locks, the information of the permanent population of the family can be obtained, and even the fingerprint information of the relevant personnel can be obtained directly. Through router forensics, you can obtain information such as the person and time of the wifi login user. Through smart TV forensics, you can obtain information such as family member composition, preferences, and living habits. The forensics of other smart appliances, such as smart rice cookers, smart refrigerators, etc., will also help the suspect's portrait to obtain clues to the case.

For forensics objects, we follow Edewede Oriwoh's model, which is divided into three levels: corresponding to the terminal forensics at the bottom, network forensics at the middle layer, and cloud forensics at the upper layer. The object of terminal forensics is the most extensive, and all IoT terminal devices are covered. Network forensics is an extension of terminal equipment forensics. The target is the network flow of all possible criminal computers, audit logs, and system logs. Cloud forensics refers to the collection of digital forensics data from cloud infrastructure. Terminal forensics and network forensics generally specifically refer to the collection of information from log files, data stored on disks, network traffic, and intrusion markers. The basic difference between terminal forensics and network forensics, and cloud forensics is that you can collect and analyze

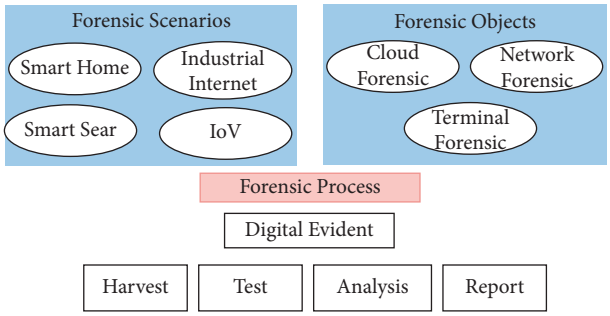


Figure 3: Generalized IoT forensics model.

information by simply entering the system using a local computer. However, when it comes to the cloud, the machine cannot be physically accessed; only certain parts of the computer can be accessed through the cloud application program interface. In addition, since cloud servers can be located in multiple countries, forensic data may also belong to multiple jurisdictions. The issue of jurisdiction cannot be ignored.

3. Research Status of Forensics in the Internet of Things

In 2013, Edewede Oriwoh et al. first proposed the concept of IoT forensics [15]. A 1-2-3 area method is proposed to be applied to DF research related to the IoT, which is the earliest IoT forensics model. After continuous improvement, in 2015, Shams Zawoad gave the definition of IoTF firstly [16]. Extend the DF to the category of IoT, study the DF process of IoT devices, and give an accurate definition of IoTF.

Aiming at the privacy issue of IoT forensics, Ana Nieto et al. conducted pioneering research. In 2016, Ana Nieto et al. published a long journal article on "digital witnesses" [17], which was the first journal article on forensics research on the Internet of Things. This article first proposed the concept of "digital witness" and gave its formal definition, discussed the new concept in personal devices, and further defined the basic components for realizing this concept in future work. In 2017, Ana Nieto and others analyzed the enhanced digital forensic investigation model (EDFIM). By including the privacy protection requirements of the 1974 "US Privacy Act" and ISO/IEC 29100:2011 [18] in the entire investigation life cycle, a privacy-aware IoT forensics model (PROFIT) [19] is proposed.

In 2017, literature [20] discussed the key issues of IoT forensics from the perspective of IoT security. First, starting with the basic elements of IoT, it discussed the three-tier framework of IoT and the key issues of IoT forensics. Then it further reviewed the research and development of the forensic model of the IoT in recent years. Literature [21] considers the heterogeneity of devices in the IoT system and the lack of uniform standards. By taking forensics in three representative IoT application scenarios, smart homes, wearable devices, and smart cities, as examples, a digital forensic investigation model (DFIM) for specific applications in the IoT is proposed. The DFIM model can collect, inspect, analyze, and report reasonable forensic evidence in a dedicated DF investigation of the IoT. Literature [22] proposed the definition of forensic state acquisition controller (FSAC) in response to problems such as the nonstandardization of IoT devices and lack of connectivity. It further proposes a general framework and a method for obtaining the forensic status of the Internet of Things (FSIoT).

In 2018, Maxim Chernyshev et al. published the first journal literature review on IoT forensics [23]. The author briefly reviewed the development of digital forensics models in the IoT environment and further discussed the open problems that exist when these digital forensics technologies are applied to Internet of Things devices. Literature [24] proposes a forensic investigation framework that uses public digital ledgers to find criminal facts based on IoT systems. By collecting the interactions occurring between various IoT entities as evidence, it securely stores them in public, distributed, and decentralized blockchain networks. Literature [25] studies the mobility in the Internet of Things at crime scenes and discusses data identification and classification methods from the Internet of Things to find the best evidence. The tools and techniques for identifying and locating IoT devices are proposed. Based on the frequency and interaction mapping between devices, the recent concept of "digital footprint" was developed in the criminal field. Literature [26] proposed a blockchain-based IoT Forensics Framework (BIFF) for IoT security issues, which records events throughout the life cycle of digital evidence in a transparent, traceable, and identity privacy protection manner. Literature [27] discussed the complexity of forensics brought about by the Internet of Everything (IoE) era and further, analyzed the actual digital forensics process and the challenges that arise, and even the difficulties of the IoT forensics standards. Literature [28] studies new security issues from the perspective of cloud forensics, which mainly focuses on solving the security risks caused by customer data after customers stop using cloud services. A framework is proposed to solve the security problem of reconstructing customer data after using cloud services to delete or stop customer data.

In 2019, Francesco Servida et al. published an overview on the digital traces of IoT devices [29]. The author considers that the massive increase in IoT devices lacks existing digital forensics tools and methods and the corresponding security and privacy issues. Aiming at the application of IoT in the field of smart homes, the opportunities and challenges of IoT forensics are discussed. Literature [30] considers the security issues of cloud forensics under fog computing and points out that archiving network traffic will become the basis for key tasks such as fog computing forensics, monitoring, and troubleshooting. A new system architecture is further proposed to subtly bridge trusted hardware and searchable encryption to build a trusted, encrypted but queryable network traffic file for fog-assisted IoT applications. Literature [31] proposed a forensic analysis model. This model can acquire and analyze various Internet of things devices and serve forensic work. Taking forensic artifacts retrieved

by the popular Amazon Echo as an example, the author demonstrates how to use the proposed model to guide the forensic analysis process of IoT devices. Literature [32] proposes an automatic knowledge-sharing forensics platform, which can automatically suggest a forensic mode from case data.

In 2020, Jianwei Hou et al. published a review of forensics on the Internet of Things in the top international journal IEEE Internet of Things Journal, giving a comprehensive overview of IoTF [33]. Stoyanova et al. published a review of the Internet of Things forensics in the top international journal IEEE Communications Surveys & Tutorials [34]. The emergence of these two top journal review papers means that the academic community is paying more and more attention to IoTF. They systematically review the development of IoTF in the past 10 years and summarize the classic forensic models and forensic methods. They discuss in detail the key issues that have been resolved and unresolved in the forensics process, especially the applicability of technology and legal boundary issues, as well as data security and privacy protection issues that will be faced in the future, so as to point out scientific research directions for latecomers.

4. Blockchain Technology

4.1. The Development of Blockchain. In 2008, Nakamoto [35] creatively proposed the framework of blockchain technology and proposed an idea of using Bitcoin as a decentralized digital currency. Soon, Bitcoin theory was put into practice, and this digital currency system without third-party guarantees came into being. In 2013, Buterin [36] inherited and developed the Bitcoin system, proposed the concept of Ethereum, and integrated the programmable features of smart contracts. Accordingly, the combination of the decentralization of blockchain and the programmable features of smart contracts has enabled the rapid development of the next-generation digital currency system, and a large number of virtual digital currencies such as Tether and Dogecoin have emerged. The application of blockchain also covers all aspects of people's lives. As shown in Figure 4, the blockchain architecture is a combination of a series of decentralization, trusted computing, and privacy protection algorithms.

4.2. Blockchain Data Structure. Blockchain is a distributed system, and its block structure determines the storage form of transaction information. The Merkle tree of chain structure is used to organize and manage transaction data which plays the function of connecting blocks. The data structure of blockchain transactions describes the transaction forms of Bitcoin and Ethereum and the characteristics of the generation of transaction addresses. The storage method of transaction data analyzes the design basis and development trend of the underlying data structure of the blockchain from a macroperspective. The blockchain data block structure is shown in Figure 5

4.3. Blockchain Chain Structure. Blockchain is a distributed database that links each block in order of generation time. As seen in Figure 5, the Prev-block Hash field is used to store the hash value of the previous block. All blocks are linked together in the order of generation with the Prev-block Hash field as the hash pointer. The chain structure of the above blocks forms a blockchain list, that is, a complete ledger.

The relationship between adjacent blocks in the blockchain structure is shown in Figure 6. According to the "Merkle-root" field and "Prev-block Hash" field in the block header, it can be verified by hash operation whether it has been tampered with. Relying on the Prev-block Hash field, all blocks are linked according to the creation time. If any of the blocks is tampered with, it will cause the hash value of all blocks generated afterward to change in a chain. Using the verifiability feature of the chain structure, when a node downloads certain blocks or the entire block from an untrusted node, the correctness of each block can be verified through a hash operation.

4.4. Blockchain Consensus Mechanism. The consensus mechanism is the core of decentralized trust in distributed systems. It establishes a set of mutually untrusted preset rules to realize the cooperation between nodes, which finally achieves the consistency of the data of different nodes.

Blockchain is essentially a distributed ledger record database. Therefore, the consensus mechanism in the blockchain must not only reflect the basic requirements of a distributed system but also consider the security issues of the blockchain, specifically for transaction records, the need to solve Byzantine fault tolerance, and possible malicious nodes to tamper with data. In general, the consensus mechanism in the blockchain is more targeted, and the consensus mechanism that meets different operational requirements can be selected according to different blockchain application scenarios.

Since Lamport et al. [37] put forward the "Byzantine Generals Problem" in 1982, a large amount of research on consensus algorithms has focused on theoretical discussions. But since Bitcoin entered people's sight in 2008, various consensus mechanisms have begun to move from theory to practice. With the iteration of Bitcoin itself and the development of the Ethereum platform, blockchain-based applications such as smart contracts and hyperledgers are becoming more and more abundant. Existing consensus algorithms have been improved in practice. At the same time, with the continuous emergence of new application scenarios, consensus mechanisms that meet corresponding requirements have been applied one after another.

Next, we talk about the current representative consensus mechanism in the blockchain. Table 3 compares the main characteristics of each algorithm.

5. Design of IoT Forensics System Based on Blockchain

5.1. Alliance Structure. The overall architecture is shown in Figure 7, including the client side and the server side.

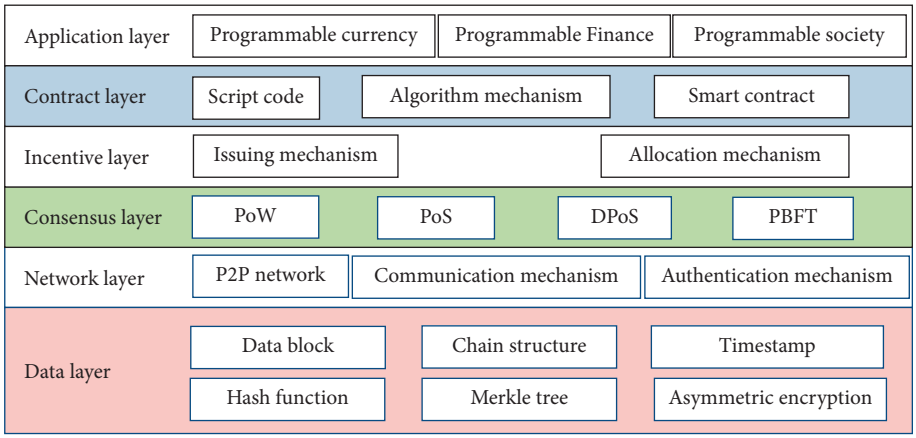


Figure 4: Blockchain infrastructure.

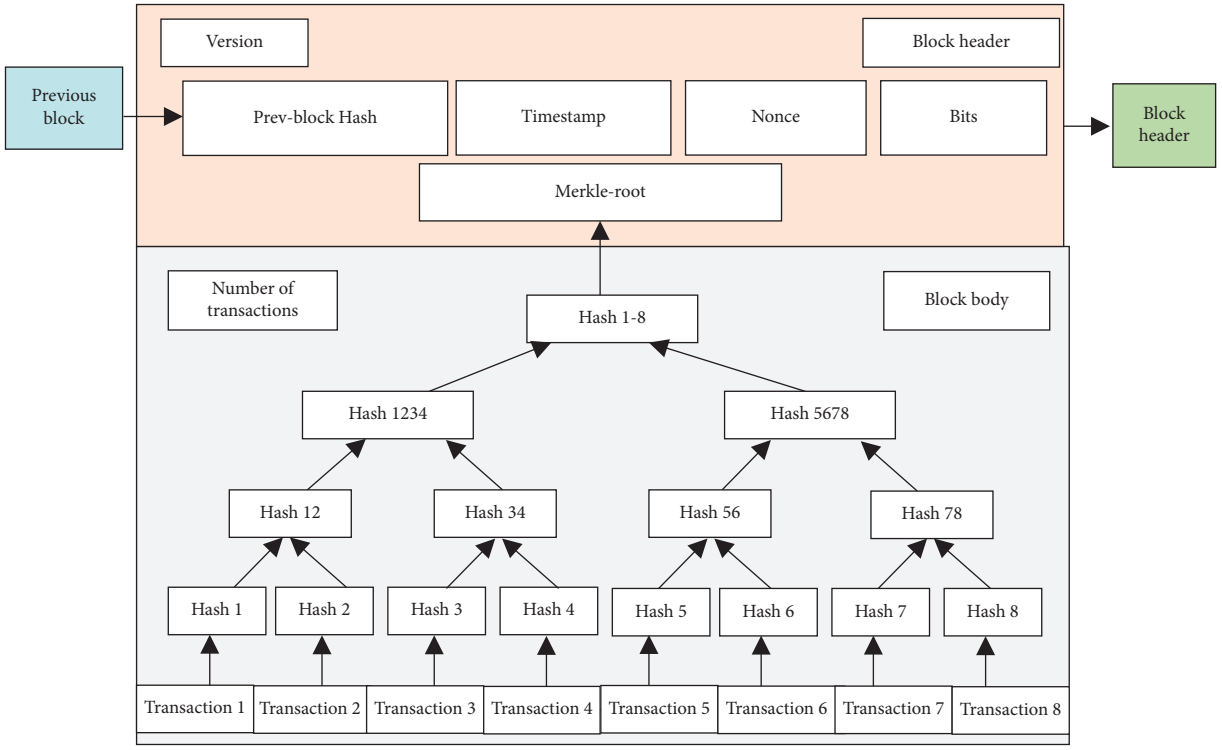


Figure 5: Blockchain data block structure.

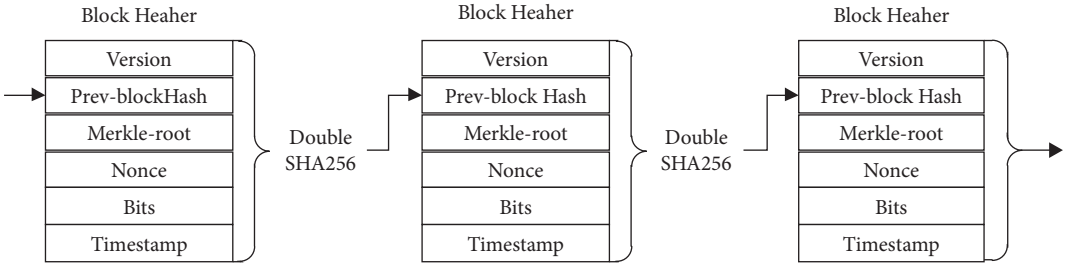


Figure 6: The relationship between adjacent blocks.

Table 3: Comparison of PoW, PoS, DpoS, and PBFT consensus mechanisms.

Parameter	PoW [38]	PoS [39]	DPoS [40]	PBFT [41]
Degree of centralization	Fully decentralized	Fully decentralized	Partially decentralized	Partially decentralized
Node access license	Not needed	Not needed	Not needed	Needed
Number of access nodes	Unlimited	Unlimited	Unlimited	Limited
Block time	Longer	Longer	Shorter	Shorter
Main resource occupation	Computing power	Equity, token	Equity, token	Bandwidth
Application scenario	Public chain	Public chain	Public chain	Alliance chain
Whether to fork	Easy to fork	Easy to fork	Not easy to fork	No fork
Final consistency	No finality	No finality	No finality	Finality
Security guarantee	More than 1/2 of computing power is credible	More than 1/2 stake is credible	More than 1/2 of equity is credible	More than 2/3 of nodes are trusted

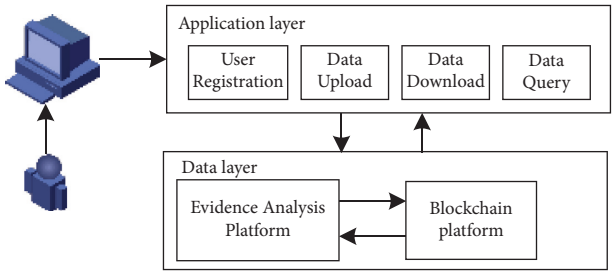


Figure 7: Blockchain-based IoT forensics overall architecture.

The overall architecture of blockchain-based IoT forensics is shown in Figure 8. In the Internet of Things environment, IoT terminals, and IoT convergence devices, regulatory agencies, judicial departments, and insurance companies form a consortium chain. Among them, IoT terminals, judicial departments, and insurance companies are light nodes, and each block header information is stored. All IoT convergence devices and regulatory departments in the jurisdiction are full nodes that are responsible for full chain storage and new block entry into the chain.

(1) IoT terminal: includes all IoT-based terminal devices, which are the most primitive generators of massive data. Equipment manufacturers regularly upload the generated data to the cloud through industry standards or corporate standards and finally upload the data to the chain through the alliance chain architecture.

(2) IoT centralized devices: usually smart switches, smart routers, and other devices with data concentration functions. These centralized devices are responsible for packaging and verifying the first-hand data, and its importance is self-evident. Relevant industry standards and regulatory measures must be promulgated first. In most cases, some cheap IoT device manufacturers have not built an enterprise cloud and will not upload data. Instead, they choose to upload data to IoT centralized devices on a regular basis.

(3) Regulatory department: it is composed of IoT enterprise representatives and industry alliances. It is mainly responsible for formulating industry standards, building an IoT forensics architecture based on the alliance chain, and guiding IoT companies to standardize data upload. When a dispute occurs or a case requires forensics and appraisal, IoT forensics shall be collected, and an appraisal report shall be issued.

(4) Judicial department: it is composed of law enforcement agencies (police and court), which can inquire and analyze the evidence stored in disputed entities in the chain and make judgments on liability and provide evidence to the insurance company to facilitate the insurance company to pay compensation.

(5) Insurance company: inquire about evidence or accept relevant evidence provided by the judicial department to decide the compensation plan alliance chain.

The IoT forensics system using the above alliance chain architecture can ensure that data is stored and obtained as safely and reliably as possible. On the one hand, the data from the Internet of Things can be directly connected to the chain, or it can be connected to the chain through an IoT centralized device. On the other hand, the supervisory departments on the chain give full play to their supervisory advantages to guide judicial departments and insurance companies to process data on the chain fairly and impartially.

5.2. IoT Forensics Model Based on Alliance Chain. Based on the overall framework of IoT forensics proposed in the previous section, it is further refined, and an IoT forensics model based on the alliance chain is proposed. As shown in Figure 9, the IoT forensics model includes 6 main modules:

(1) Data provider: it specifically refers to IoT terminal equipment. Through the device that senses and collects surrounding environment data, it sends the data upload transaction form, and after identity authentication, it provides data to the alliance chain. The data is stored in the distributed storage system in ciphertext form, and the data summary is stored in the blockchain network.

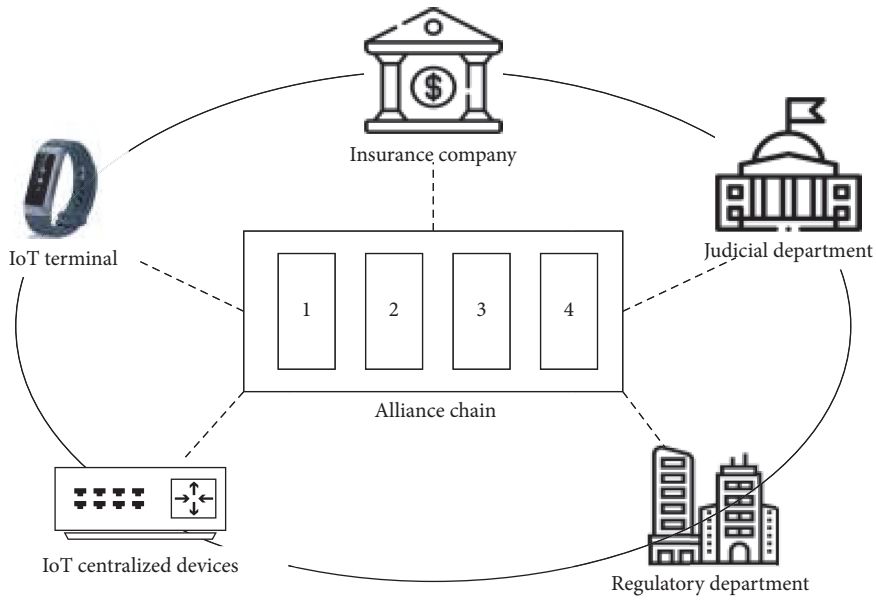


Figure 8: IoT forensics alliance architecture.

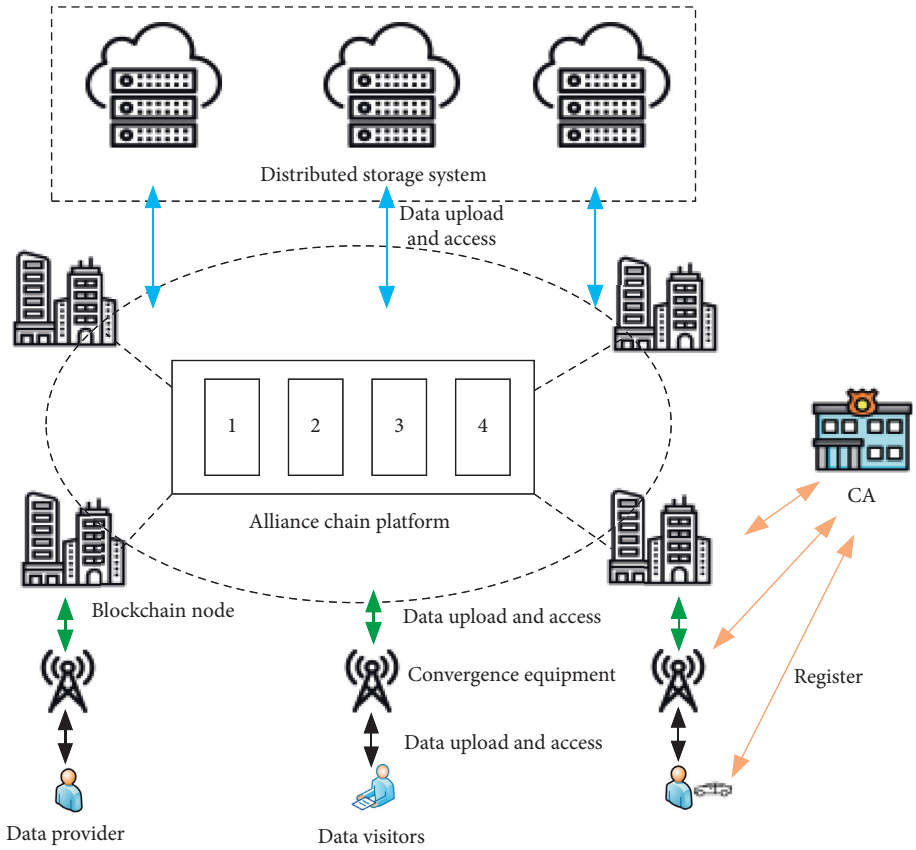


Figure 9: IoT forensics model based on alliance chain.

(2) Data visitors: they mainly include ordinary users, judicial departments, regulatory authorities, and insurance companies. Request corresponding data from the alliance chain by sending data access transactions. After identity authentication and access authority verification, the required data is obtained from the alliance chain.

(3) Convergence equipment: it is responsible for receiving data from the data provider and verifying the identity of the data requester and the integrity of the data packet.

(4) Alliance chain platform: iIt is built and maintained by a group of blockchain nodes to record data upload and data access in the consortium chain. The members of the alliance chain include IoT terminals, IoT convergence devices, regulatory agencies, judicial departments, and insurance companies. The purpose is to solve the problem of protection of the integrity and verifiability of evidence in the platform.

(5) Distributed storage system: it is used to store encrypted data packets in the blockchain network.

(6) CA: iInitialize the entire IoT alliance blockchain network, register each entity, and then keep it offline.

5.3. Data Provider's Perspective-Evidence Flowchart. This section discusses the IoT storage process based on the alliance chain from the perspective of the data provider, as shown in Figure 10.

(1) Login/Registration: before completing the deposit, the data provider (the deposit certificate user node) needs to complete the login/registration procedure with the CA to confirm the identity information and corresponding permissions. At the same time, the performance information of other user nodes is obtained, and reference data is obtained for subsequent data fragment storage.

(2) Electronic data fragmentation: the system uses a redundant fragmentation algorithm to fragment the uploaded electronic data And then select a number of nodes with the best performance based on the node performance information obtained previously to store the fragmented data of the system.

(3) Upload certificate files: users upload the files that need to be certificated and write the key information of the files into the contract file category. In this way, the corresponding mapping relationship between users and data is established.

(4) Verification of deposit documents: the system reads the relevant storage information of electronic data from the contract, obtain the location of the electronic data storage by storing the information and downloading the electronic data fragments, and restore the data and compare the file hash value to verify the integrity of the electronic data.

(5) Offline: the user logs out and ends this deposit operation.

5.4. Data Visitor's Perspective-Evidence Flowchart. This section discusses the IoT forensics process based on the alliance chain from the perspective of data visitors, as shown in Figure 11.

(1) Preparation: the forensics personnel needs to sort out the briefcase of the case before performing the verification operation, confirm the specific IoT forensics scenarios, and evaluate the potential forensic objects on-site and the evidence that needs to be collected.

(2) Initialization: the main event detection, first response, and investigation preparation aspects of the forensic initialization work. It is mainly to respond to the on-site evidence collection environment in a timely manner, and it is best to obtain on-site evidence as quickly as possible.

(3) Investigation: during the investigation phase, forensics personnel obtains, tests, analyzes, and screens evidence and tries to reconstruct the incident using the obtained evidence. Based on the refactored matter, the question of investigation and evidence collection is reversed, and relevant evidence for evidence collection is supplemented.

(4) On-chain interaction: in on-site investigation and evidence collection, through hash calculations and electronic signatures, the first-hand forensic data can be stored on the chain for the first time through the network, thereby curing it into data that cannot be tampered with. At the same time, it is also necessary to call the existing information on the chain for verification when collecting evidence on the spot, and the use of the alliance chain will be more flexible.

(5) Report: at the final stage of the forensic collection, a forensic report is issued. Through feedback from relevant units, confirm whether to return to the investigation stage to supplement the secondary evidence collection work.

5.5. Examples of IoT Forensics. The IoT has been integrated into every aspect of our lives, which makes the examples of IoT forensics everywhere. With the innovation of cloud computing, big data, artificial intelligence, and other technical means, UAV has led the current trend of smart consumption due to their high technological content and fashion sense. The UAV has more and more applications in daily consumption scenarios such as entertainment and life, which makes the demand for drone forensics more and more urgent.

This section takes DJI Mavic 2 pro as the experimental object to extract and analyze its data and try to give everyone a preliminary understanding of IoT forensics. The Mavic 2 pro series have a built-in SD card and are assisted by the APP installed on the mobile phone, which makes the mobile phone retain a large amount of original data of the UAV. This experiment adopts the mobile phone APP forensics method, which is also one of the most mature methods of digital data forensics.

As shown in Figure 12, the red curve represents the flight path of the UAV, and the blue curve represents the movement path of the UAV controller. In addition, the detouring and stagnant behavior during UAV flights can also be reflected by repeating the lines and adding groups. In

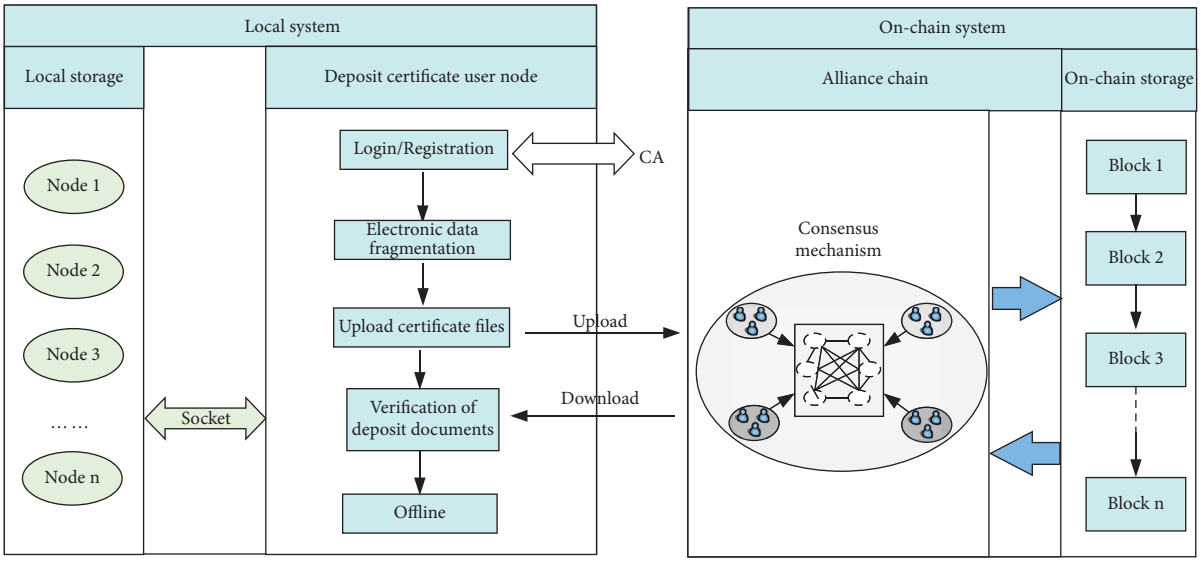


Figure 10: Data Provider's perspective-evidence flow chart.

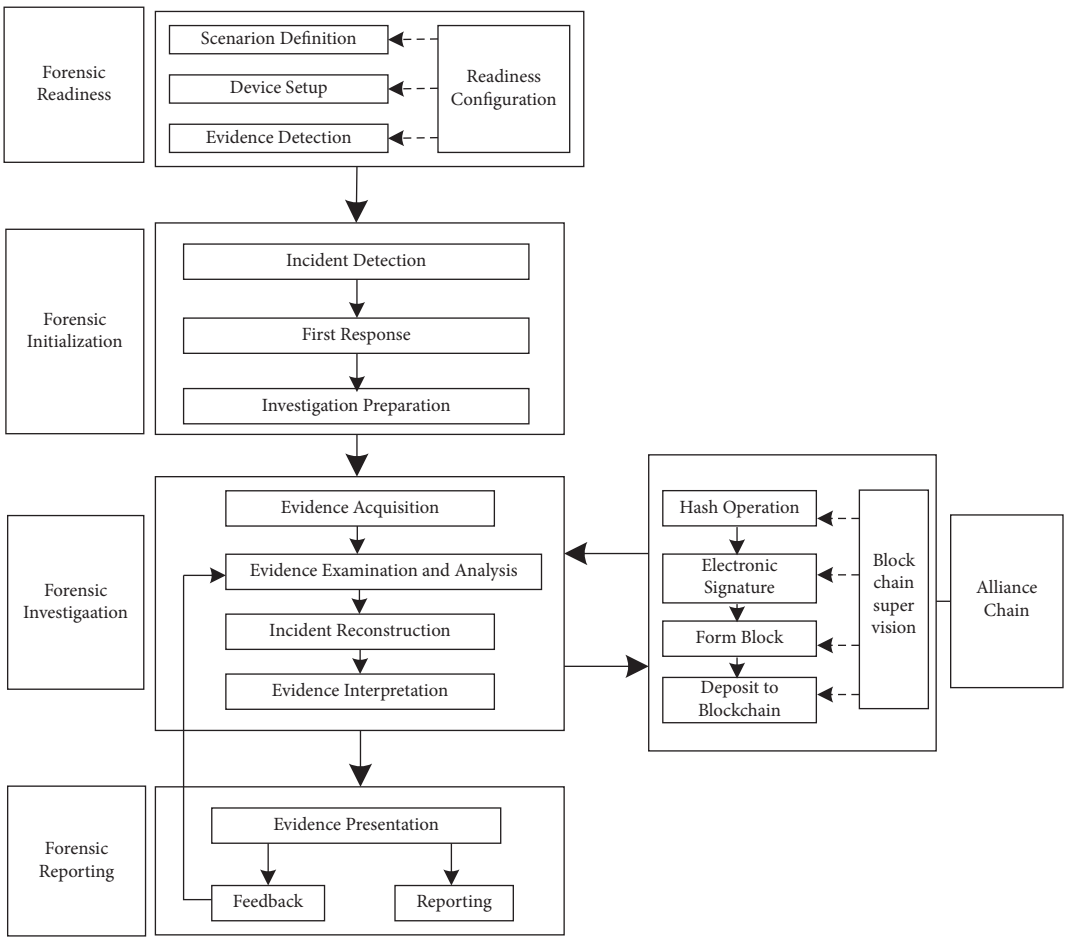


Figure 11: Data visitor's perspective-evidence flowchart.

addition to analyzing the flight path of the UAV, the photos and videos taken by the UAV are also important evidence, which involves traditional digital forensics, and we do not do much research.

Another important feature of the UAV is the lack of battery life, so the power management of the UAV is an important research direction. As seen in Figure 13, the voltage of the UAV and the remaining battery power can be analyzed. We found

Figure 12: UAV flight path.

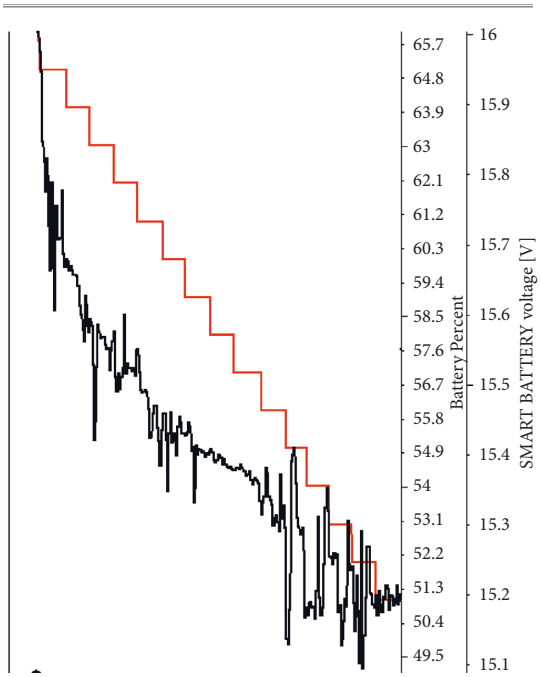


Figure 13: UAV voltage and remaining battery power.

that the battery power of the drone is 65% (red line) during flight, and as the power usage of the drone declines in steps, the battery voltage also peaks at around 16V due to the drone taking off. Then the battery voltage drops slowly and fluctuates slightly. By analyzing the battery and charge of the UAV, it is possible to confirm the behavior of the UAV at the time, which is valuable for further forensic analysis.

6. Conclusions

This article conducts research on IoT forensics and compares its differences with traditional DF. We further sort out the research results of IoT forensics in recent years. Through the research of blockchain technology, it is introduced into the IoT forensics framework. An IoT storage and forensics system based on the alliance chain is proposed. Subsequent research will consider privacy issues in the forensics process.

Data Availability

The experimental data used t support the findings of the study can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

An earlier version of this research has been presented as a conference publication at International Conference on Artificial Intelligence and Security (ICAIS) [14]. It has been supported by the National Natural Science Foundation of China (Grant no. 61802155), The Major Project Contract for Natural Science Research of Jiangsu Colleges and Universities (Grant no. 20KJA520004), Open project of the National and Local Joint Engineering Laboratory of Radio Frequency Integration and Micro-assembly Technology (Grant no. KFJJ20200201), 2021 Jiangsu Police Officer Academy Scientific Research Project: Research on D2D Cache Network Resource Optimization Based on Edge Computing Technology (2021SJYZK01), and High-Llevel Introduction of Talent Scientific Research Start-up Fund of Jiangsu Police Institute (JSPI19GKZL407).

References

[1] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for internet of Vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[2] X. Liang and Y. Kim, "A Survey on Security Attacks and Solutions in the IoT Network," in *Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference, (CCWC)*, pp. 0853–0859, NV, USA, June 2021.

[3] W. Zhou, "Reviewing IoT security via logic bugs in IoT platforms and systems," *IEEE Internet of Things Journal*, vol. 8, 2021.

[4] N. Miloslavskaya and A. Tolstoy, "Internet of Things: information security challenges and solutions," *Cluster Computing*, vol. 22, no. 1, pp. 103–119, 2019.

[5] E. Al-Masri, Y. Bai, and J. Li, "A Fog-Based Digital Forensics Investigation Framework for IoT Systems," in *Proceedings of the 2018 IEEE International Conference on Smart Cloud (SmartCloud)*, pp. 196–201, New York, NY, USA, September 2018.

[6] V. R. Silvarajoo, S. Yun Lim, and P. Daud, "Digital evidence case management tool for collaborative digital forensics investigation," in *Proceedings of the 2021 Digital Evidence Case Management Tool for Collaborative Digital Forensics Investigation*, pp. 1–4, CRC, Langkawi Island, Malaysia, January 2021.

[7] M. Elhoseny, M. M. Selim, and K. Shankar, "Optimal deep learning based convolution neural network for digital

forensics face sketch synthesis in internet of things (IoT)," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 11, pp. 3249–3260, 2020.

[8] Z. Su, H. Wang, H. Wang, and X. Shi, "A Financial Data Security Sharing Solution Based on Blockchain Technology and Proxy Re-encryption Technology," in *Proceedings of the 2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, pp. 462–465, Chongqing City, China, November 2020.

[9] D. Sathya, S. Nithyaroopa, D. Jagadeesan, and I. J. Jacob, "Block-chain technology for food supply chains," in *Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 212–219, Tirunelveli, India, February 2021.

[10] K. O.-B. O. Agyekum, Q. Xia, E. B. Sifah, C. N. A. Cobblah, H. Xia, and J. Gao, "A proxy Re-encryption approach to secure data sharing in the internet of things based on blockchain," *IEEE Systems Journal*, vol. 16, no. 1, pp. 1685–1696, 2022.

[11] G. Kumar, R. Saha, C. Lal, and M. Conti, "Internet-of-Forensic (IoF): a blockchain based digital forensics framework for IoT applications," *Future Generation Computer Systems*, vol. 120, no. 120, pp. 13–25, 2021.

[12] M. Li, C. Lal, M. Conti, and D. Hu, "LEChain: a blockchain-based lawful evidence management scheme for digital forensics," *Future Generation Computer Systems*, vol. 115, no. 6, pp. 406–420, 2021.

[13] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020.

[14] G. Liang, J. Xin, Q. Wang, X. Ni, and X. Guo, "A Blockchain-Based Internet of Things Forensics Model," in *Advances in Artificial Intelligence and Security*, pp. 687–696, Springer, New York, NY, USA, 2021.

[15] E. Oriwoh, D. Jazani, G. Epiphaniou, and P. Sant, "Internet of things forensics: challenges and approaches," in *Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp. 608–615, Austin, TX, USA, June 2013.

[16] S. Zawoad and R. Hasan, "FAIoT: towards building a forensics aware eco system for the internet of things," in *Proceedings of the 2015 IEEE International Conference on Services Computing*, pp. 279–284, New York City, NY, USA, June 2015.

[17] A. Nieto, R. Roman, and J. Lopez, "Digital witness: safeguarding digital evidence by using secure architectures in personal devices," *IEEE Network*, vol. 30, no. 6, pp. 34–41, 2016.

[18] Iso, "Information technology - Security techniques - Privacy framework," 2020, https://www.iso.org/standard/73722.html.

[19] A. Nieto, R. Rios, and J. Lopez, "A methodology for privacy-aware IoT-forensics," in *Proceedings of the 16th IEEE International Conference on Trust, Security And Privacy in Computing and Communications*, pp. 626–633, Sydney, NSW, Australia, October, 2017.

[20] M. Banday, "Enhancing the security of IOT in forensics," in *Proceedings of the International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, Gurgaon, India, October 2017.

[21] T. Zia, P. Liu, and W. Han, "Application-specific digital forensics investigative model in internet of things (IoT)," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, New York; NY, USA, September 2017.

[22] C. Meffert, D. Clark, I. Baggili, and F. Breitinger, "Forensic state acquisition from internet of things (FSAIoT)," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ACM ,Calabria, Italy, August 2017.

[23] M. Chernyshev, S. Zeadally, Z. Baig, and A. Woodward, "Internet of things forensics: the need, process models, and open issues," *IT Professional*, vol. 20, no. 3, pp. 40–49, 2018.

[24] M. Hossain, R. Hasan, and S. Zawoad, "Probe-IoT: a public digital ledger based forensic investigation framework for IoT," in *Proceedings of the IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, Paris, France, June 2018.

[25] F. Bouchaud, G. Grimaud, and T. Vantroys, "IoT forensic," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, IEEE, Hamburg, Germany, August 2018.

[26] D. P. Le, H. Meng, L. Su, S. L. Yeo, and V. Thing, "BIFF: a blockchain-based IoT forensics framework with identity privacy," in *Proceedings of the TENCON 2018 - 2018 IEEE Region 10 Conference*, Jeju, Korea, October 2018.

[27] A. Macdermott, T. Baker, and Q. Shi, "Iot forensics: challenges for the ioa era," in *Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, IEEE, Paris- France, February, 2018.

[28] J. Surbiryala and C. Rong, "Secure customer data over cloud forensic reconstruction," in *Proceedings of the 2018 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, January 2018.

[29] F. Servida and E. Casey, "IoT forensic challenges and opportunities for digital traces," *Digital Investigation*, vol. 28, pp. S22–S29, 2019.

[30] H. Duan, Y. Zheng, C. Wang, and X. Yuan, "Treasure collection on foggy islands: building secure network archives for internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2637–2650, 2019.

[31] S. Li, K.-K. R. Choo, Q. Sun, W. J. Buchanan, and J. Cao, "IoT forensics: Amazon Echo as a use case," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6487–6497, 2019.

[32] X. Zhang, K. K. R. Choo, and N. L. Beebe, "How do I share my IoT forensic experience with the broader community? An automated knowledge sharing IoT forensic platform," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6850–6861, 2019.

[33] J. Hou, Y. Li, J. Yu, and W. Shi, "A survey on digital forensics in internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 1–15, 2020.

[34] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches and Open Issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.

[35] S. Nakamoto, "Bitcoin:A peer-to-peer electronic cash system," 2009, https://bitcoin.org/bitcoin.pdf.

[36] V. Buterin, "next-generation smart contract and decentralized application platform(white paper)," 2019, https://github.com/ethereum/wiki/wiki/White-Paper.

[37] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.

[38] B. Adam, "The Hashcash Proof-Of-Work function(draft)," 2003, http://www.hashcash.org/papers/draft-hashcash.txt.
[39] D. Larimer, "Transactions as Proof-Of-Stake," 2013, http://7fvhfe.com1.z0.glb.clouddn.com/wp-content/uploads/2014/01/TransactionsAsProofOfStake10.pdf.
[40] A. Bisola, "Delegated Proof-Of-Stake (DPoS) explained," 2018, https://www.mycryptopedia.com/delegated-proof-stake-dpos-explained/.
[41] J. Fan, L.-T. Yi, and J.-W. Shu, "Research on the technologies of byzantine system," *Journal of Software*, vol. 24, no. 6, pp. 1346–1360, 2014.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 1557233, 9 pages
https://doi.org/10.1155/2022/1557233

WILEY | Hindawi

Research Article

An Improved Secure Public Cloud Auditing Scheme in Edge Computing

Zhengge Yi, Lixian Wei, Haibin Yang, Xu An Wang, Wenyong Yuan, and Ruifeng Li

Engineering University of PAP, Xi'an, China

Correspondence should be addressed to Xu An Wang; wangxazjd@163.com

Received 4 November 2021; Revised 24 January 2022; Accepted 28 February 2022; Published 26 April 2022

Academic Editor: Xiaolong Xu

Cloud storage plays an important role in the data processing of edge computing. It is very necessary to protect the integrity of these data and the privacy of users. Recently, a cloud auditing scheme which can be used to smart cities has been proposed, which is lightweight and privacy-preserving. Although this scheme has very good performance and is a very valuable work, we find that there is insecurity in it. By giving two kinds of attacks, we prove that a malicious cloud server provider (CSP) can forge auditing proof and can successfully pass the verification of the third-party auditor (TPA) even if the CSP deletes the user's data. Then, based on this scheme, we propose an improved scheme, which can resist the forgery attack from malicious CSP. Through security analysis, our scheme improves the security compared to the original scheme without reducing the efficiency.

1. Introduction

The rise of the Internet of Things and the 5G network has led to many new services, including intelligent transportation, smart city, location service, and so on [1, 2]. The number of smartphones, wearable devices, Internet-connected televisions, and other sensor devices shows an explosive growth trend, followed by "sea-scale" data generated by these Internet of Things terminals [3–6].

In edge computing, some or all of the private data of end users need to be outsourced to third parties (such as cloud computing data centers and edge data centers) [7–9]. By using the cheap storage and computing services provided by the cloud server, users with limited resources can be freed from the complex hardware system, reduce the storage burden, and at the same time be able to easily access their own data [10–12]. Compared to the traditional cloud computing model which relies solely on the computing center, the edge computing can handle the big data at the network edge effectively.

However, the users' data stored in the third-party data center have the features of separation of control, storage randomization, and so on, which can easily lead to data security problems such as data loss, data leakage, and so on [13, 14]. When the integrity of users' data is destroyed, the interests of these users may receive huge losses. Therefore, it is significant to design a cloud auditing scheme for edge computing.

1.1. Related Work. Recently, in order to meet different application requirements, various cloud storage audit schemes have been proposed. At present, the research on data integrity audit is mainly focused on four functional requirements, namely, dynamic audit, batch audit, privacy protection, and lightweight computing.

At the CCS conference in 2007, Jules and Ateniese et al. proposed proofs of retrievability (POR) and provable data possession (PDP), respectively, to audit cloud storage data [15, 16]. Both of them use the idea of sampling testing to audit the integrity of the data. That is, only a small part of the data in the cloud can ensure the integrity and reliability of all data with a high probability. Then, Ateniese et al. proposed a scalable PDP scheme based on the original PDP [17], which is the first verifiable data holding protocol that supports partial dynamic operation. The design of this protocol provides a new idea for the construction of the cloud audit protocol and takes an important step towards the more

practical PDP protocol. Inspired by Ateniese et al., Erway et al. [18] extended the above PDP protocol and designed a protocol that supports the dynamic update of cloud data. The audit protocol uses jump tables to support complete dynamic operation of data. Compared with the protocol of Ateniese et al., it has a greater breakthrough in practical value and the probability of detecting cloud data errors. However, the protocol does not have the performance of privacy protection, batch auditing, and so on.

Wang et al. [19] proposed a distributed data audit system to protect privacy in order to solve the problems of privacy disclosure and batch audit in the process of data integrity audit. The system uses a third-party audit platform to perform integrity audits, and the data owner can delete the local original data after the data are outsourced and stored to the cloud server. At the same time, homomorphic MAC and random mask technology are used to ensure that the third-party audit platform cannot know the content of the stored data in the effective audit process to achieve privacy protection. Subsequently, Wang et al. further improved the scheme in reference [20] by constructing a Merkle hash tree structure based on block authentication tags to improve the proof of the storage model. A study [20] further improved the bilinear aggregation signature method and improved the batch audit efficiency of TPA. Yang et al. [21] proposed an efficient and privacy-protected dynamic auditing protocol, which can be extended to realize dynamic data operations and batch auditing. At the same time, combining cryptography and bilinear properties, this scheme can protect the data privacy. In view of mobile devices with insufficient computing power, a lightweight integrity audit scheme supporting privacy protection is proposed in reference [22]. This scheme uses an online/offline signature method where the offline phase undertakes a lot of computing work. When the data file to be outsourced is given, the user just needs to construct the outsourced data signature in the online phase, which is lightweight.

1.2. Motivation. At present, in most public audit systems, in order to ensure the integrity of user data, the third-party auditor usually initiates an integrity challenge to the CSP, and then the CSP generates evidence to prove that it honestly stores user data. In this model, we first need to ensure that the cloud service provider cannot complete the forgery attack; that is, the forged evidence cannot be verified by the third-party auditor.

Recently, a public cloud auditing scheme has been proposed by Jing Han et al. [23]. This scheme is pairing-free and allows a third-party auditor to generate authentication metasets on behalf of users, which can achieve lightweight computing. It can protect the privacy of a user's data by blinding the raw data before storing them in the CSP and sending to the third-party auditor. At the same time, this scheme can realize batch auditing. Their proposed scheme is very valuable.

However, we find this scheme is not secure. A malicious CSP can easily forge auditing proof. Even if the CSP deletes all the data of a user, it can still generate the correct data possession proof to pass the verification of TPA. According to our findings, we have carried out the following work:

(1) We give two attack methods to prove the insecurity of Han's scheme. The first attack proves that the audit proof can be forged by the CSP, and the second attack proves that the CSP can pass the verification of the TPA even if it deletes the user's data.

(2) Based on the original scheme, we propose an improved scheme, which can effectively resist the forgery attack from CSP.

2. System Model and Design Goals

2.1. The System Model. The cloud storage system (CSS) includes three entities as depicted in Figure 1: users, CSP, and TPA. The specific definitions are as follows:

(1) *Users*: the owner of the data, outsources the data to the CSP for storage, and delivers the audit work to the TPA.

(2) *CSP*: a provider of cloud storage services, has large storage space and powerful computing capabilities, and can realize data sharing.

(3) *TPA*: the third-party auditor, generates the authentication metaset for users' data and audits the integrity of data stored in the cloud for users.

As depicted in Figure 2, the workflow of this scheme is as follows:

(1) When a user needs to store a data file in the cloud server, they blind it and send the blinded data file to the CSP and TPA. Then, they delete the local data;

(2) After receiving the blinded data from user, the TPA generates the tags for the data and sends it to the CSP;

(3) In order to ensure whether their data is correctly stored in the cloud server, the user sends an auditing request to TPA;

(4) Upon receiving the auditing request, the TPA randomly selects a small set of data blocks as the audit objects and sends an auditing challenge to the CSP;

(5) The cloud server, based on the challenge and the authentication metaset, generates a proof and sends it to the TPA.

(6) After receiving the proof, the TPA verifies the correctness of it. Finally, the TPA sends the auditing report to the user.

2.2. The Design Goals. Our cloud storage audit scheme would achieve the requirements of public auditability, correctness, and unforgeability.

(1) *Pubic auditability*: TPA can replace the user to remotely audit the integrity of the data when the user does not need to download the data stored in the cloud.

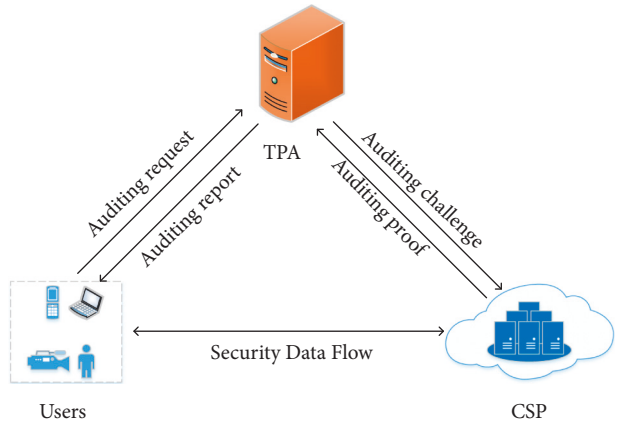


Figure 1: System model.

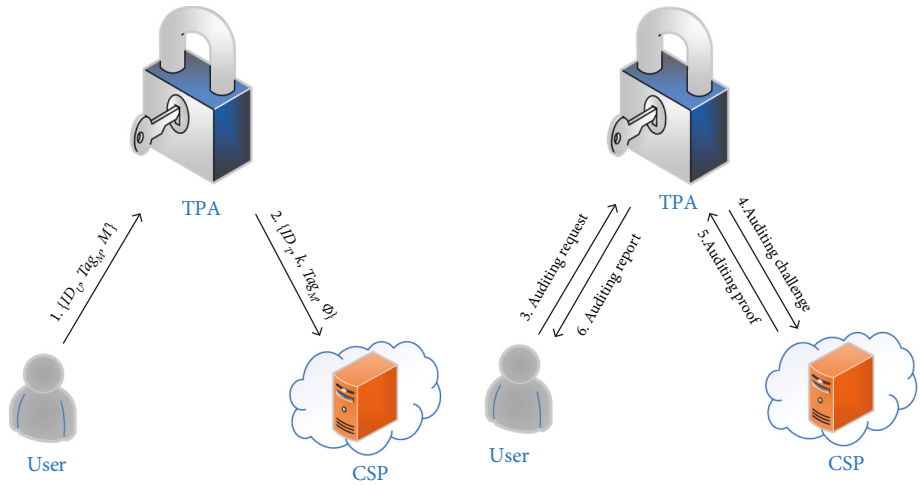


Figure 2: Workflow.

(2) *Correctness*: if CSP honestly stores user data, it can be audited by TPA. Otherwise, the generated proof cannot be verified by TPA.

(3) *Unforgeability*: any party cannot forge the authentication meta set of a user's data unless it has the user's secret key.

3. Review of Han's Scheme

Jing Han et al. (2020) proposed a public cloud auditing scheme, which consists of six algorithms as follows. Before reviewing this scheme, we first introduce the concept of HomMAC (homomorphic message authentication code). For a more specific definition, please refer to [24]. For the specific descriptions of the symbols that appear below, please refer to Table 1.

Given a data block $m_i = (m_{i1}, m_{i2} \ldots, m_{is}) \in Z_q^s$, then computes

$$\rho_i = \sum_{l=1}^{s} \omega_l m_{il} + \varpi_i, \tag{1}$$

where ρ_i is the HomMAC of m_i and $\omega_1, \omega_2, \ldots, \omega_s \in Z_q$, $\varpi \in Z_q$.

(1) *Setup*: input a security parameter λ , and then outputs p, q , which are two large primes. The CSS selects $h(\cdot)$: $\{0,1\}^* \longrightarrow Z_q$ and G . The CSS sets a PRG: $\mathfrak{R}_{prg} \longrightarrow Z_q^s$ and PRF: $\mathfrak{R}_{prf} \times \Gamma \longrightarrow Z_q$. Besides, the CSS set two time upper limits Δ_S and Δ_A, where Δ_S is the longest time for CSP to generate auditing proof, Δ_A is the longest time for the TPA to generate authentication meta set. Finally, the $cp = \{p, q, G, g, \text{PRF}, \text{PRG}, h(\cdot), \Delta_S, \Delta_A\}$ is made public.

(2) *KeyGen*: the identifier of TPA is $\text{ID}_T \in Z_q$ and $\text{ID}_U \in Z_q$. TPA generate their secret $sk_T \in Z_q^*$ and public key $pk_T = g^{sk_T}$. The user generates this secret/public key pair $(sk_u pk_u)$ from cp . Besides, the user chooses s random values $\alpha_1 \alpha_2 \ldots, \alpha_s \in Z_q^*$ and keeps them secret.

(3) SigGen:

3.1. SigGen1. First, the user divided file M into n data blocks. Then, divided each data block into s segments. Then, they establishe a unique tag $\text{Tag}_M = \text{name} \| \text{SSig}_{skU}(\text{name})$ for the file M.

Table 1: Symbols.

Symbols	Descriptions
G	A multiplicative cyclic group.
h	A secure hash function such that $h(\cdot)$: $\{0,1\}^* \longrightarrow Z_p$.
Z_q	A prime field.
q	The order of group G.
g	The generator of group G.
Γ	The index set of data blocks.
$m = \{(m_{11}, ..., m_{ns})\}$	The user's data file with n blocks and s slices.
α	The secret random values of user.
Δ	Time upper limit.
k	a key pair, where $k_g \in \mathfrak{R}_{prg}$ and $k_f \in \mathfrak{R}_{prf}$.
ID_U	The identifier of user.
ID_T	The identifier of TPA.
sk_U	The secret key of user.
pk_U	The public key of user.
sk_T	The secret key of TPA.
pk_T	The public key of TPA.
t	Time stamp.
Tag_M	The unique tag of the file M.
σ_i	The authentication label for the i-th data block.
Φ	The authentication meta set of data blocks.
chal	The challenge set.
p	The proof generated by CSP or auditor.

The user blinds each data blocks to protect the privacy of the file M as follows:

Chooses a random value $u \in_R G$ and then compute $\beta_l = u^{\alpha_l} \in G$ and $\varphi_l = h(\beta_l)$, where $l = 1, 2, \ldots, s$. Blind each data block m_i:

$$\begin{aligned} m'_{il} &= (\alpha_l m_{il} + \varphi_l) \bmod q, \quad l = 1, 2, \ldots, s, \\ m'_i &= (m'_{i1}, m'_{i2}, \ldots, m'_{is}). \end{aligned} \tag{2}$$

The blinded file is $M' = (m'_1, m'_2, \ldots, m'_n)$.

Finally, the user sends $\{M', \mathrm{Tag}_M, \mathrm{ID}_U\}$ to TPA and sends $\{\mathrm{ID}_U, \mathrm{Tag}_M, M', t_{s1}\}$ to CSP.

3.2. SigGen2. The TPA choose a key pair $k = (k_g, k_f)$, where $k_g \in \mathfrak{R}_{prg}$ and $k_f \in \mathfrak{R}_{prf}$. Then, they compute

$$\begin{aligned} \omega = (\omega_1, \omega_2, \ldots, \omega_s) &\longleftarrow \mathrm{PRG}\left(k_g\right), \omega_1, \omega_2, \ldots, \omega_s \in Z_q, \\ \varpi_i &\leftarrow \mathrm{PRF}\left(k_f\right), \varpi_1, \varpi_2, \ldots, \varpi_n \in Z_q. \end{aligned} \tag{3}$$

and the HomMAC:

$$\rho_i = \sum_{l=1}^{s} \omega_l m'_{il} + \varpi_i, i = 1, 2, \ldots, n. \tag{4}$$

The TPA compute $r_i = g^{\eta_i}$ and $s_i = (r_i \eta_i + \rho_i sk_T) \bmod q$, and then output $\sigma_i = (r_i, s_i)$, where $\eta_i \in Z_q^*$ is random value. Let $\Phi = \{\sigma_i\}$ be the authentication meta set of data blocks m'_i. Then, $\{\mathrm{ID}_T, k, \mathrm{Tag}_M, \Phi\}$ is sent to the CSP.

3.3. Storage. The CSP stores file M'.

When receiving the data from TPA, the CSP records time stamp t_{s2}, and computes:

$$\Delta'_s = t_{s2} - t_{s1}. \tag{5}$$

If $\Delta'_S > \Delta_S$, the CSP refuse to store data. Otherwise, they store data.

Next, the CSP computes the validity of Φ by performing the following computations:

$$\omega = (\omega_1, \omega_2, \ldots, \omega_s) \leftarrow \mathrm{PRG}\left(k_g\right), \quad \omega_1, \omega_2, \ldots, \omega_s \in Z_q, \tag{6}$$

$$\varpi_i \leftarrow \mathrm{PRF}\left(k_f\right), \varpi_1, \varpi_2, \ldots, \varpi_n \in Z_q, \tag{7}$$

$$g^{s_i} = r_i^{r_i} \cdot pk_T^{\sum_{l=1}^{s} \omega_l \cdot m'_{il} + \varpi_i} \bmod p. \tag{8}$$

If the (8) holds, the CSP stores the file and other information.

3.4. Challenge. The user sends an auditing request to the TPA. If it is validity, the TPA generates an auditing challenge chal as follows:

The TPA randomly chooses c elements as a subset $I \in \Gamma$ and chooses a random value $v_i \in Z_q^*$. Then, output $\mathrm{chal} = \{(i, v_i)\}$. Finally, they send the chal to the CSP.

3.5. ProofGen. The CSP computes:

$$\begin{aligned} R &= \prod_{i \in I} r_i^{v_i s_i} \bmod q, \\ S &= \sum_{i \in I} v_i s_i \bmod q, \\ \mu_l &= \sum_{i \in I} v_i m'_{il} \bmod q, \quad l = 1, 2, \ldots, s. \end{aligned} \tag{9}$$

Then, the proof $\{\mu, R, S\}$ is sent to the TPA, in which $\mu = (\mu_1, \mu_2, \ldots, \mu_s)$.

ProofVer: after receiving the proof, the TPA records time stamp t_{A2} immediately and computes $\Delta_A^{'} = t_{A2} - t_{A1}$. If $\Delta_A^{'} > \Delta_A$, stop audit work and return "*Expiration*" to the CSP. Otherwise, proceed to the following steps.

Compute:

$$\tau = \left(\sum_{l=1}^{s} (\omega_l \mu_l) + \sum_{i \in I} (v_i \varpi_i) \right) \bmod q. \tag{10}$$

Then, verify the following equation:

$$g^s = R \cdot pk_T^{\tau} \bmod p. \tag{11}$$

If the (11) does not hold, the TPA concludes that the user's data is corrupted. Otherwise, the TPA believe the user's data is integrity. Finally, the TPA sends the auditing report to the user.

3.6. Attack I. In this section, we will show the scheme of Jing Han et al. is not secure by giving the attack I. From the protocol of *ProofVer*, we can know that the TPA verifying the integrity of the data stored in the CSP by determines whether the following equation holds:

$$g^{s'} = R \cdot pk_T^{\tau} \bmod p. \tag{12}$$

Through observation, we can obtain the following information:

(1) The pk in this equation is the public key of user, which can be obtained by the CSP

(2) $\tau = (\sum_{l=1}^{s} (\omega_l \mu_l) + \sum_{i \in I} (v_i \varpi_i)) \bmod q$, the CSP can obtain the ω_l, ϖ_i and v_i , and the μ is computed by CSP

(3) The S and R is generated by CSP

Through the abovementioned points, the CSP can forge an auditing proof. The specific process is as follows:

(1) In the *audit phase*, after receiving the chal from TPA, the CSP randomly chooses s numbers as $\mu_l^{'} \in Z_q$, $l = 1, 2, \ldots, s$.

(2) The CSP computes the $\tau^{'}$ based on the $\mu_l^{'} \in Z_q$, $l = 1, 2, \ldots, s$:

$$\tau^{'} = \left(\sum_{l=1}^{s} (\omega_l \mu_l^{'}) + \sum_{i \in I} (v_i \varpi_i) \right) \bmod q, \tag{13}$$

and then computes the value of pk_T^{τ}.

(3) The CSP randomly selects a number as $S\prime \in Z_q$, and computes $g^{S\prime}$

(4) With the value of pk_T^{τ} and $g^{S\prime}$, the CSP computes $R\prime$:

$$R^{'} = \left(\frac{g^{s'}}{pk_T^{\tau}} \right) \bmod q, \tag{14}$$

(5) Finally, the CSP generates the forged proof $p^{'} = \{\mu^{'}, R^{'}, S^{'}\}$ and sends it to the TPA, where $\mu\prime = (\mu_1^{'}, \mu_2^{'}, \ldots, \mu_s^{'})$.

3.7. Attack II. Our attack II is based on this observation because the CSP can forge the auditing proof without using the blinded data of the user, which has been proved in the attack I. The malicious CSP can even delete the data stored in the cloud server but can still pass the verification of the TPA. Concretely, the attack is as follows:

(1) In the *storage phase*, after receiving the message from the TPA, the CSP verifies the validity of it and the correctness of Φ as the original scheme. If the message is valid and the Φ is correct, the CSP computes:

$$\omega = (\omega_1, \omega_2, \ldots, \omega_s) \leftarrow \mathrm{PRG}(k_g), \quad \omega_1, \omega_2, \ldots, \omega_s \in Z_q. \tag{15}$$

$$\varpi_i \leftarrow \mathrm{PRF}(k_f), \varpi_1, \varpi_2, \ldots, \varpi_n \in Z_q, \tag{16}$$

then the CSP deletes the user's data file.

(2) In the *audit phase*, to verify the integrity of the data in the CSP, TPA sends a challenge chal to the CSP. Upon receiving the chal, the CSP generates an auditing proof $p^{'} = \{\mu^{'}, R^{'}, S^{'}\}$ according to the method in the attack II. Note that the user's data are not stored when the CSP generates proof at this time

(3) After receiving the proof $p^{'}$ from the CSP, the TPA verifies the correctness of $p^{'}$. First, the TPA generates $\omega = (\omega_1, \omega_2, \ldots, \omega_s), \omega_l \in Z_q \neq$ and $\varpi_i \in Z_q$, where $l = 1, 2, \ldots, s$. Then they compute

$$\tau\prime = \left(\sum_{l=1}^{s} (\omega_l \mu_l^{'}) + \sum_{i \in I} (v_i \varpi_i) \right) \bmod q, \tag{17}$$

based on the $\mu_l^{'}$ from the CSP. Finally, the TPA verifies the whether the following equation holds:

$$g^{s'} = R^{'} \cdot pk_T^{\tau'} \bmod p. \tag{18}$$

Because the $S\prime$ and $R\prime$ in the $p\prime$ all are generated by the CSP, here we can prove the forged proof $p\prime$ is a valid one for the eq. holds:

$$\begin{aligned} g^{s'} &= R^{'} \cdot pk_T^{\tau'} \bmod p \\ &= \left[\left(\frac{g^{s'}}{pk_T^{\tau}} \right) pk_T^{\tau'} \right] \bmod p, \\ &= g^{s'}. \end{aligned} \tag{19}$$

4. Our Improved Scheme

In order to resist the abovementioned attack, in this section, we give our improved security scheme. The details of this scheme are as follows.

4.1. A Single-User Scenario. Based on the original scheme, our scheme consists of six algorithms: Setup, KeyGen, SigGen, Challenge, ProofGen, and ProofVer.

(1) *Setup*: the cloud storage system (CSS) inputs a security parameter λ, and then outputs p, q, which are two large primes. The CSS chooses a secure hash function $h(\cdot)$: $\{0,1\}^* \longrightarrow Z_q$ a multiplicative cyclic group G, where the order of G is q and the generator of G is g. The CSS sets a PRG: $\mathfrak{R}_{prg} \longrightarrow Z_q^s$ and PRF: $\mathfrak{R}_{prf} \times \Gamma \longrightarrow Z_q$. $\Gamma = \{1, 2, \ldots, n\}$ is the index set of data blocks. Besides, the CSS set two time upper limits Δ_S and Δ_A, where Δ_S is the longest time for CSP to generate auditing proof, Δ_A is the longest time for the TPA to generate authentication meta set. Finally, the $cp = \{p, q, G, g, \mathrm{PRF}, \mathrm{PRG}, h(\cdot), \Delta_S, \Delta_A\}$ is made public.

(2) *KeyGen*: the identifier of TPA is $\mathrm{ID}_T \in Z_q$ and $\mathrm{ID}_U \in Z_q$. TPA generate their secret $sk_T \in Z_q^*$ and public key $pk_T = g^{sk_T}$. The user generate this secret/public key pair (sk_u, pk_u) from cp. Besides, the user chooses s random values $\alpha_1, \alpha_2, \ldots, \alpha_s \in Z_q^*$ and keeps them secret.

(3) *SigGen*: this algorithm is run by user, TPA, and CSP, including three subalgorithms SigGen1, SigGen2, and storage.

4.1.1. SigGen 1. The user processes the file and generates the tags of data blocks.

First, the user divided file M into n data blocks and each data block is divided into s segments.

$$\begin{aligned} M &= \{m_1, m_2, \ldots, m_n\}, \\ m_i &= m_{i1}, m_{i2}, \ldots, m_{is}, \quad i \in 1, 2, \ldots, n. \end{aligned} \tag{20}$$

Then, they establishe a unique tag $\mathrm{Tag}_M = \mathrm{name} \| \mathrm{SSig}_{skU}(\mathrm{name})$ for the file M, where the $\mathrm{SSig}_{skU}(\mathrm{name})$ is the signature of the file's name using sk_U.

The user blinds each data blocks to protect the privacy of the file M as follows:

Chooses a random value $u \in_R G$ and then compute $\beta_l = u^{\alpha_l} \in G$ and $\varphi_l = h(\beta_l)$, where $l = 1, 2, \ldots, s$. Blind each data block m_i:

$$\begin{aligned} m'_{il} &= (\alpha_l m_{il} + \varphi_l) \mathrm{mod} q, \; l = 1, 2, \ldots, s, \\ m'_i &= (m'_{i1}, m'_{i2}, \ldots, m'_{is}). \end{aligned} \tag{21}$$

The blinded file is $M\prime = (m'_1, m'_2, \ldots, m'_n)$.

Finally, the user sends $\{\mathrm{ID}_U, \mathrm{Tag}_M, M\prime\}$ to the TPA and sends $\{\mathrm{ID}_U, \mathrm{Tag}_M, M\prime, t_{s1}\}$ to CSP.

4.1.2. SigGen 2. The TPA generates authentication meta set for the user.

The TPA choose a key pair $k = (k_g, k_f)$, where $k_g \in \mathfrak{R}_{prg}$ and $k_f \in \mathfrak{R}_{prf}$. Then, they compute

$$\begin{aligned} \omega = (\omega_1, \omega_2, \ldots, \omega_s) &\leftarrow \mathrm{PRG}(k_g), \omega_1, \omega_2, \ldots, \omega_s \in Z_q, \\ \varpi_i &\leftarrow \mathrm{PRF}(k_f), \varpi_1, \varpi_2, \ldots, \varpi_n \in Z_q, \end{aligned} \tag{22}$$

and the HomMAC:

$$\rho_i = \sum_{l=1}^{s} \omega_l m'_{il} + \varpi_i, \; i = 1, 2, \ldots, n. \tag{23}$$

The TPA compute $r_i = g^{\eta_i}$ and $s_i = (r_i \eta_i + \rho_i sk_T) \mathrm{mod} q$, and then output $\sigma_i = (r_i, s_i)$, where $\eta_i \in Z_q^*$ is random value. Let $\Phi = \{\sigma_i\}$ be the authentication meta set of data blocks m'_i for $i = 1, 2, \ldots, n$. Then, the TPA send $\{\mathrm{ID}_T, k, \mathrm{Tag}_M, \Phi\}$ to the CSP and delete the file $M\prime$ from their local record.

4.1.3. Storage. The CSP stores file M'.

When receiving the data from TPA, the CSP records time stamp t_{s2}, and computes:

$$\Delta'_S = t_{s2} - t_{s1}. \tag{24}$$

If $\Delta'_S > \Delta_S$, the CSP refuse to store data. Otherwise, they store data.

Next, the CSP computes the validity of Φ by performing the following computations:

$$\begin{aligned} \omega = (\omega_1, \omega_2, \ldots, \omega_s) &\leftarrow \mathrm{PRG}(k_g), \omega_1, \omega_2, \ldots, \omega_s \in Z_q, \\ \varpi_i &\leftarrow \mathrm{PRF}(k_f), \varpi_1, \varpi_2, \ldots, \varpi_n \in Z_q. \end{aligned} \tag{25}$$

$$g^{s_i} = r_i^{r_i} \cdot pk_T \sum_{l=1}^{s} s\omega_l \cdot m'_{il} + \varpi_i \, \mathrm{mod} p. \tag{26}$$

If the (26) holds, the CSP returns "Correct" to the user and stores the file, the file tag Tag_M and Φ. Otherwise, the CSP does not store the file and returns "Error" to the user.

4.1.4. Challenge. The user sends an auditing request to the TPA. If it is validity, the TPA generates an auditing challenge chal as follows:

The TPA randomly chooses c elements as a subset $I \in \Gamma$ and chooses a random value $v_i \in Z_q^*$ for each element $i \in I$. Then, output chal $= \{(i, v_i)\}$ for $i \in I$. Finally, they send the chal to the CSP and record the time stamp t_{A1} immediately.

4.1.5. ProofGen. After receiving the chal, the CSP computes the proof p.

The CSP computes:

$$\begin{aligned} S &= \sum_{i \in I} v_i s_i \mathrm{mod} q, \\ \mu_l &= \sum_{i \in I} v_i m'_{il} \mathrm{mod} q, \; l = 1, 2, \ldots, s. \end{aligned} \tag{27}$$

Then, they generates the proof $p = \{\mu, S\}$ and sends it to the TPA, where $\mu = (\mu_1, \mu_2, \ldots, \mu_s)$. Note that CSP no longer needs to generate R an element of audit proof.

4.1.6. ProofVer. After receiving the proof, the TPA records time stamp t_{A2} immediately and computes $\Delta'_A = t_{A2} - t_{A1}$. If $\Delta'_A > \Delta_A$, stop audit work and return "*Expiration*" to the CSP. Otherwise, proceed to the following steps.

Compute:

$$R = \prod_{i \in I} r_i^{v_i s_i} \bmod q,$$
$$\tau = \left(\sum_{l=1}^{s} (\omega_l \mu_l) + \sum_{i \in I} (v_i \varpi_i) \right) \bmod q. \tag{28}$$

Then, verify the following equation:

$$g^s = R \cdot pk_T^{\tau} \bmod p. \tag{29}$$

If the (29) does not hold, the TPA concludes that the user's data are corrupted. Otherwise, the TPA believe the user's data are integrity. Finally, the auditing report is sent to the user.

4.2. A Multiuser Scenario. In edge computing, it is common for multiple end users to apply for an audit at the same time. Compared with the single-user scheme, batch auditing can reduce the computational consumption and thus improve the auditing efficiency. In this section, we extend the scheme in section 6.1 to the one that TPA can conduct batch auditing for multiple users.

Suppose there are N users. They send their auditing requests to the TPA. In the three phases of Setup, KeyGen and SigGen, users, TPA and CSP do the same as described in section 6.1.

(1) *Challenge*: upon receiving the auditing requests, the TPA randomly chooses c elements as a subset $I \in \Gamma$ and chooses a random value $v_i \in Z_q^*$ for each element $i \in I$. Then, output chal $= \{(i, v_i), Ms\}$, where Ms includes the message of the N users. Finally, they send the chal to the CSP and record the time stamp t_{A1} immediately.

(2) *ProofGen*: after receiving the chal from TPA, CSP perform the following calculations:

$$S = \sum_{\theta=1}^{N} \sum_{i \in I} v_i s_i^{(\theta)} \bmod q, \quad \theta = 1, 2, \ldots, \mathrm{N},$$
$$\mu_l^{(\theta)} = \sum_{i \in I} v_i m_{il}^{(\theta)} \bmod q, \quad l = 1, 2, \ldots, s. \tag{30}$$

Then, the TPA send auditing proof $p = \{\mu^{(\theta)}, S\}$ to the CSP, where $\mu^{(\theta)} = (\mu_1^{(\theta)}, \mu_2^{(\theta)}, \ldots, \mu_s^{(\theta)})$.

(3) *ProofVer*: after receiving the proof, the TPA records time stamp t_{A2} immediately and computes $\Delta_A' = t_{A2} - t_{A1}$. If $\Delta_A' > \Delta_A$, stop audit work and return "*Expiration*" to the CSP. Otherwise, the TPA computes:

$$\omega^{(\theta)} = \left(\omega_1^{(\theta)}, \omega_2^{(\theta)}, \ldots, \omega_s^{(\theta)}\right) \leftarrow \mathrm{PRG}\left(k_g^{(\theta)}\right)$$
$$\omega_1^{(\theta)}, \omega_2^{(\theta)}, \ldots, \omega_s^{(\theta)} \in Z_q \leftarrow \mathrm{PRF}\left(k_f^{(\theta)}, i^{(\theta)}\right). \tag{31}$$

Then, compute:

$$R = \prod_{\theta=1}^{N} \prod_{i \in I} r_i^{v_i s_i} \bmod q,$$
$$\tau^{(\theta)} = \left(\sum_{l=1}^{s} \left(\omega_l^{(\theta)} \mu_l^{(\theta)}\right) + \sum_{i \in I} \left(v_i \varpi^{(\theta)}\right) \right) \bmod q, \tag{32}$$
$$\widetilde{\tau} = \sum_{\theta=1}^{N} \tau^{(\theta)} \bmod q.$$

Finally, verify the following equation:

$$g^s = R \cdot pk_T^{\widetilde{\tau}} \bmod p. \tag{33}$$

If the (29) does not hold, the TPA concludes that the users' data is corrupted. Otherwise, the TPA believes the users' data is integrity. Finally, the auditing reports are sent to the users.

5. Security Analysis

In this section, we first prove the correctness of the improved scheme. Then, we prove that the auditing proof cannot be forged, which proves that our proposed scheme can resist attack I and attack II. The proof process of privacy preserving users' data can refer to Han's scheme.

5.1. Correctness. The correctness of verification (8) is proved as follows:

$$\begin{aligned} g^{s_i} &= g^{r_i \eta_i + \rho_i \cdot sk_T} \bmod p \\ &= g^{r_i \eta_i} + g^{\rho_i \cdot sk_T} \bmod p \\ &= r_i^{r_i} pk_T^{\sum_{l=1}^{s} \omega_l m_{il} + \varpi_i} \bmod p. \end{aligned} \tag{34}$$

The correctness of verification (11) is elaborated as follows:

$$\begin{aligned} g^S &= g^{\sum_{i \in I} v_i s_i} \bmod p \\ &= g^{\sum_{i \in I} v_i (r_i \eta_i + \rho_i \cdot sk_T)} \bmod p \\ &= R \cdot pk_T^{\sum_{l=1}^{s} \omega_l \sum_{i \in I} v_i m_{il}' + \sum_{i \in I} v_i \varpi_i} \bmod p \\ &= R \cdot pk_T^{\sum_{l=1}^{s} \omega_l \mu_l + \sum_{i \in I} v_i \varpi_i} \bmod p \\ &= R \cdot pk_T^{\tau} \bmod p. \end{aligned} \tag{35}$$

The correctness of verification (33) is proved in the following:

$$
\begin{aligned}
g^{S} &= g^{\sum_{\theta=1}^{N}\sum_{i\in I}\nu_i s_i^{(\theta)}} \bmod q \\
&= g^{\sum_{\theta=1}^{N}\sum_{i\in I}\nu_i\left(r_i^{(\theta)}\eta^{(\theta)}+\rho_i^{(\theta)}\cdot sk_T\right)} \bmod p \\
&= R\cdot\prod_{\theta=1}^{N} pk_T^{\sum_{i\in I}\nu_i\rho_i^{(\theta)}} \bmod p \\
&= R\cdot\prod_{\theta=1}^{N} pk_T^{\sum_{i\in I}\nu_i\left(\sum_{l=1}^{s}\omega_l^{(\theta)}m_{il}^{\prime(\theta)}\varpi_i^{(\theta)}\right)} \bmod p \\
&= R\cdot\prod_{\theta=1}^{N} pk_T^{\sum_{l=1}^{s}\omega_l^{(\theta)}\mu_l^{(\theta)}+\sum_{i\in I}\nu_i\varpi_l^{(\theta)}} \bmod p \\
&= R\cdot\prod_{\theta=1}^{N} pk_T^{\tau^{(\theta)}} \bmod p \\
&= R\cdot pk_T^{\widetilde{\tau}} \bmod p.
\end{aligned}
\tag{36}
$$

5.2. Unforgeability. In our improved scheme, a malicious CSP cannot forge a correct audit proof that can pass the verification of TPA.

Proof. the malicious CSP forge a proof $\widehat{p}=\left\{\widehat{\mu},\widehat{S}\right\}$. If it is valid, the (7) will hold.

$$
\begin{aligned}
g^{\widehat{s}} &= \widehat{R}\cdot pk_T^{\tau} \bmod p, \\
pk_T^{\tau} &= \frac{g^{\widehat{S}}}{\widehat{R}}.
\end{aligned}
\tag{37}
$$

Because P is valid, (38) must hold.

$$
\begin{aligned}
g^{s} &= R\cdot pk_T^{\tau} \bmod p, \\
pk_T^{\tau} &= \frac{g^{S}}{R}.
\end{aligned}
\tag{38}
$$

According to (37) and (38), we can get:

$$
\frac{g^{\widehat{S}}}{\widehat{R}}=\frac{g^{S}}{R}.
\tag{39}
$$

In the original scheme, both R and S are calculated by the CSP and sent to the TPA, so the CSP can easily calculates the value of g^S/R and then forges $g^{\widehat{S}}/\widehat{R}$ that makes the (39) hold according to the method in attack 1. However, in our improved scheme, R is generated by TPA, so the (40) must hold.

$$
g^{S-\widehat{S}}=R\widehat{R}^{-1}.
\tag{40}
$$

From the abovementioned equations, $S=\widehat{S}$ and $R=\widehat{R}$ must hold. Otherwise, we can easily get the value of $S-\widehat{S}$ when $g, g^{S-\widehat{S}}\in G$ is given. It means that there is a solution of a DLP instance in G. However, this contradicts to the proven DLP difficult problem. Therefore, a malicious CSP cannot forge a valid auditing proof to pass the verification of TPA. □

5.3. Privacy Preserving. The proposed scheme provides privacy preserving for users' data.

Proof. before sending the data to TPA and CSP, the user has blinded each data block by using random mask technique as follows:

$$
\begin{aligned}
m_{il}^{\prime} &= (\alpha_l m_{il}+\varphi_l)\bmod q, \quad l=1,2,\ldots,s, \\
m_i^{\prime} &= (m_{i1}^{\prime}, m_{i2}^{\prime},\ldots,m_{is}^{\prime}),
\end{aligned}
\tag{41}
$$

where $u\in_R G$, $\beta_l=u^{\alpha_l}\in G$, $\varphi_l=h(\beta_l)$, $l=1,2,\ldots,s$. The curious TPA or CSP may want to obtain some privacy information of user from the blinded data $M\prime$. Only know the value of $\alpha_l\in_R Z_q^*$ for $l=1,2,\ldots,s$ can they do that successfully. However, that computing α_l given $\beta_l=u^{\alpha_l}\in G$ is to solve the DLP in G, which is infeasible in calculation. Therefore, the curious TPA or CSP have no ability to get privacy information of user's data. □

6. Conclusion

In edge computing, it will do great harm to the running of terminal users if their data stored in the CSP can be deleted without being found. In this paper, we proved that Han's scheme is not secure because the cloud server provider can successfully forge auditing proof to prove to TPA that it honestly stores users' data. Then, we proposed an improved scheme that can effectively avoid the forgery attack from the cloud server.

In the future, cloud storage auditing schemes will be proposed to adapt to more different situations in edge computing, but we should give more attention to the security of the schemes.

Data Availability

The data of this article are available on request from the authors.

Conflicts of Interest

There are no potential conflicts of interest.

Authors' Contributions

Zhengge Yi and Lixian Wei contributed equally to this work. Zhengge Yi is responsible for the writing of the article and the construction of new scheme, Lixian Wei is responsible for the derivation of the formulas in the article and gives some significant ideas, Haibin Yang is responsible for the polishing of the language of the article, Xu An Wang gives the main ideas for the writing of this article, Wenyong Yuan is responsible for collecting the information related to this article, and Ruifeng Li is responsible for the verification of the security of this article.

Acknowledgments

This work is supported by the Foundation of Foundation of National Natural Science Foundation of China (No.

62172436), State Key Laboratory of Public Big Data (No. 2019BDKFJJ008), Engineering University of PAP's Funding for Scientific Research Innovation Team (No. KYTD201805), and Engineering University of PAP's Funding for Key Researcher (No. KYGG202011).

References

[1] L. Ren, Y. Laili, L. Xiang, and X. Wang, "Coding-based large-scale task assignment for industrial edge intelligence," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2286–2297, 2020.

[2] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for Internet of Things," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.

[3] L. Ren, Y. Liu, X. Wang, and J. Lu, "Cloud-edge-based lightweight temporal convolutional networks for remaining useful life prediction in IIoT," *IEEE Internet of Things Journal*, vol. 8, no. 16, Article ID 12587, 2021.

[4] L. Kong, L. Wang, W. Gong, C. Yan, Y. Duan, and L. Qi, "LSH-aware multitype health data prediction with privacy preservation in edge environment," *World Wide Web*, vol. 1, no. 9, 2021.

[5] R. Bi, Q. Liu, J. Ren, and G. Tan, "Utility aware offloading for mobile-edge computing," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 239–250, 2021.

[6] Y. Yi, Z. Zhang, L. T. Yang, X. Deng, L. Yi, and X. Wang, "Social interaction and information diffusion in social Internet of Things: dynamics, CloudEdge, traceability," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2327–4662, 2020.

[7] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for IoV in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[8] Z. He and J. Zhou, "Inference attacks on genomic data based on probabilistic graphical models," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 225–233, 2020.

[9] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for Internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[10] X. Xie, X. Yang, X. Wang, H. Jin, D. Wang, and X. Ke, "BFSI-B: an improved K-hop graph reachability queries for cyber-physical systems," *Information Fusion*, vol. 38, no. 2, pp. 35–42, 2017.

[11] W. Zhang, X. Chen, and J. Jiang, "A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems," *Tsinghua Science and Technology*, vol. 26, no. 1, pp. 95–111, 2021.

[12] G. Orsini, D. Bade, and W. Lamersdorf, "Computing at the mobile Edge: Designing Elastic Android Applications for Computation offloading," in *Proceedings of the 9th Conference on the Joint IFIP Wireless and Mobile Networking (WMNC'16)*, pp. 112–119, Colmar, France, July 2016.

[13] K. Yang and X. Jia, "Data storage auditing service in cloud computing: challenges, methods and opportunities," *World Wide Web*, vol. 15, no. 4, pp. 409–428, 2012.

[14] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-Aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1145–1153, 2021.

[15] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson et al., "Provable Data Possession at Untrusted stores," in *Proceedings of the Acm Conference on Computer & Communications Security ACM*, Alexandria, Virginia, USA, October 2007.

[16] A. Juels and B. S. Kaliski, "PoRs: Proofs of Retrievability for Large Files," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS' 07).ACM Press*, pp. 584–597, Alexandria, Virginia, USA, October 2007.

[17] G. Ateniese, R. D. Pietro, and L. V. Mancini, "Scalable and Efficient Provable Data possession," in *Proceedings of the 4th international conference on Security and Privacy in Communication Networks*, Istanbul Turkey, September 2008.

[18] C. C. Erway, A. Küpçü, C. Papamanthou, and T. Roberto, "Dynamic Provable Data Possession," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS'09)*, pp. 17–38, ACM Press, IL, USA, November 2009.

[19] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceedings of the 29th IEEE Annual International Conference on Computer Communications (INFOCOM'10)*, pp. 1–9, San Diego, CA, USA, March 2010.

[20] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, 2011.

[21] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1717–1726, 2013.

[22] J. Li, L. Zhang, J. K. Liu, H. Qian, and Z. Dong, "Privacy-preserving public auditing protocol for low-performance end devices in cloud," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2572–2583, 2016.

[23] J. Han, Y. Li, and W. Chen, "A Lightweight And privacy-preserving public cloud auditing scheme without bilinear pairings in smart cities," *Computer Standards & Interfaces*, vol. 62, no. FEB, pp. 84–97, 2019.

[24] S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-Based integrity for network coding," in *Proceedings of the International Conf. On Applied Cryptography and Network Security, Springer-Verlag*, pp. 292–305, Singapore, June 2009.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 8146656, 14 pages
https://doi.org/10.1155/2022/8146656

Research Article

Research on Automatic Cargo Recognition in Smart City Environment

Lanlan Yin, Feng Mo, Qiming Wu, and Zhixun Liang

Hechi University, Yizhou 546300, China

Correspondence should be addressed to Feng Mo; fengmo@hcnu.edu.cn

Received 12 October 2021; Accepted 11 March 2022; Published 18 April 2022

Academic Editor: Gautam Srivastava

Smart city refers to the use of various information technologies to improve the lives of citizens. However, in terms of transportation and sales of goods, traditional methods require a lot of manpower and material resources, and cannot be automatically identified. In order to improve the efficiency and accuracy of product identification, product sorting is automated. It uses the powerful feature learning and expression capabilities of deep convolutional neural networks to automatically learn product features, thereby achieving high-precision image classification. Therefore, this paper first proposes an improved VGG network, combines transfer learning to establish a deep learning recognition model, and finally conducts multiple sets of experiments on the 131-category Fruit-360 dataset. The results show that when the Adam optimizer is used for iterative training for 30 rounds and the batch_size is 64, the accuracy of the algorithm proposed in this paper reaches 94.19% on the training set, 97.91% on the validation set, and 92.2% on the test set top1. The accuracy rate on the test set top5 is as high as 100%. Therefore, the method in this paper can solve the problems caused by traditional methods and provide useful help for smart cities.

1. Introduction

With the advent of the era of big data and artificial intelligence, people's lives have undergone tremendous changes. Information technology has penetrated into all walks of life and has introduced dramatic changes. How to use various information technologies and innovative concepts to connect and integrate urban systems and services, improve the efficiency of resource utilization, optimize urban management and services, and improve the quality of life of citizens is an urgent problem that needs to be solved. Smart city is an advanced form of urban informatization that fully utilizes the new generation of information technology in all walks of life in the city, and realizes the deep integration of informatization, industrialization, and urbanization, which helps alleviate the "big city disease," improve the quality of urbanization, and achieve refinement and dynamic management, and enhance the effectiveness of urban management and improve the quality of life of citizens.

In order to achieve this goal, it is necessary to change many fields with the help of information technology, such as smart transportation, smart logistics, smart agriculture, Internet of Things, Internet of Vehicles, cloud computing, and smart medical care. At present, many excellent scholars have achieved some good research results, including [1–4] for the research on the Internet of Vehicles, and their research can improve the efficiency and safety of urban traffic. [5, 6] Applying deep learning technology to air quality prediction can take measures to solve air problems in advance, which is of great significance to the construction of smart cities. [7] The application of deep learning technology to the problem of sentiment analysis can sense the emotions and psychology of college students in advance, which has become a research hotspot in the fields of psychology, health medicine, and computer science, and has high practical application value.

At present, in the time of artificial intelligence, deep learning technology has penetrated into all walks of life, causing huge changes in the city. Today, it promotes the process of urban informatization, making urban construction gradually move toward digitization, integration, networking, intelligence, and direction development. However,

in people's daily life, there are still traditional manual cargo sorting methods, which cause a lot of resources and labor costs, which runs counter to the construction of smart cities.

In view of the above problems, this paper combines product classification and packaging with deep learning technology to streamline product packaging and automatic distribution, which can greatly reduce the cost of goods loss and personnel sorting. In addition, the application of this technology is of great significance to the whole process of goods distribution, warehousing, warehousing, inventory counting, etc. It changes the way of traditional manual sorting and recording of information, realizes the active perception of logistics information, and greatly simplifies the logistics distribution process, which improves distribution efficiency and is an indispensable link in the construction of smart cities.

Goods recognition and classification based on deep learning have many applications in many fields, such as autonomous navigation, object modeling, process control, or human-computer interaction. We take the fruits and vegetables that are indispensable in people's daily life as examples for identification research. The most interesting application is to create an autonomous robot that can perform more complex tasks than ordinary industrial robots. An example of this is a robot that can perform inspections in the aisles of a store to identify inappropriate items or shelves with insufficient inventory. In addition, the robot can also be enhanced to enable it to interact with the product so that it can solve problems on its own. In addition, this research is of great help to another field of autonomous fruit harvesting.

Although there have been several papers on this topic, as far as we know, they only focus on a few fruits or vegetables. In this article, we are trying to create a network that can classify a variety of fruits and vegetables so that it can be useful in more situations.

We choose to identify fruits and vegetables for several reasons. On the one hand, there are some indistinguishable categories of fruits and vegetables, such as citrus, including oranges and grapefruits. Therefore, we want to see to what extent artificial intelligence can complete the task of classifying them. Another reason is that fruits are common in stores, so they are a good starting point for the aforementioned projects.

In summary, the main contributions of this paper are as follows:

(1) Based on the background of smart city, an intelligent cargo identification method is proposed, which can greatly save resource costs and improve the efficiency of cargo transportation;

(2) The cargo identification method based on artificial intelligence is the starting point of many projects in related fields, and our high-precision identification algorithm can provide favorable support for related fields;

(3) We did not retrain a neural network, but performed partial training with transfer learning technology, which can greatly save hardware costs;

(4) Considering the difficulty and cost of obtaining training data, we use data augmentation to expand the amount of data, which can greatly save time and cost and improve the generalization ability of the model;

(5) By improving the original VGG network, more advanced features are obtained, making the data and network more stable.

2. Related Works

There are mainly two types of target detection models based on deep learning, namely, two-stage target detection models and single-stage target detection models. In 2014, Facebook Artificial Intelligence Laboratory researcher Ross B. Girishick proposed the Region-CNN (R-CNN for short) algorithm [8], which is a two-stage target detection algorithm. The principle of this algorithm is to first extract candidate regions using heuristic algorithms and then perform feature extraction, target classification, and detection on the candidate regions. The R-CNN algorithm applied deep learning technology to target detection for the first time and achieved good results, but at the same time, there are a large number of redundant feature calculations. In 2015, Girishick used the Spatial Pyramid Pooling Network (SPPNet) to propose the Fast R-CNN algorithm, which greatly shortened the running time [9].

In 2015, Ren Shaoqing of Microsoft Research Asia and others proposed the Faster R-CNN algorithm based on Fast R-CNN. This algorithm is based on a convolutional neural network to obtain the feature map of the entire image and replaces it with a custom region suggestion network. The traditional image block extraction algorithm generates the candidate area frame. The feature expression of each candidate area of a fixed length is obtained from the feature map through the method of the region of interest pooling. Finally, the Softmax classifier is used for classification, and the area is obtained through the bounding box regression. The offset of the actual target frame position makes the detected target frame closer to the real position. The innovation is to use the region suggestion network to improve the extraction method of candidate regions, which significantly improves the speed of obtaining candidate regions. In addition, the process of training the network also sets the parameters of the RPN network and the Fast R-CNN network to share the convolutional layer to further improve the learning efficiency of Fast R-CNN and the speed of network detection [10].

In the single-stage target detection algorithm, there is no step of generating candidate regions, and the position size and target category of the frame to be detected are directly predicted, and the detection step is completed at one time. At present, single-stage target detection algorithms can be divided into anchor-based (algorithms) and anchor-free (algorithms). Typical detection models based on anchor points include SSD, YOLOv3, RetinaNet, SqueezeDet, and DetectNet. The YOLO series is a classic single-stage target detection algorithm [11]. This series of algorithms divide the image to be detected into $n \times n$ images of the same size. Each

area corresponds to a certain bounding box. This type of algorithm has fast detection, low background false detection rate, and strong versatility.

At present, many scholars have done some research in the related fields of product identification. Using advanced computer vision, artificial neural network, and PLC control technology, Zhou Wei and others proposed an intelligent control system for fruit classification based on the FX3U-48MT/ES-A PLC controller and analyzed the working principle of the system and the collection and processing of mango samples. With the recognition model, the software and hardware design of the PLC control system is given. The test results show that the fruit classification intelligent control system can use neural network and computer vision methods to properly classify mangoes, with an accuracy of up to 94.23%, which has very important practical significance [12].

Zhu Ling first introduced the idea and principle of the K-means algorithm, then analyzed and studied the acquisition and preprocessing of fruit images, and finally realized the fruit classification and recognition model combining K-means clustering and BP neural network. The test results show that the combination of K-means clustering and BP neural network greatly improves the accuracy of fruit classification and recognition, and greatly shortens the recognition time, which has certain practical significance [13].

Using visual capture technology, Qin National Defence and others have designed a set of automatic fruit sorting systems for picking robots, including conveying mechanism, image acquisition system, control module, and actuators, which can be sorted according to the diameter of apples. The experimental results show that the system classification accuracy rate reached 93.6%, which meets the design requirements, and did not cause any damage to the apple during the classification process, which has a certain degree of effectiveness and reliability [14].

Song proposed a method to identify and count fruits from cluttered greenhouse images [15]. The target plant is pepper, with complex fruit shapes and variable colors, similar to the canopy of plants. The purpose of this application is to locate and count the green and red pepper fruits in large, densely growing pepper plants in the greenhouse. The training and validation data they used included 28,000 images of more than 1,000 plants and their fruits. The pepper positioning and counting method used are two steps: in the first step, the fruit is positioned in a single image, and in the second step, multiple views are combined to improve the detection rate of the fruit. The method of finding pepper fruits in a single image is (1) based on finding points of interest, (2) applying a complex high-dimensional feature descriptor around the points of interest, and (3) using so-called word bags to perform small areas of classification.

Sa proposed a new method to detect fruits from images using deep neural networks [16]. To this end, the author uses a faster region-based convolutional network. Their goal is to create a neural network that can be used by autonomous robots that can harvest fruits. The network uses RGB and near-infrared images for training. The combination of RGB and near-infrared models is completed in two independent situations: early and late fusion. Early fusion means that the input layer has 4 channels: 3 channels for RGB images and 1 channel for near-infrared images. Later fusion uses two independently trained models to merge by obtaining predictions from the two models and averaging the results. The result is a multimodal network with better performance than existing networks.

In the literature [17–19], a fruit detection method based on color, shape, and texture is proposed. They emphasized the difficulty of correctly classifying different kinds of similar fruits. They suggest combining existing methods to detect regions of interest from images using texture, shape, and color. Similarly, in [20], the shape, size, color, texture, and k-nearest neighbor algorithms are combined to improve the accuracy of recognition.

The latest paper [21] proposed an algorithm based on the improved Chan-Vese level-set model, combining the level-set idea and the M-S model [22]. The recommended goal is to conduct night-time green grape testing. Combining the principle of the smallest circumscribed rectangle of the fruit and the Hough line detection method, the picking point is calculated.

In 2021, Richa will use three methods to solve the carrot classification problem [23]. The three methods are KNN, KNN based on cross-validation, and neural network. For the first two methods, the K value needs to be adjusted manually, which will undoubtedly increase the workload. For the third method it uses, it only achieves 77% accuracy on the validation set, which is not very good.

Haq *Z* A studied classification models based on CNN algorithms [24], focusing on the effects of activation functions and convolutional layers on model accuracy and latency. Using a database of 9600 images of three different fruits: apple, banana, and orange, and it can be seen from the simulation that the combination of ReLu-Softmax as an activation function provides the highest percentage increase in accuracy. It can also be seen from the simulation results that ReLu runs the fastest, but has relatively low accuracy for other activation functions.

Mohammad's model detects the open and closed states of pistachios in videos [25]. It is first trained on a RetinaNet network using our dataset to detect different types of pistachios in video frames. Then, after the detections were collected, they were applied to a new counter algorithm based on the new tracker to distribute pistachios in consecutive frames with high accuracy. The algorithm executes very fast and achieves good counting results. Their algorithm achieved a computational accuracy of 94.75% on six videos (9486 frames).

Siddiqi demonstrated how adversarial training improves the robustness of fruit image classifiers [26]. Three convolutional neural network (CNN)-based classifiers IndusNet, fine-tuned VGG16, and fine-tuned MobileNet are proposed. The fine-tuned VGG16 yielded the best test set accuracy of 94.82%, while the other two models were 92.32% and 94.28%, respectively. However, the proposed study also has some limitations. For example, it is still possible to achieve higher accuracy on undisturbed clear images, further reducing overfitting. Also in this study, little preprocessing

was performed on the dataset images, and image preprocessing can be added to the classification process to improve model accuracy.

The following structure of the paper is organized as follows: first, we will describe the Fruit-360 dataset: how it was created and what it contains. Then, we will introduce the principle of the algorithm used in this article and why we chose it. After that, we will introduce in detail the structure of the neural network we use. Next, we will briefly discuss the experimental environment and hardware configuration of this article. Next, the results obtained using the training and test data are described. Finally, we will summarize some improved methods and plans.

3. Materials and Methods

In the field of image recognition and classification, the most successful result is the use of artificial neural networks. These networks form the basis of most deep learning models. Deep learning is a type of machine learning algorithm that uses multilayer nonlinear processing units. Each level learns to transform its input data into a slightly abstract and composite representation.

Deep neural networks have successfully surpassed other machine learning algorithms. They also achieved the first superhuman pattern recognition in some fields. Deep learning is considered to be an important step in gaining powerful artificial intelligence. Second, deep neural networks, especially convolutional neural networks, have achieved good results in the field of image recognition. For this reason, the convolutional neural network is specially applied to the problem of fruit and vegetable recognition.

Next, the rest of this section is organized. We will first introduce the data used by the algorithm, give an overview of transfer learning, introduce the VGG network and its principles, and finally describe the VGG network model based on transfer learning in this article in detail.

3.1. Materials. This article uses an image dataset of popular fruits. The dataset is named Fruit-360 and can be downloaded from the address pointed to by reference [27, 28]. Currently (as of 2020.05.18), the collection contains 90,483 images of 131 types of fruits and vegetables. Each image contains a fruit or vegetable. As shown in Table 1, we use 75% of the data for training, 12.5% for verification, and 12.5% for the final test. The dataset is also available on GitHub and Kaggle, and the original size of the data is $100 \times 100 \times 3$.

3.2. Improved VGG Network Model. Transfer learning is to make the convolutional neural network model trained on a task suitable for a new task through simple adjustments. The convolutional layer of the trained convolutional neural network can perform feature extraction on the image. The extracted feature vector and then input into the fully connected layer with a simple structure can achieve better recognition and classification, so the feature vector extracted by the convolutional layer can be as an image, a more streamlined and more expressive vector. Therefore, the trained convolutional layer plus the fully connected layer suitable for the new task will form a new network model. A little training on the new network model can handle new classification and recognition tasks.

Transfer learning first keeps the structure of the model convolutional layer unchanged and then loads the trained weights and parameters into the convolutional layer. Then, we designed a fully connected layer for the new task and replaced the original fully connected layer with the newly designed fully connected layer to form a new convolutional network model with the original convolutional layer. Finally, use the new image dataset to train the new model. There are two training methods for the new model. One is to freeze the convolutional layer and train only the fully connected layer, and the other is to train all layers of the network.

Convolutional neural network (CNN) is part of the deep learning model. Such a network can be composed of a convolutional layer, a pooling layer, a ReLU layer, a fully connected layer, and a loss layer. In a typical CNN architecture, each convolutional layer is followed by a rectified linear unit (ReLU) layer, and then, a pooling layer is followed by one or more convolutional layers and finally one or more fully connected layers. One feature that distinguishes CNN from ordinary neural networks is that the structure of the image is taken into account when processing the image.

The convolutional layer calculates the input image and the convolution kernel to generate a new feature map. The size of the convolution kernel is generally 3×3 or 5×5. It should be noted that the depth of the input image is the same as the depth of the convolution kernel. Usually, multiple convolution kernels of different sizes can be extracted from the input image to obtain different feature maps.

Assuming that the width and height of the input image are W_{input}, W_{input}, and H_{input}, H_{input}; the width and height of the convolution kernel are W_{filter}, W_{filter}, and H_{filter}, H_{filter}; the step size is S; and the padding is P, and then, the width and height (W_{out}, W_{out}, H_{out}, H_{out}) of the resulting feature map are defined by the following calculation formulas:

$$\begin{aligned} \text{W}_{\text{out}} &= \frac{W_{\text{input}} - W_{\text{filter}} + 2P}{S} + 1, \\ \text{H}_{\text{out}} &= \frac{H_{\text{input}} - H_{\text{filter}} + 2P}{S} + 1. \end{aligned} \tag{1}$$

The function of the pooling layer is to reduce the size of the model, increase the calculation speed, and also improve the robustness of the extracted features. There are two types of pooling operations, namely, maximum pooling and average pooling.

Assuming that the width and height of the input image are Winput and H_{input}, the width and height of the convolution kernel are W_{filter} and H_{filter}, the step size is S, and the padding is P, the calculation formulas for the width and the height (W_{out}, H_{out}) of the obtained feature map are as follows:

$$\begin{aligned} \text{W}_{\text{out}} &= \frac{W_{\text{input}} - W_{\text{filter}}}{S} + 1, \\ \text{H}_{\text{out}} &= \frac{H_{\text{input}} - H_{\text{filter}}}{S} + 1. \end{aligned} \tag{2}$$

Table 1: Number of images for each fruit.

Category	n_train	n_valid	n_test
Grape blue	984	164	164
Plum 3	900	152	152
Peach 2	738	123	123
Strawberry wedge	738	123	123
Tomato 1	738	123	123
Melon Piel de Sapo	738	123	123
Tomato 3	738	123	123
Cherry rainier	738	123	123
Cherry 2	738	123	123
Walnut	735	124	125
Pear stone	711	118	119
Pepper orange	702	117	117
Cauliflower	702	117	117
Fig	702	117	117
Pear Forelle	702	117	117
Pear 2	696	116	116
Tomato heart	684	114	114
Tomato 2	672	112	113
Apple red yellow 2	672	109	110
Pepper yellow	666	111	111
Pear red	666	111	111
Pepper red	666	111	111
Nut forest	654	109	109
Nut pecan	534	89	89
Pineapple mini	493	81	82
Rambutan	492	82	82
Grape pink	492	82	82
Grape white 3	492	82	82
Grapefruit white	492	82	82
Physalis	492	82	82
Lemon	492	82	82
Pomegranate	492	82	82
Pear	492	82	82
Peach flat	492	82	82
Peach	492	82	82
Papaya	492	82	82
Mulberry	492	82	82
Nectarine	492	82	82
Physalis with husk	492	82	82
Redcurrant	492	82	82
Apple braeburn	492	82	82
Apple red yellow 1	492	82	82
Apple red 2	492	82	82
Cantaloupe 1	492	82	82
Cherry 1	492	82	82
Cherry wax black	492	82	82
Cherry wax red	492	82	82
Tomato cherry red	492	82	82
Apricot	492	82	82
Cherry wax yellow	492	82	82
Cantaloupe 2	492	82	82
Apple red 1	492	82	82
Apple granny smith	492	82	82
Strawberry	492	82	82
Apple golden 2	492	82	82
Avocado ripe	491	83	83
Pear abate	490	83	83
Pineapple	490	83	83
Pepino	490	83	83
Cactus fruit	490	83	83

Table 1: Continued.

Category	n_train	n_valid	n_test
Banana red	490	83	83
Pear Williams	490	83	83
Banana	490	83	83
Apple red delicious	490	83	83
Passion fruit	490	83	83
Pear monster	490	83	83
Dates	490	83	83
Salak	490	81	81
Carambola	490	83	83
Maracuja	490	83	83
Kaki	490	83	83
Granadilla	490	83	83
Tamarillo	490	83	83
Grape white	490	83	83
Tangelo	490	83	83
Cocos	490	83	83
Raspberry	490	83	83
Grapefruit pink	490	83	83
Clementine	490	83	83
Guava	490	83	83
Pitahaya red	490	83	83
Huckleberry	490	83	83
Quince	490	83	83
Kumquats	490	83	83
Meyer Lemon	490	83	83
Limes	490	83	83
Lychee	490	83	83
Mandarine	490	83	83
Mango	490	83	83
Grape white 2	490	83	83
Apple golden 3	481	80	81
Nectarine flat	480	80	80
Apple golden 1	480	80	80
Orange	479	80	80
Tomato 4	479	80	80
Watermelon	475	78	79
Tomato not ripened	474	79	79
Grape white 4	471	79	79
Kohlrabi	471	78	79
Cucumber ripe 2	468	78	78
Eggplant	468	78	78
Kiwi	466	78	78
Hazelnut	464	78	79
Blueberry	462	77	77
Corn husk	462	77	77
Tomato yellow	459	76	77
Apple pink lady	456	76	76
Potato red washed	453	75	76
Banana lady finger	450	76	76
Pomelo sweetie	450	76	77
Corn	450	75	75
Potato sweet	450	75	75
Potato white	450	75	75
Beetroot	450	75	75
Onion red	450	75	75
Chestnut	450	76	77
Potato red	450	75	75
Plum	447	75	76
Onion red peeled	445	77	78
Pepper green	444	74	74

Table 1: Continued.

Category	n_train	n_valid	n_test
Apple crimson snow	444	74	74
Onion white	438	73	73
Apple red 3	429	72	72
Avocado	427	71	72
Mango red	426	71	71
Plum 2	420	71	71
Cucumber ripe	392	65	65
Tomato maroon	367	63	64
Pear kaiser	300	51	51
Mangosteen	300	51	51
Ginger root	297	49	50

As the last layer or multiple layers of the neural network, the fully connected layer plays a role in feature space transformation and dimensionality reduction. It can transform the feature transformation of the previous layer into a new feature space and convert high-dimensional features into one-dimensional features, which is convenient for the final classification prediction of the model.

The classic representative of the convolutional neural network is the VGG16 network, which is composed of 13 convolution modules and 3 fully connected modules. The output number of the last fully connected layer is 1000, corresponding to the number of target categories, and SoftMax is used to calculate the loss, as shown in Figure 1.

VGG-16 completed training on the ImageNet dataset, and the training set alone reached 1.28 million. The amount of data and the number of sample types are enough to get a model with strong expressive ability. However, there is currently no large enough dataset for fruit images, and considering the high training cost, it is difficult to train the network model to the ideal classification effect. Therefore, the method of transfer learning can be used to realize the task of fruit and vegetable classification. We retain the model structure of the first 13 layers in Figure 1 and then redesign the fully connected module. The improved fully connected module is shown in Figure 2.

The input image can be converted into a $7 \times 7 \times 512$ three-dimensional vector after extracting features in the first 13 layers of VGG16, and the dimension is reduced to 1×4096 through the fully connected layer 1. After entering the nonlinear activation function ReLU, the model uses the ReLU activation function. The ReLU function has the characteristics of simple calculation and fast convergence, and its expression is as follows:

$$\text{Re}lu = f(x) = \max(0, x). \tag{3}$$

Then, we enter the dropout layer [29]. The dropout layer temporarily sets the weight of some neurons to 0.5 according to a certain probability during each training process of the network, which can alleviate the coordinated adaptation between neurons and reduce the dependence between neurons, avoid overfitting of the network, and then enter the fully connected layer 2 to further reduce the dimension of the vector to 1×4096.

After that, the ReLU layer is also subjected to nonlinear transformation, and then, into the dropout layer, some neurons are disabled according to the probability of 0.5. Then proceed to the third fully connected layer to reduce the dimensionality of the feature to a one-dimensional vector of 1×256.

Then, the 1×256 features are activated by ReLU, enter the dropout layer to inactivate some neurons with a probability of 0.4, and then reduce the dimensionality to 1×131. Finally, LogSoftmax is performed to calculate the scoring probability of each category. Softmax is not used here because it compresses the value to (0, 1), while the value range of LogSoftmax is $(-\infty, 0)$. This can prevent overflow problems and facilitate the calculation of the loss function.

$$\begin{aligned} \text{Softmax} &= \sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{131} e^{z_i}}, \\ \text{LogSoftmax} &= \log(\sigma(z_i)) = \log\left(\frac{e^{z_i}}{\sum_{j=1}^{131} e^{z_i}}\right). \end{aligned} \tag{4}$$

In summary, the model structure in this section is shown in Figure 3 below. First, we migrate the model parameters of the classic VGG16 network on ImageNet to our VGG16 network, while freezing the first 16 layers of VGG16 and changing the last layer of VGG16. The number of classifications in the connection layer is 131, and then, LogSoftmax is used for classification prediction and verification, and finally tested through the test set.

Based on the overview diagram in Figure 3, the model in this paper can be further divided into a feature extraction part and a classifier part, as shown in Tables 2 and 3 respectively. The input image is a $100 \times 100 \times 3$ image, and the data become $224 \times 224 \times 3$ after data enhancement as the input of the model. After that, feature extraction is performed according to the parameters in Table 2. After each convolution, there is a ReLu nonlinear operation. Because it does not change the input and output sizes, it is not reflected in Table 3.

As shown in Table 3, the feature in Table 2 is first subjected to a linear operation into a 4096-dimensional vector, and then, a 50% dropout operation is performed, and so on. Until the 131-dimensional column vector is finally obtained, the final classification result is obtained through LogSoftmax operation.

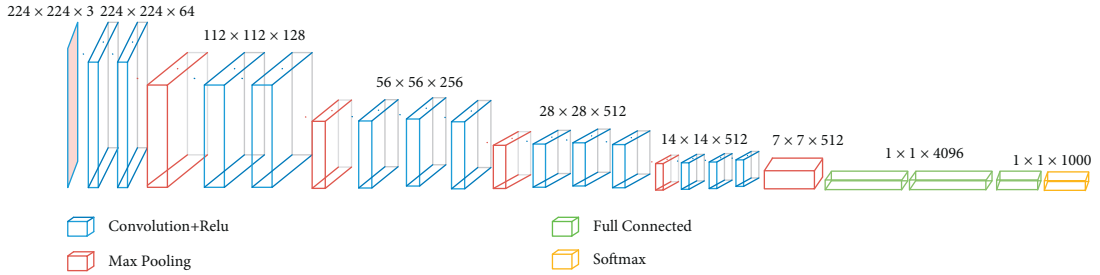


Figure 1: Traditional VGG16 network structure.

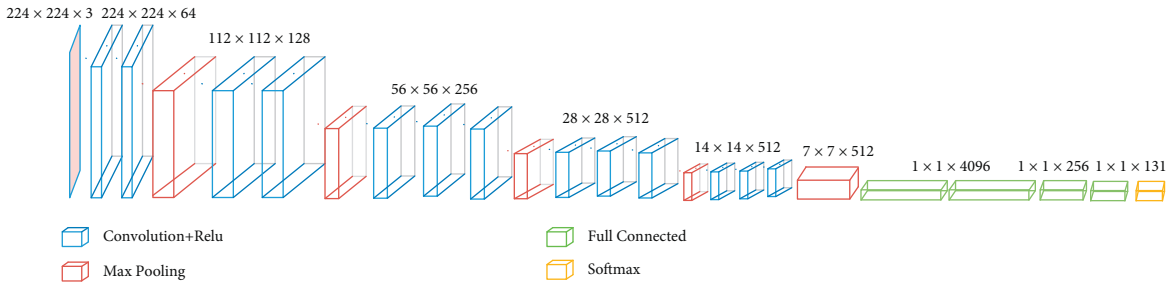


Figure 2: The improved structure of VGG16 in this article.

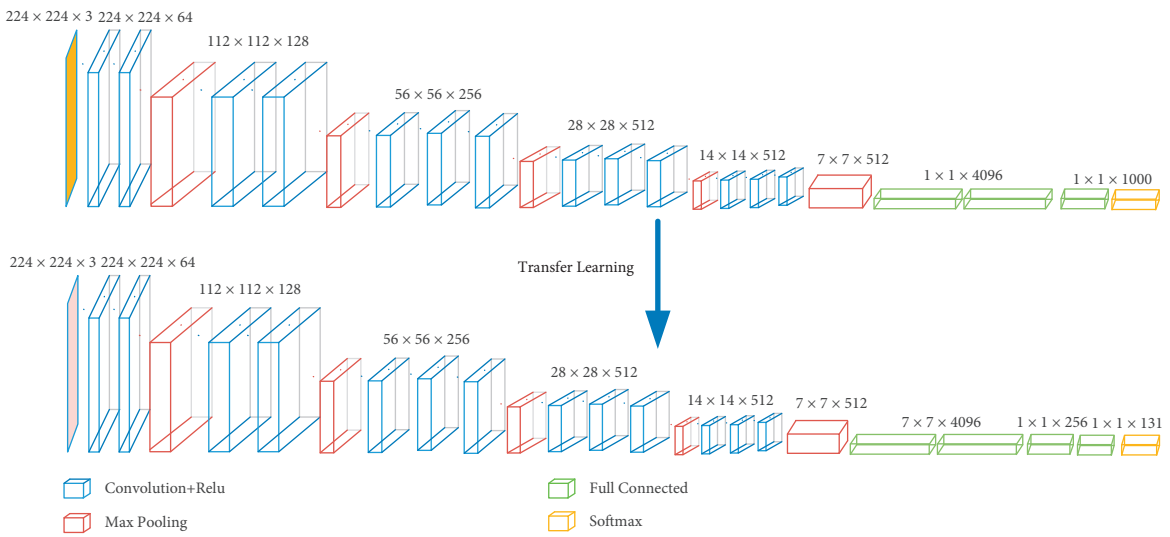


Figure 3: Overview of the network structure mentioned in this article.

4. Results and Discussion

4.1. Experimental Environment. The experiment was completed in Python 3, PyTorch 1.7, and CUDA 10.2 software environment. In the hardware environment, the CPU adopts Intel core i7-10875H, the main frequency is 2.3 GHz eight-core, and the GPU adopts Nvidia GeForce RTX 2070 Super, 8 GB video memory.

4.2. Experimental Results and Analysis. Due to the limited number of images in certain categories, we first use image enhancement to artificially increase the number of images "see" by the network. This means that for the training set, we will randomly adjust, crop, and flip the image horizontally to increase the dataset. Each stage (during training) applies a different random transformation, so the network effectively sees many different versions of the same image. All data are also converted to Torch tensor before normalization. The training set enhancement methods and enhancement effects used in the experiment are shown in Table 4 and Figure 4, respectively. The validation and test data are not augmented, just resized, and normalized, as shown in Table 5. It can be seen that the same image in the training set has been transformed into 16 different images after data enhancement, and this operation can expand the data by 16 times.

Then Batch_size selects 64, and the optimizer selects Adam. The final experiment achieves 94.19% accuracy on the

Table 2: The layout of the features part.

Type	Kernels/Kernel_size	Stride	Input/output
Con2d	64/3 × 3	1	224 × 224 × 3/224 × 224 × 64
Con2d	64/3 × 3	1	224 × 224 × 64/224 × 224 × 64
MaxPool2d	2 × 2	2	224 × 224 × 64/112 × 112 × 64
Con2d	128/3 × 3	1	112 × 112 × 64/112 × 112 × 128
Con2d	128/3 × 3	1	112 × 112 × 128/112 × 112 × 128
MaxPool2d	2 × 2	2	112 × 112 × 128/56 × 56 × 128
Con2d	256/3 × 3	1	56 × 56 × 128/56 × 56 × 256
Con2d	256/3 × 3	1	56 × 56 × 256/56 × 56 × 256
Con2d	256/3 × 3	1	56 × 56 × 256/56 × 56 × 256
MaxPool2d	2 × 2	2	56 × 56 × 256/28 × 28 × 256
Con2d	512/3 × 3	1	28 × 28 × 256/28 × 28 × 512
Con2d	512/3 × 3	1	28 × 28 × 512/28 × 28 × 512
Con2d	512/3 × 3	1	28 × 28 × 512/28 × 28 × 512
MaxPool2d	2 × 2	2	28 × 28 × 512/14 × 14 × 512
Con2d	512/3 × 3	1	14 × 14 × 512/14 × 14 × 512
Con2d	512/3 × 3	1	14 × 14 × 512/14 × 14 × 512
Con2d	512/3 × 3	1	14 × 14 × 512/14 × 14 × 512
MaxPool2d	2 × 2	2	14 × 14 × 512/7 × 7 × 512

Table 3: The layout of the classifier part.

Type	Parameters
Linear	(25088, 4096)
Dropout	0.5
Linear	(4096, 4096)
Dropout	0.5
Linear	(4096, 256)
Dropout	0.4
Linear	(256, 131)
LogSoftmax	

Table 4: Training data augmentation.

Augmentation	Setting
RandomResizedCrop	size = 256, scale = (0.8, 1.0))
RandomRotation	degrees = 15
CenterCrop	size = 224
Normalize	Mean=([0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225])

training set, 98.46% accuracy on the validation set, 93% accuracy on the test set top1, and even more on the test set top5 reaching 100% accuracy rate. Figure 5 below is the error change of the training set and the validation set for 30 rounds of training, and Figure 6 is the accuracy change curve of the training set and the validation set.

Next, the model is tested, and the results of the test set are as follows. As shown in Figures 7 and 8, the accuracy of the top 5 fruits and vegetables such as redcurrant and apricot can reach 100%.

Then, the 131 types of fruits and vegetables in the training set were tested. The average accuracy of top1 reached 92.2, and the average accuracy of top5 reached 100%. The experimental results of some fruits and vegetables are shown in Table 1. It can be seen that on the five fruits and vegetables in Table 6, the model proposed in this article can achieve an accuracy of more than 98%.

Finally, verify the relationship between the number of fruit and vegetable images in the training set and the accuracy of top1. As shown in Figure 9, it can be seen that the number of random images of the model in this paper is increasing, and the accuracy is gradually improving, basically above 90%, and gradually approaching 100%. Figure 10 verifies the relationship between the number of images of fruits and vegetables in the training set and the accuracy of top5. It can be seen that as the number of images increases, the effect of the model in this paper is relatively stable, all at 100%.

Figure 4: Data enhancement results.

Table 5: Val or Test datasets.

Augmentation	Setting
Resize	size = 256,
CenterCrop	size = 224
Normalize	Mean = ([0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225])

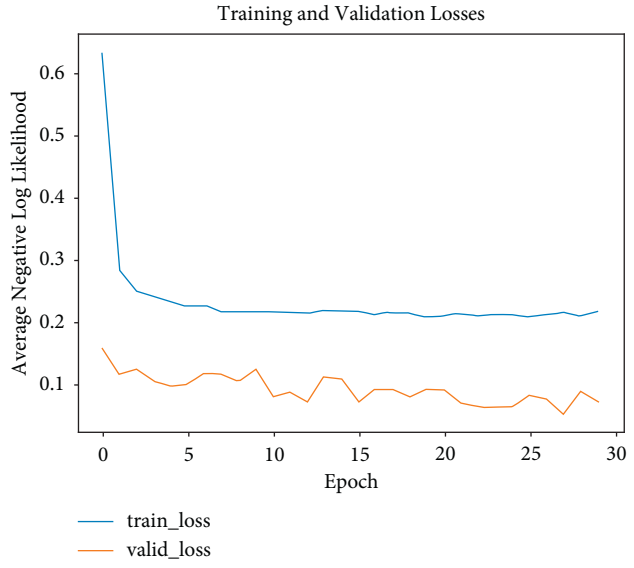


Figure 5: Error curve of training set and validation set.

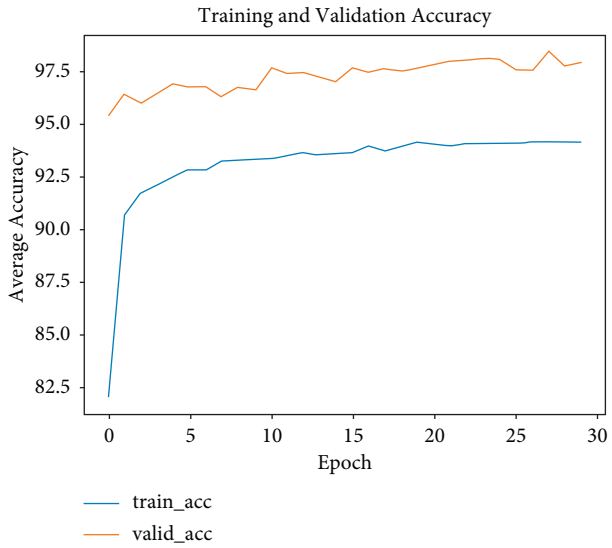


Figure 6: Accuracy of the training set and validation set.

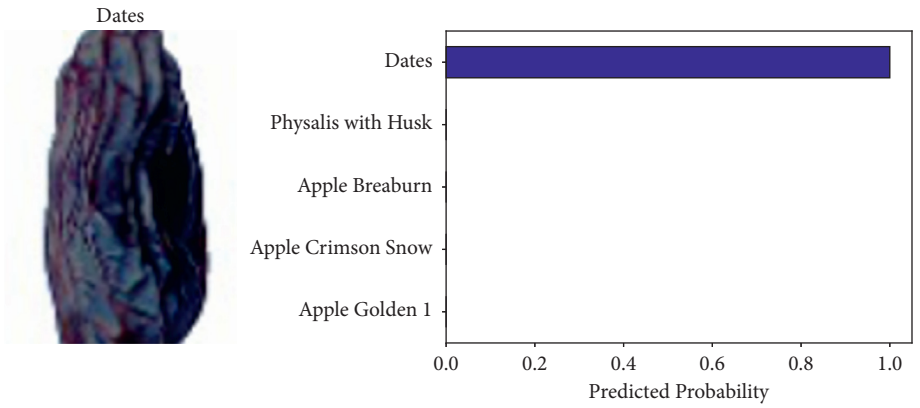


Figure 7: Dates top5 accuracy rate.

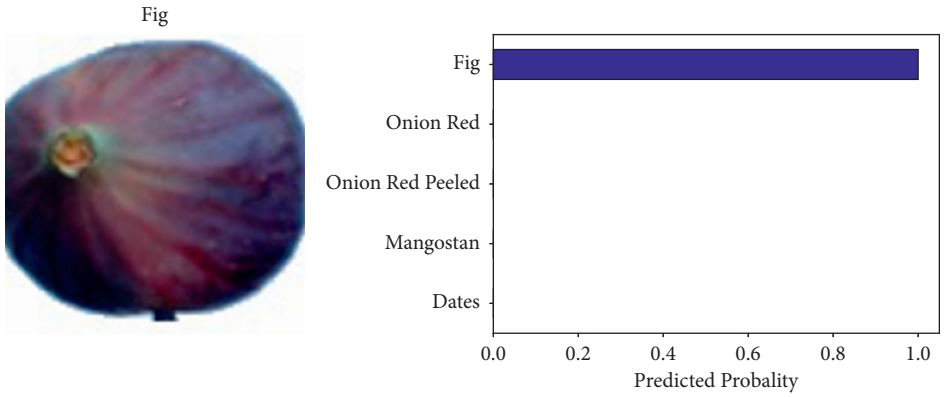


Figure 8: Fig top5 accuracy rate.

Table 6: Part of the test results in the test data.

Class	top1 (%)	top5 (%)	Loss
Apple braeburn	100.0	100.0	$1.5e-02$
Apple crimson snow	98.7	100.0	$7.9e-02$
Apple golden 1	100.0	100.0	$3.2e-07$
Apple golden 2	100.0	100.0	$3.7e-03$
Apple golden 3	84.0	100.0	$4.9e-01$

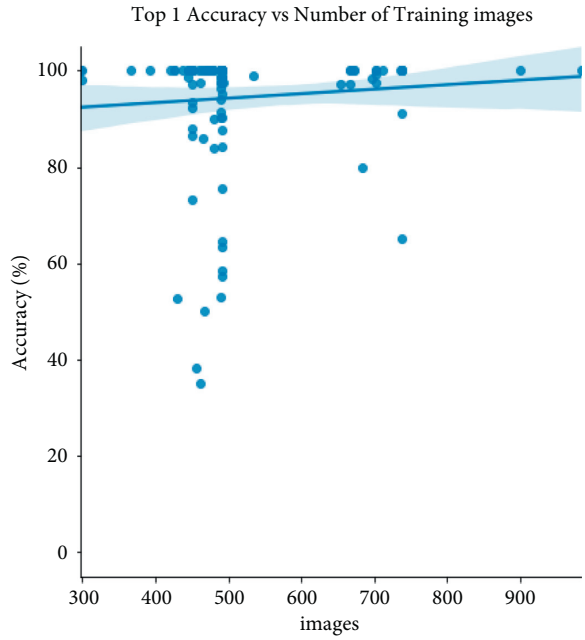


Figure 9: The number and accuracy of top1 in the training set.

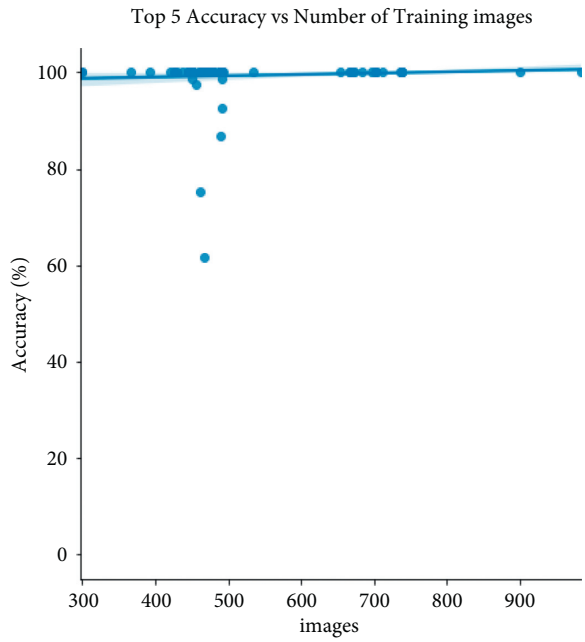


Figure 10: The number of images in the training set and the accuracy of the top5.

5. Conclusion

Aiming at the time-consuming and labor-intensive problems of traditional fruit and vegetable sorting and identification, a multicategory fruit and vegetable identification model based on transfer learning are proposed. First, in order to ensure the diversity of data and improve the generalization ability of the model, we performed data enhancement on the original data and performed random adjustment, cropping, and horizontal flipping operations, respectively. This allows the network to see more different data and improve the recognition rate of the model.

Second, in order to save the training cost, the parameters on the ImageNet dataset are transferred to the improved VGG network model in this paper, which speeds up the training time. Because we freeze the first 12 layers of VGG16, the network parameters of the first 12 layers have been trained on the ImageNet dataset for multiple rounds, and only the last few layers are mainly trained, which can ensure that there is a relatively high level at the beginning of training (recognition accuracy).

Finally, the classic VGG16 network is improved, the last 1000-dimensional fully connected layer and Softmax layer are deleted, two layers of 256-dimensional and 131-dimensional fully connected layers are added, and the LogSoftmax function is used to replace Softmax because we believe that deeper networks will extract higher-level features, which will ensure higher recognition accuracy. Compared with Softmax, LogSoftmax will better ensure the stability of data and prevent overflow.

The research presented in this paper also has some limitations. These limitations could form the basis of future research work. The following are the identified constraints and related directions for future work:

(1) This paper directly conducts classification experiments on 131 types of fruits, but there are inevitably errors in identification, because the gap between some fruits is extremely small. Apple, for example, has 13 different types of subcategories. Next, you can first perform the division prediction of large categories and then perform the prediction of subcategories under the same category, which may have a better recognition effect.

(2) Considering the hardware and computing costs, this paper does not train a brand new neural network from 0, which can be used as the next method to improve the classification accuracy.

(3) The work in this paper only considers the recognition and classification work under the single-target situation and does not consider the training work under the multiobjective situation. In practical scenarios, the recognition of multiobjective tasks may be more general and more meaningful.

(4) In the future, considering the problems of model landing and deployment, the model should be miniaturized to improve the model recognition accuracy and operation efficiency.

Data Availability

The public dataset can be obtained from https://github.com/Horea94/Fruit-Images-Dataset/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The study was supported by 2019 Guangxi Basic Research Ability Improvement Project for Young and Middle-Aged University Teachers (2019KY0640).

References

[1] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for IoV in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[2] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He, and X. Zhou, "TripRes," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 2, pp. 1–21, 2021.

[3] B. Shen, X. Xu, L. Qi, X. Zhang, and G. Srivastava, "Dynamic server placement in edge computing toward internet of vehicles," *Computer Communications*, vol. 178, pp. 114–123, 2021.

[4] X. Xu, H. Li, W. Xu, Z. Liu, L. Yao, and F. Dai, "Artificial intelligence for edge service optimization in internet of vehicles: a survey," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 270–287, 2022.

[5] Z. Hu, X. Xu, Y. Zhang et al., "Cloud-edge cooperation for meteorological radar big data: a review of data quality control–edge cooperation for meteorological radar big data: a review of data quality control," *Complex & Intelligent Systems*, pp. 1–15, 2021.

[6] W. Kong and B. Wang, "Combining trend-based loss with neural network for air quality forecasting in internet of Things," *Computer Modeling in Engineering and Sciences*, vol. 125, no. 2, pp. 849–863, 2020.

[7] X. Lu and H. Zhang, "An emotion analysis method using multi-channel convolution neural network in social networks," *Computer Modeling in Engineering and Sciences*, vol. 125, no. 1, pp. 281–297, 2020.

[8] R. Girshick, J. Donahue, and T. Darrell, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, IEEE, Columbus, Ohio, November, 2014.

[9] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, June, 2015.

[10] S. Q. Ren, K. M. He, and R. Girshick, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2015.

[11] J. Redmon, S. Divvala, and R. Girshick, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*, pp. 779–788, IEEE, USA, June, 2016.

[12] W. Zhou, X. Yingruo, Intelligent control system for fruit classification based on PLC and image processing," *Journal of Agricultural Mechanization Research*, vol. 43, no. 05, pp. 235–239, 2021.

[13] L. Zhu, "Recognition and application of fruit classification based on K - means clustering algorithms," *Journal of Agricultural Mechanization Research*, vol. 42, no. 08, pp. 46–50, 2020.

[14] G. Qin and M. Qin, "Application of visual capture picking robot in fruit classification system," *Journal of Agricultural Mechanization Research*, vol. 42, no. 09, pp. 212–216, 2020.

[15] Y. Song, C. A. Glasbey, G. W. Horgan, G. Polder, J. A. Dieleman, and G. W. A. M. van der Heijden, "Automatic fruit recognition and counting from multiple images," *Biosystems Engineering*, vol. 118, pp. 203–215, 2014.

[16] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "Deepfruits: a fruit detection system using deep neural networks," *Sensors*, vol. 16, p. 8, 2016.

[17] A. Selvaraj, N. Shebiah, S. Nidhyananthan, and L. Ganesan, "Fruit recognition using color and texture features," *Journal of Emerging TRends in Computing and Information Sciences*, vol. 1, no. 10, pp. 90–94, 2010.

[18] H. Zawbaa, M. Abbass, M. Hazman, and A. E. andHassanien, "Automatic fruit image recognition system based on shape and color features," *Communications in Computer and Information Science*, vol. 488, no. 11, pp. 278–290, 2014.

[19] D. Li, H. Zhao, X. Zhao, Q. Gao, and L. Xu, "Cucumber detection based on texture and color in greenhouse," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 31, no. 01, 2017.

[20] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," *CoRR abs/*, vol. 1507, p. 06228, 2015.

[21] J. Xiong, Z. Liu, R. Lin et al., "Green grape detection and picking-point calculation in a night-time natural environment using a charge-coupled device (ccd) vision sensor with artificial illumination," *Sensors*, vol. 18, p. 4, 2018.

[22] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[23] R. Sharma, A. Agarwal, and H. R. Mamatha, "Classification of carrots based on shape analysis using machine learning techniques," in *Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 1407–1411, IEEE, Tirunelveli, India, February, 2021.

[24] Z. A. Haq and Z. A. Jaffery, "Impact of activation functions and number of layers on the classification of fruits using CNN," in *Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 227–231, IEEE, New Delhi, India, March, 2021.

[25] M. Rahimzadeh and A. Attar, "Detecting and counting pistachios based on deep learning," *Iran Journal of Computer Science*, vol. 5, pp. 1–13, 2021.
[26] R. Siddiqi, "Fruit-classification model resilience under adversarial attack," *SN Applied Sciences*, vol. 4, no. 1, pp. 1–22, 2022.
[27] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. F. Schmid-huber, "High performance convolutional neural networks for image classification," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 1237–1242, AAAI Press, Barcelona, Catalonia, Spain, July, 2011.
[28] H. Muresan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Universitatis Sapientiae, Informatica*, vol. 10, pp. 26–42, 2018.
[29] J. Schmidhuber, "Deep learning in neural networks: an overview," *CoRR abs/*, vol. 1404, p. 7828, 2014.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 4967775, 16 pages
https://doi.org/10.1155/2022/4967775

Research Article

A Novel Self-Adaptive Mixed-Variable Multiobjective Ant Colony Optimization Algorithm in Mobile Edge Computing

Yiguang Gong,[1] **Weixue Wang**,[1] **and Siqi Gong**[2]

[1]*School of Automation, Nanjing University of Information Science & Technology, Nanjing 210000, China*
[2]*School of Water Resources and Hydropower Engineering, WuHan University, Wuhan 430000, China*

Correspondence should be addressed to Yiguang Gong; yiguang-gong@nuist.edu.cn

Received 15 May 2021; Revised 19 October 2021; Accepted 21 January 2022; Published 8 March 2022

Academic Editor: Xuyun Zhang

Mobile edge computing (MEC) provides physical resources closer to end users, becoming a good complement to cloud computing. The booming MEC brings many multiobjective optimization problems. The paper proposes a multiobjective optimization (MOO) algorithm called $SAMOACO_{MV}$, which provides a new choice for solving MOO problems of MEC. We improve the ACO_{MV} algorithm that is only suitable for solving mixed-variable single-objective optimization (SOO) problems and propose a $MOACO_{MV}$ algorithm suitable for solving mixed-variable MOO problems. And aiming at the dependence of $MOACO_{MV}$ algorithm performance on parameter setting, we proposed the $SAMOACO_{MV}$ algorithm using a self-adaptive parameter setting scheme. Furthermore, the paper also designs some mixed-variable MOO benchmark problems for the purpose to test and compare the performance of the $SAMOACO_{MV}$ algorithm. The experiments indicate that the $SAMOACO_{MV}$ algorithm has excellent comprehensive performance and is an ideal choice for solving mixed-variable MOO problems.

1. Introduction

In recent years, mobile edge computing (MEC), as a powerful computing paradigm, provides sufficient computing resources for the internet of things (IoT) [1]. Edge computing extends traditional cloud services to the edge of the network and closer to users and is suitable for network services with low latency requirements. There are many multiobjective optimization (MOO) problems in MEC, and the research on MOO for MEC is also a hot topic. Liu et al. [1] propose a multiobjective resource allocation method, named MRAM, and the method is leveraged to optimize the time cost of IoT applications, load balance, and energy consumption of MEC servers. Huang et al. [2] present a multiobjective whale optimization algorithm (MOWOA) based on time and energy consumption to solve the optimal offloading mechanism of computation offloading in MEC. Fan et al. [3] propose an algorithm based on particle swarm optimization (PSO) to solve the MOO of the container-based microservice scheduling, aiming to optimize network latency among microservices, reliability of microservice applications, and load balancing of the cluster.

Xu et al. [4] present a multiobjective computation offloading method (MOC) for internet of vehicles (IoV) in MEC to realize the multiobjective optimization of decreasing the load balancing rate and reduce the energy consumption in ECDs and shorten the time during processing the computing tasks.

This paper studies the multiobjective optimization algorithm, which provides a new choice for MOO in MEC. The classic MOO algorithm converts the multiple objective function values into a single value according to certain rules and then applies single-objective optimization algorithms to solve them [5]. There are three common converting rules [6]: weighted sum of multiple objective function values, calculating the distance between the objective function value vector and a given decision vector and finding the maximum value of the relative difference between the respective objective function values and their corresponding given values. The classic MOO algorithm is essentially a single-objective optimization algorithm, which cannot really solve the MOO

problem. Most of the modern MOO algorithms are heuristic algorithms that can find the Pareto solution set. Some famous algorithms are NSGA-II [7], SPEA2 [8], PAES [9], and NSGA-III [10] based on an evolutionary algorithm; SMPSO [11] and OMOPSO [12] based on particle swarm algorithm; GDE3 [13], MOEAD [14], and MOEA/D-IEpsilon [15] based on differential evolution algorithm; MOACO [16], P-ACO [17], MACS [18], Monaco [19], and SACO [20] based on ant colony algorithm; and so on. Other heuristic MOO algorithms include: MOO algorithms based on simulated annealing, tabu search and immune algorithms, and new algorithms obtained by improving or mixing various algorithms. According to the no free lunch (NFL) theorems in [21], when dealing with MOO problems, the average performance of various algorithms is the same, but the algorithms can show different performances for different optimization problems. Therefore, it is another hot spot for scholars to study the applicable algorithms for specific optimization problems or to study the applicable problems according to the characteristics of optimization algorithms.

Refer to the literature [22] for the classification of optimization problems, MOO problems can be divided into four categories according to whether their variable domains are continuous or not:

(i) Continuous-variable (CV) MOO: the range of all variables is the continuous domain. These continuous variables are usually mapped to real numbers

(ii) Pseudo-discrete variable (PDV) MOO: the range of all variables is ordered discrete domain, which means that the variable values are arranged in ascending or descending order according to certain rules. The pseudo-discrete variables are usually mapped to integers.

(iii) Real-discrete-variable (RDV) MOO: the range of all variables is a disordered discrete domain, which means that the variable values cannot be arranged according to certain rules. The discrete variables are usually called categorical variables.

(iv) Mixed-variable MOO: the range of the variables includes continuous domain and discrete domain.

According to the NFL theorem, in order to obtain better optimization performance, different types of MOO problems should use different types of optimization algorithms. The research on continuous-variable MOO and pseudo-discrete variable MOO is relatively mature. Most of the aforementioned heuristic algorithms or their variants are suitable for solving these two types of problems. There are a few studies on mixed-variable MOO.

Manson et al. [23] present a novel Bayesian multiobjective algorithm (MVMOO) capable of simultaneously optimizing both discrete and continuous input variables. The algorithm utilizes Gaussian processes as surrogates in combination with a novel distance metric based upon Gower similarity. MVMOO was able to perform competitively when compared to NSGA-II with a substantially reduced experimental budget, providing a viable, efficient option when optimizing expensive mixed-variable multiobjective optimization problems.

Li et al. [24] propose an improved version of OLAR-PSO-d named OLAR-PSO-DE. The OLAR-PSO-DE utilizes a modified stagnation strategy and a dynamic hybridization strategy. The OLAR-PSO-DE is employed to optimize the design of the engine hood, which is a high-dimensional, multiobjective, and mixed-variable optimization problem. The comparative study and final hood optimization results prove that the proposed method can effectively solve complicated engineering problems.

Khokhar et al. [25] modify the continuous-variable version of the PSP algorithm to handle mixed variables. The performance of PSP was tested using a set of quality indicators with a benchmark test suite. And the performance was compared with the state-of-the-art multiobjective optimization algorithms. The modified PSP is found to be competitive when the total number of function evaluations is limited but faces an increased computational challenge when the number of design variables increases.

However, there are relatively few studies on discrete variable MOO and mixed-variable MOO, but such MOO problems are often encountered in engineering. Therefore, the research on these two types of MOO algorithms is of great significance. This paper proposes the $\text{SAMOACO}_{\text{MV}}$ algorithm by improving the ACO_{MV} algorithm [26]. The main work of the author is as follows:

(i) Improve the ACO_{MV} algorithm used to solve mixed-variable SOO problems to make it suitable for solving mixed-variable MOO problems

(ii) Propose a self-adaptive parameter setting scheme for the algorithm and verify the superiority of the self-adaptive parameter setting scheme by comparison with the manual parameter adjustment scheme

(iii) Design some mixed-variable MOO benchmark problems to test and compare the performance of the $\text{SAMOACO}_{\text{MV}}$ algorithm

(iv) Apply $\text{SAMOACO}_{\text{MV}}$ algorithm to solve spring design engineering problems and compare the algorithm performance with other well-known MOO algorithms

2. Materials and Methods

2.1. ACO_{MV} Algorithm. The ACO_{MV} algorithm [26] is an ant colony optimization algorithm proposed by K. Socha and M. Dorigo for solving mixed-variable problems. The algorithm has excellent comprehensive performance when dealing with mixed-variable optimization problems, but for pure continuous optimization or pure discrete optimization, it has weaker performance than some specialized algorithms.

The basic process of the ACO_{MV} algorithm is as follows: the first step is to initialize the solution archive by randomly creating some solutions and storing them in the solution archive. In the second step, the ants construct some new solutions based on the solution archive. Many algorithms, such as local search, gradient descent, can be used to construct and improve the quality of new solutions. The

third step is to refresh the solution archive with the new solutions, and the best solutions will be stored in the solution archive. Repeat steps 2 and 3 until the termination criteria are met.

2.1.1. The Structure, Initialization, and Refresh of Solution Archive. $\mathrm{ACO_{MV}}$ maintains a solution archive T, whose dimension $|T| = k$ can be set in advance. Assume that there is an n-dimensional continuous optimization problem that has k feasible solutions, $\mathrm{ACO_{MV}}$ stores n variable values of each feasible solution and its objective function value in the solution archive. Figure 1 depicts the structure of the solution archive, where s_j^i represents the value of the i-th variable of the j-th solution and w_j represents the weight of the j-th solution. The solutions in the solution archive are sorted by their quality (such as the value of the objective function), so the position of the solution in the archive reflects its preference (pheromone).

Before the algorithm starts, k solutions are randomly generated and stored in the solution archive T. In each iteration of the algorithm, m ants generate m new solutions. The new solutions and the solutions from the solution archive T form a solution set including $k+m$ solutions and take the k solutions with the best quality (such as objective function value) from the solution set to refresh solution archive T. The solutions in the solution archive are always sorted by their quality, and the best quality solution is at the top. In this way, the search process will always tend to find the best quality solution, so as to achieve the solution of the optimization problem.

2.1.2. Constructing New Solutions Probabilistically. Each ant constructs a new solution incrementally, that is, selects the value of the solution variable one by one. First, the ants select a solution from the solution archive based on the selection probability. The selection probability of the j-th solution is as follows:

$$P_j = \frac{\omega_j}{\sum_{r=1}^{k} \omega_r}, \tag{1}$$

where ω_j can be calculated by using various formulas. In this paper, the Gaussian function $g(\mu, \sigma) = g(1, qk)$ is selected, which formula is as (2). Besides, q is the algorithm parameter, and k is the number of solutions in the solution archive.

$$\omega_j = \frac{1}{qk\sqrt{2\pi}} e^{\left(-(j-1)^2/2q^2k^2\right)}. \tag{2}$$

Then, construct a new solution based on the selected solution. According to the probability density function $P(x)$ for each dimension variable of the solution, the ant probabilistically extracts a new value in the neighborhood of the variable value of the solution, and these new values form a new solution. For different types of variables, the structure of the probability density function is different.

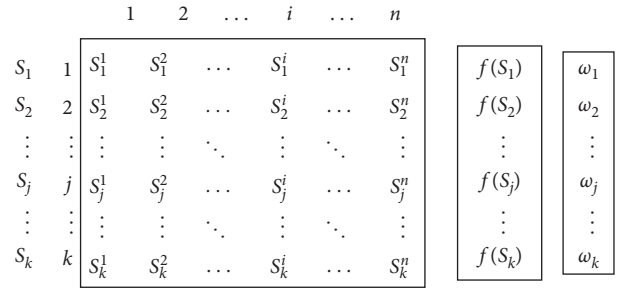


Figure 1: The structure of solution archive.

The $P(x)$ of continuous variables is as follows:

$$P(x) = g(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-(x-\mu)^2/2\sigma^2\right)}, \tag{3}$$

$$\mu = s_j^i \sigma = \xi \sum_{r=1}^{k} \frac{\left|s_r^i - s_j^i\right|}{k-1}, \tag{4}$$

where $g(x, \mu, \sigma)$ represents the Gaussian function with the variable x, μ is the mean value, σ is the mean square error, and ξ is the algorithm parameter.

The $P(x)$ of ordered discrete variables is the same as (3), but it needs to be modified as follows:

(i) The variable x is the index number of the ordered discrete variable value in its range. If the variable value range of x is {large, medium, small}, then $x = 1$ when the variable value is "large", $x = 2$ when the variable value is "medium", and $x = 3$ when the variable value is "small".

(ii) The new value obtained by probability extraction according to $P(x)$ needs to be rounded to the closest value of the index number in the domain. If the extracted value is 2.3, it needs to be rounded to 2, which corresponds to "middle."

The probability density function of disordered discrete variables is as follows:

$$O_l^i = \frac{w_l}{\sum_{r=1}^{c} w_r}, \tag{5}$$

where O_l^i represents the probability of selecting the l-th variable value from the domain $D_i = \left\{v_1^i, \ldots, v_{c_i}^i\right\}$ of the i-th variable of the solution. And ω_l is the weight associated with the l-th available value; it is calculated as follows:

$$w_l = \frac{\omega_{jl}}{u_l^i} + \frac{q}{\eta}, \tag{6}$$

where ω_{jl} is the weight corresponding to the best quality solution in the solution archive whose value of the dimension variable is not empty, and it can calculate as (2). In particular, if this dimension variable of all solutions is empty, then ω_{jl} is taken as 0. u_l^i is the number of solutions whose value of this dimension variable is not empty in the solution archive. q is an algorithm parameter, which is the same as q in (2). η is the number of unused values in the domain D_i of the dimension variable.

2.2. $MOACO_{MV}$ Algorithm. Improve the single-objective optimization algorithm ACO_{MV} and obtain the $MOACO_{MV}$ algorithm suitable for solving MOO problems. The main improvement of the $MOACO_{MV}$ algorithm is to introduce the Pareto set into the solution archive, which is the non-inferior solution set [27]. The specific method is to sort according to the Pareto characteristics of the solution in the solution archive, and the solution with the best quality is placed at the top of the solution archive. After improvement, the probability of selecting a good solution is higher so that the $MOACO_{MV}$ algorithm can find non-inferior solutions.

The solutions in the solution archive are arranged according to the following two rules:

(i) The solutions in the solution archive are sorted according to the non-inferior order, and the solutions with the smaller order value are arranged at the top of the solution archive. Referring to reference [28], the definition of non-inferior order of the solution is in Definition 1.

(ii) For solutions with the same non-inferior order, they are sorted according to the degree of congestion of the solution, and the solution with a lower degree of congestion is ranked at the top of the solution archive. Referring to reference [9], the definition of the congestion degree of the solution is in Definition 2.

In the above two rules, the first rule ensures that the algorithm can find non-inferior solutions, and the second rule ensures that the distribution of these non-inferior solutions is as uniform as possible. The $MOACO_{MV}$ algorithm designed according to the above rules has excellent comprehensive performance.

Definition 1. Non-inferior order of one solution $NIO(S_j)$: In the solution set $T = \{S_1, \ldots S_j \ldots S_k\}$, take out its non-inferior solutions to form a solution set *TU(z)* whose sequence number $z = 0$, and the remaining solutions refresh the solution set T; repeat the above process until T is an empty set, and every time it is repeated, z increases by 1. Then the $NIO(S_j)$ is the sequence number z of the non-inferior solution set *TU(z)* in which the solution S_j is.

Definition 2. Congestion degree of the solution $C\ D(S_x)$: In the solution set $T = \{S_1, \ldots S_i \ldots S_k\}$, the objective function corresponding to the solution S_j is $F(S_j) = (f_1(S_j) \ldots f_i(S_j) \ldots f_v(S_j))$. Calculate the distance between $F(S_x)$ of one solution S_x and $F(S_y)$ of other solutions and take out the minimum distance $d(x, T)$. The calculation process of $d(x, T)$ is as equation (1). Then the $C\ D(S_x)$ is $d(x, T)$ multiplied by the adjustment coefficient α; the calculation process is as equation (2).

$$d(x, T) = \min_{y \in T, y \neq x} \left\| F(S_x) - F(S_y) \right\|^2, \quad CD(S_x) = \alpha \times d(x, T). \tag{7}$$

2.3. $SAMOACO_{MV}$ Algorithm. The ant colony algorithm needs to set some parameters, which have a huge impact on the performance of the algorithm. Since the convergence speed of the algorithm and the diversity of the solution are always contradictory, how to obtain a compromised excellent performance through proper parameter settings is the purpose of studying parameter settings.

This paper adopts the self-adaptive parameter control method to adjust the parameters of the $MOACO_{MV}$ algorithm according to the quality of the solution archive and the convergence speed of the algorithm. And we call this MOO algorithm as $SAMOCO_{MV}$ algorithm.

The $SAMOACO_{MV}$ algorithm needs to set four parameters, which are: the convergence speed ξ, the size of search solution archiving area q, the number of ants m, and the solution archive size k. In this paper, to balance the diversity and convergence abilities of $SAMOACO_{MV}$, two modifications for four parameters are proposed.

2.3.1. Set Method for Parameters ξ and q. The parameter ξ is used to adjust the convergence speed of the algorithm, and the parameter q is used to change the size of the search area. These two parameters are in conflict. When the search area increases or the convergence speed decreases, more Pareto solutions can be found with higher probability, but the calculation time becomes longer, and vice versa. In order to obtain a good Pareto solution archive with reasonable calculation time, we calculate the quality index of the solution archive and adjust the parameter ξ and q according to the value of the quality index. The set method for parameters ξ and q is shown in Algorithm 1:

In Algorithm 1, the quality index $P_i(T)$ of the solution archive for the i-th iteration is calculated firstly. The $P_i(T)$ is the mean value of the weighted sum of each objective function and congestion degree of all solutions in the solution archive. Next, the quality index's increment $\Delta P_i(T)$ and the parameters' increment Δe_i, Δq_i is calculated. Finally, the new parameter values are set by subtracting the product of $\Delta\xi_i$, $\Delta\ q_i$, step size constant B, and random number r from the old parameter values.

2.3.2. Set Method for Parameters m and k. The parameter m is the number of ants, and the parameter k is the solution archive size. The larger the values of these two parameters are, the higher the probability of obtaining more Pareto solutions is, but the larger parameters' value will also bring more calculations and increase time-consuming. We set the expected number of Pareto solutions according to the complexity of the problem and then adjust these two parameters in real time according to the difference between *ENUM* and the actual number of Pareto solutions. *ENUM* is the number of non-inferior solutions expected from the solution archive. The set method for parameters m and k is shown in Algorithm 2.

In Algorithm 2, count the number of solutions whose non-inferior order $NIO_i(S_j)$ are zero in the solution archive. Then calculate the ratio factors *rateArchive* and *rateAnt*, which represent the size of the solution archive and the

Input: $F_i(S_j)$, $CD_i(S_j)$, ξ_i, q_i, where $i \in (1, l-1)$, $j \in (1, k)$;
(1) $P_i(T) = (1/k)\sum_{j\in(1,k)}(\beta \cdot F_i(S_j) + \gamma CD_i(S_j))$
(2) $\Delta P_i(T) = P_i(T) - P_{i-1}(T)$
(3) $\Delta\xi_i = \xi_i - \xi_{i-1}$
(4) $\Delta q_i = q_i - q_{-1}$
(5) $\xi_{i+1} = \xi_i - r * B * \Delta P_i(T) * \Delta\xi_i$
(6) $q_{i+1} = q_i - r * B * \Delta P_i(T) * \Delta q_i$

Algorithm 1: Set method for parameters ξ and q.

Input: $NIO_i(S_j)$, m_i, k_i, where $i \in (1, l-1)$, $j \in (1, k)$;
(1) **for** $j = 1$ to k **do**
  if $NIO_i(S_j) == 0$**then**
  num++;
(2) $rateArchive = k_i/num$;
(3) $rateAnt = m_i/num$;
(4) $k_{i+1} = C * rateArchive * ENUM$;
(5) $m_{i+1} = C * rateAnt * ENUM$;

Algorithm 2: Set method for parameters m and k.

number of ants needed to produce one non-inferior solution, respectively. Finally, set the new parameters' value to the product of the old parameters' value, the ratio factors, adjustment coefficient C, and expected number $ENUM$.

3. Experiment Results and Discussion

The application field and performance of the algorithm are usually studied by comparing the performance of different MOO algorithms when solving benchmark problems. Referring to some existing mixed-variable MOO algorithms [29–33], this paper designs some problems for algorithm experiments, besides comparing with other well-known MOO algorithms to verify the performance of the algorithm.

3.1. Experimental Environment. The operating environment of the experiment is as follows: Thinkpad T470p computer; Core i7-7700HQ CPU (4cores) * 2; 24 GB memory; 512 GB solid hard disk; and equipped with Windows 10 operating system. The programming tool is Microsoft Visual Studio 2017, and the programming language is C#.

3.2. Benchmark Problem. In this paper, we select eight well-known benchmark problems to evaluate MOO algorithms, that is, Schaffer, Fonseca, Kursawe, ZDT problems, Viennet2, and Viennet3 [34]. These benchmark problems have two (Schaffer, Fonseca, Kursawe, and ZDT family) or three objectives (Viennet2 and Viennet3), and they occupy different properties: separability, unimodality multimodality, convexity, linearity, non-convexity, continuity, discontinuity, bias, Pareto many-to-one, and so on.

The problem name, variable count(N), variable bounds, designed variables, and objective functions are shown in Table 1.

The variables of the eight benchmark problems are all continuous variables. In order to test MOO algorithms with mixed variables, we modify the problems to make some variables as PDV and some variables as RDV; then the continuous problems become mixed problems. PDV and RDV are calculated by the following equations:

$$x_{PDV} \in \left\{x | x_{\mathrm{MIN}} + I * \frac{x_{\mathrm{MAX}} - x_{\mathrm{MIN}}}{N}, \quad I = 0, \ldots, N\right\}, \tag{8}$$

$$x_{RDV} \in \left\{x | x_{\mathrm{MIN}} + I * \frac{x_{\mathrm{MAX}} - x_{\mathrm{MIN}}}{N}, I = RND(N)\right\}, \tag{9}$$

where N is the number of equal divisions of the value range. In order to make the variable a value of 0, N takes a positive even number. $RND(N)$ is a random nonnegative integer not greater than N. In order to make the distribution range of x_{RDV} larger, we need to take every number in $\{0, \ldots, N\}$ once. The domain of x_{PDV} is a set of $N+1$ ordered discrete variables increasing from x_{MIN} to x_{MAX}, and the domain of x_{RDV} is a set of $N+1$ disordered discrete variables between x_{MIN} and x_{MAX}.

If N is large enough, the Pareto set of the mixed problems is similar to the Pareto set of the continuous problems.

3.3. Performance Metrics. Convergence and diversity are usually the two most important criteria for the evaluation of MOO algorithms. The convergence refers to the distance from the non-dominated front generated by the

Table 1: Test benchmark problems.

Problem name	Variable count (N)	Variable bounds	Designed variables	Objective functions
Schaffer	1	$[-10^3, 10^3]$	x_1 is RDV	$f_1 = x^2,\ f_2 = (x-2)^2$
Fonseca	3	$[-4, 4]$	x_2 is PDV x_3 is RDV	$f_1 = 1 - \exp(-\sum_{i=1}^{3}(x_i - (1/\sqrt{3}))^2)$ $f_2 = 1 - \exp(-\sum_{i=1}^{3}(x_i + (1/\sqrt{3}))^2)$
Kursawe	3	$[-5,5]$	x_2 is PDV x_3 is RDV	$f_1 = \sum_{i=1}^{n-1}(-10\exp(-0.2 * \sqrt{x_i^2 + x_{i+1}^2}))\ f_2 = \sum_{i=1}^{n}(\lvert x_i\rvert^{0.8} + 5(\sin\ x_i)^3)$
ZDT1	4	$[0,1]$	x_3 is PDV x_4 is RDV	$f_1 = x_1,\ f_2 = g(x)(1 - \sqrt{f_1/g(x)}),\ g(x) = 1 + 9/(N-1)\sum_{i=2}^{N} x_i$
ZDT2	4	$[0,1]$	x_3 is PDV x_4 is RDV	$f_1 = x_1,\ f_2 = g(x)(1 - (f_1/g(x))^2),\ g(x) = 1 + 9/(N-1)\sum_{i=2}^{N} x_i$
ZDT3	4	$[0,1]$	x_3 is PDV x_4 is RDV	$f_1 = x_1,\ f_2 = g(x)(1 - \sqrt{f_1/g(x)} - f_1/g(x)\sin(10\pi f_1)),$ $g(x) = 1 + 9/(N-1)\sum_{i=2}^{N} x_i$
Viennet2	2	$[-4,4]$	x_2 is PDV	$f_1 = 1/2(x_1 - 2)^2 + 1/13(x_2 + 1)^2 + 3$ $f_2 = 1/36(x_1 + x_2 - 3)^2 + 1/8(x_2 - x_1 + 2)^2 - 17$ $f_3 = 1/175(x_1 + 2x_2 - 1)^2 + 1/17(2x_2 - x_1)^2 - 13$
Viennet3	2	$[-3,3]$	x_2 is PDV	$f_1 = 1/2(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2),$ $f_2 = 1/8(3x_1 - 2x_2 + 4)^2 + 1/27(x_2 - x_2 + 1)^2 + 15$ $f_3 = 1/(x_1^2 + x_2^2 + 1) - 1.1e^{-(x_1^2 + x_2^2)}$

optimization algorithm to the true Pareto front; the diversity involves coverage area and uniformity; and a front with wide coverage and good uniformity is always pursued.

We have used generational distance (GD) [35] and inverted generational distance plus (IGD^+) [36] for measuring convergence and spread for measuring coverage.

GD: let $T^* = \{F_1^*, \ldots F_i^* \ldots F_{|T^*|}^*\}$ be a set of uniformly distributed Pareto optimal points in the true PF(TPF), and $T = \{F_1, \ldots F_i \ldots F_{|T|}\}$ be a non-dominated front of the problems. The GD of T is the average distance from each solution in T to the nearest reference point:

$$GD(T^*, T) = \frac{1}{|T|}\sum_{j=1}^{|T|}\min_{F_i^* \in T^*} d_{GD}(F_i^*, F_j), \tag{10}$$

where F_i is the objective function corresponding to the solution S_i and $F = (f_1(S_i) \ldots f_i(S_i) \ldots f_v(S_i))$. $d_{GD}(F_i^*, F_j)$ is the Euclidean distance between F_j and F_i^*.

IGD^+: the IGD^+ of T is the average distance from each reference point in T^* to the nearest solution.

$$IGD^+(T^*, T) = \frac{1}{|T^*|}\sum_{i=1}^{|T^*|}\min_{F_j \in T} d_{IGD^+}(F_i^*, F_j). \tag{11}$$

In IGD^+, the distance between a reference point $F^* = (f_1^*, f_2^*, \ldots, f_v^*)$ and a solution $F = (f_1, f_2, \ldots, f_v)$ is calculated in the objective space for the v-objective minimization problem as follows:

$$d_{IGD^+}(F^*, F) = \sqrt{\sum_{i=1}^{v}(\max\{f_i - f_i^*, 0\})^2}. \tag{12}$$

Generalized Spread (see [36]). The generalized spread is an indicator that measures the distribution and spread of the obtained non-dominated front of the problems with two or more objectives:

$$\Delta(T^*, T) = \frac{\sum_{i=1}^{v} d(e_i, T) + \sum_{X \in T^*}|d(X, T) - \overline{d}|}{\sum_{i=1}^{v} d(e_i, T) + |T^*|\overline{d}}, \tag{13}$$

where $\{e_1, e_2, \ldots, e_m\}$ are m extreme solutions in T^* and

$$d(X, T) = \min_{Y \in T, Y \neq X}\|F(X) - F(Y)^2\|,$$
$$\overline{d} = \frac{1}{|T^*|}\sum_{X \in T^*} d(X, T). \tag{14}$$

3.4. Performance Improvement of SAMOACO$_{MV}$. In order to test the performance of SAMOACO$_{\mathrm{MV}}$, some experiments are carried out under the same conditions, for example, when the problem is the modified Fonseca problem, the maximum number of algorithm iterations is the same. Table 2 lists the setting schemes of the algorithm parameters in the six experiments. The first five experiments test the performance of the MOACO$_{\mathrm{MV}}$ algorithm that have different parameter values of ξ and q and the same values of m and k. The sixth experiment tests the performance of the SAMOCO$_{\mathrm{MV}}$ algorithm.

Table 3 shows the performance of the 6 experiments for the Fonseca test problem. For each major cell of Table 3, the first column indicates the mean of 25 runs, the second column indicates the standard deviation, and the third column indicates the rank.

Figure 2 shows the Pareto points obtained with reference to the true Pareto frontier graphically using results from 1 of the 25 runs. MOACO$_{\mathrm{MV5}}$ generates only a few Pareto points, so it is not shown in the figure.

It can be seen from the figure and the table:

(i) The figure shows that the Pareto points generated by the SAMOACO$_{\mathrm{MV}}$ algorithm are right on TPF, and the table shows the overall rank value of the

Table 2: Deployment of algorithms' parameters.

	m	k	ξ	q
$MOACO_{MV}1$	50	200	0.001	0.001
$MOACO_{MV}2$	50	200	0.01	0.01
$MOACO_{MV}3$	50	200	0.1	0.1
$MOACO_{MV}4$	50	200	1	1
$MOACO_{MV}5$	50	200	10	10
$SAMOACO_{MV}$	Self-adaptive control			

Table 3: The performance of six experiments for the Fonseca problem.

Fonseca	Generational distance			Inverted generational distance			Generalized spread			Sum of ranks	Overall rank
	Mean	Stdev	Rank	Mean	Stdev	Rank	Mean	Stdev	Rank		
$MOACO_{MV}1$	0.0167	0.0118	5	0.2153	0.0893	5	1.1378	0.1869	6	16	6
$MOACO_{MV}2$	0.0031	0.0014	4	0.1545	0.0872	4	0.9836	0.0605	5	13	4
$MOACO_{MV}3$	0.0001	1.1788	2	0.0096	0.0103	3	0.6524	0.0561	2	7	3
$MOACO_{MV}4$	0.0002	1.32E-05	3	0.0035	0.0004	2	0.5147	0.0282	1	6	1
$MOACO_{MV}5$	0.0815	0.0323	6	0.246	0.0788	6	0.6905	0.2014	3	15	5
$SAMOACO_{MV}$	9.45E-05	1.24E-05	1	0.0026	0.0003	1	0.7064	0.0384	4	6	1

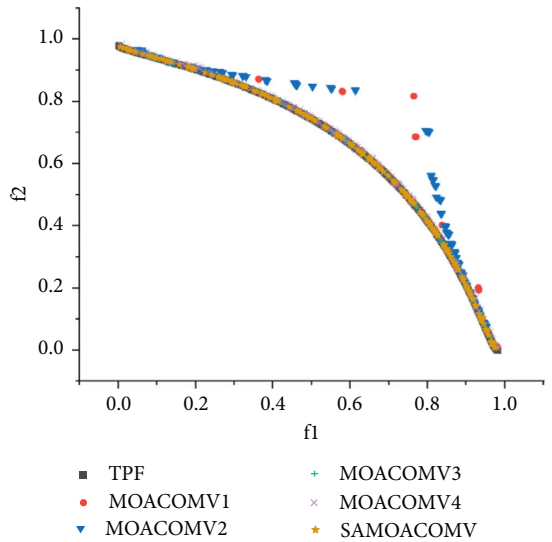


Figure 2: Pareto front for Fonseca problem.

$SAMOACO_{MV}$ algorithm is minimum, that is, the performance of the $SAMOACO_{MV}$ algorithm is the best in all experiments.

(ii) When the $MOACO_{MV}$ algorithm adopts setting schemes 3 and 4, the algorithm performance is basically the same as that of the $SAMOCO_{MV}$ algorithm, but when other schemes are used, the algorithm performance is very poor, which shows that the performance of the $MOACO_{MV}$ algorithm relies heavily on parameter settings.

More experiments show that the performance of the $SAMOCO_{MV}$ algorithm is better than that of the $MOACO_{MV}$ algorithm; especially, this advantage is more obvious when the values of m and k are small.

3.5. Performance Comparison Using Benchmark Problems. In order to test the performance of the algorithm, this paper compares the $SAMOCO_{MV}$ algorithms with the well-known MOO algorithm NSGAII, SPEA2, SMPSO, MOEAD, NSGAIII, and MOEA/D-IEpsilon. These algorithm programs come from jMetal [30], and the two algorithms can only be used to deal with CV MOO.

In order to compare the multiobjective optimization algorithms, each algorithm is allowed to run for the test problems for a constant number of function evaluations. The performance metrics are calculated for each algorithm run. This procedure is repeated for 20 runs, and the mean and standard deviation of the performance metrics are recorded for each algorithm.

3.5.1. Results Based on Schaffer, Fonseca, and Kursawe Problems. Tables 4–6 show the mean and standard deviation of generational distance, inverted generational distance plus, and generalized spread for different algorithms, respectively. The $SAMOACO_{MV}$ fetches good performance metric values in terms of the Schaffer problem, while other algorithms cannot obtain or only obtain a few Pareto points. It may be because the only variable of Schaffer problem is changed to a discrete variable, and other algorithms cannot solve the pure discrete variable problem. For the Fonseca and Kursawe problems, compared with other techniques, $SAMOACO_{MV}$ obtains excellent GD and IGD^{+} values, only slightly weaker than MOEA/D-IEpsilon, but obtains relatively poor generalized spread value.

Figures 3–5 provide a graphical visualization of the Pareto points obtained for Schaffer, Fonseca, and Kursawe problems, respectively. For the Schaffer problem, none of the other algorithms apart from $SAMOACO_{MV}$ was able to produce any Pareto points close to the TPF. For the Fonseca

Table 4: Mean and standard deviation of generational distance.

		Schaffer	Fonseca	Kursawe	ZDT1	ZDT2	ZDT3	Viennet2	Viennet3	Sum of ranks	Overall rank
NSGAII	Mean	—	0.0004	0.0005	0.0017	0.0029	0.0008	0.0008	0.0002	29	4
	Stdev	—	3.51E-05	8.89E-05	0.0003	0.0006	0.0002	0.0003	3.72E-05		
	Rank	2	4	2	5	5	3	4	4		
SPEA2	Mean	—	0.0004	0.0005	0.0021	0.0036	0.0017	0.0008	0.0003	39	6
	Stdev	—	4.82E-05	0.0001	0.001	0.0022	0.0018	0.0001	0.0001		
	Rank	2	4	2	7	7	7	4	6		
SMPSO	Mean	—	0.0004	0.0012	0.0011	0.0019	0.0009	0.001	0.0002	32	5
	Stdev	—	3.92E-05	0.0004	0.0001	0.0003	0.0002	0.0003	9.86E-05		
	Rank	2	4	6	3	3	4	6	4		
MOEAD	Mean	—	0.0003	0.0015	0.0014	0.0024	0.0011	—	—	39	6
	Stdev	—	4.00E-05	0.0007	0.0002	0.0003	0.0002	—	—		
	Rank	2	3	7	4	4	5	7	7		
NSGAIII	Mean	—	0.0004	0.0005	0.0019	0.0032	0.0012	1.77E-05	1.15E-05	28	3
	Stdev	—	3.82E-05	0.0001	0.0007	0.0009	0.0008	4.61E-06	4.77E-07		
	Rank	2	4	2	6	6	6	1	1		
MOEA/D-IEpsilon	Mean	—	9.93E-05	0.0002	0.0005	0.0004	0.0001	0.0003	4.37E-05	17	2
	Stdev	—	2.76E-06	1.79E-05	8.28E-05	3.33E-05	1.23E-05	7.56E-05	7.48E-06		
	Rank	2	2	1	2	2	2	3	3		
$SAMOACO_{MV}$	Mean	5.38E-05	9.68E-05	0.0005	0.0001	0.0002	8.25E-05	3.21E-05	2.12E-05	11	1
	Stdev	9.01E-05	3.23E-06	7.08E-05	6.29E-06	6.63E-06	4.86E-06	1.54E-06	3.03E-06		
	Rank	1	1	2	1	1	1	2	2		

Table 5: Mean and standard deviation of inverted generational distance plus.

		Schaffer	Fonseca	Kursawe	ZDT1	ZDT2	ZDT3	Viennet2	Viennet3	Sum of ranks	Overall rank
NSGAII	Mean	—	0.0048	0.0044	0.0156	0.0418	0.0078	0.0141	0.0052	37	5
	Stdev	—	0.0005	0.0005	0.0055	0.0674	0.003	0.0026	0.0006		
	Rank	2	6	3	7	5	4	6	4		
SPEA2	Mean	—	0.0049	0.0048	0.0150	0.0696	0.0086	0.0066	0.0041	39	6
	Stdev	—	0.0004	0.0005	0.0017	0.1054	0.0011	0.0005	0.0004		
	Rank	2	7	4	6	7	6	4	3		
SMPSO	Mean	—	0.0046	0.0123	0.0098	0.0128	0.0074	0.0135	0.0053	32	4
	Stdev	—	0.0002	0.0021	0.0008	0.0013	0.0010	0.0024	0.0006		
	Rank	2	4	7	3	3	3	5	5		
MOEAD	Mean	—	0.0042	0.008	0.0131	0.0188	0.0101	—	—	39	6
	Stdev	—	0.0004	0.0009	0.0016	0.0022	0.0019	—	—		
	Rank	2	3	5	4	4	7	7	7		
NSGAIII	Mean	—	0.0046	0.0089	0.0149	0.0622	0.0079	0.0005	0.0005	30	3
	Stdev	—	0.0004	0.003	0.0049	0.0944	0.0040	9.83E-06	3.64E-05		
	Rank	2	4	6	5	6	5	1	1		
MOEA/D-IEpsilon	Mean	—	0.0015	0.0031	0.0097	0.0058	0.0030	0.0062	0.0064	19	2
	Stdev	—	5.32E-05	0.0003	0.0017	0.0004	0.0002	0.0006	0.0007		
	Rank	2	1	1	2	2	2	3	6		
$SAMOACO_{MV}$	Mean	5.32E-06	0.0017	0.0040	0.0021	0.0029	0.0016	0.0007	0.0005	11	1
	Stdev	2.38E-06	7.78E-05	0.0003	6.90E-05	0.0001	0.0001	7.19E-06	4.73E-05		
	Rank	1	2	2	1	1	1	2	1		

Table 6: Mean and standard deviation of generalized spread.

		Schaffer	Fonseca	Kursawe	ZDT1	ZDT2	ZDT3	Viennet2	Viennet3	Sum of ranks	Overall rank
NSGAII	Mean	—	0.2297	0.3980	0.5673	0.6321	0.5606	0.4579	0.4094	22	1
	Stdev	—	0.0311	0.0383	0.0649	0.1363	0.0703	0.0356	0.0335		
	Rank	2	3	1	4	5	2	3	2		
SPEA2	Mean	—	0.1948	0.4531	0.6467	0.7259	0.7707	0.2058	0.5898	31	4
	Stdev	—	0.0184	0.0529	0.0637	0.1715	0.1301	0.0247	0.0207		
	Rank	2	1	2	6	7	7	1	5		
SMPSO	Mean	—	0.2306	0.8257	0.4907	0.5718	0.5416	0.3143	0.2296	22	1
	Stdev	—	0.0156	0.0610	0.0437	0.0966	0.0528	0.0276	0.0269		
	Rank	2	4	6	2	4	1	2	1		
MOEAD	Mean	—	0.2958	0.5525	0.4562	0.4961	0.6611	—	—	33	6
	Stdev	—	0.0416	0.0561	0.0465	0.0540	0.0484	—	—		
	Rank	2	5	4	1	1	6	7	7		
NSGAIII	Mean	—	0.3346	0.8579	0.5899	0.7028	0.626	0.6129	0.9181	42	7
	Stdev	—	0.0355	0.0783	0.0729	0.1438	0.1078	0.0146	1.23E-02		
	Rank	2	6	7	5	6	5	5	6		
MOEA/D-IEpsilon	Mean	—	0.2166	0.5719	0.8105	0.5029	0.5857	0.8131	0.5781	32	5
	Stdev	—	0.0107	0.0527	0.0957	0.0440	0.0470	0.0583	0.0934		
	Rank	2	2	5	7	2	4	6	4		
$SAMOACO_{MV}$	Mean	0.0958	0.4827	0.4747	0.5028	0.5278	0.5607	0.5403	0.4938	27	3
	Stdev	0.0006	0.0254	0.0447	0.0203	0.0166	0.0207	0.0099	0.0093		
	Rank	1	7	3	3	3	3	4	3		

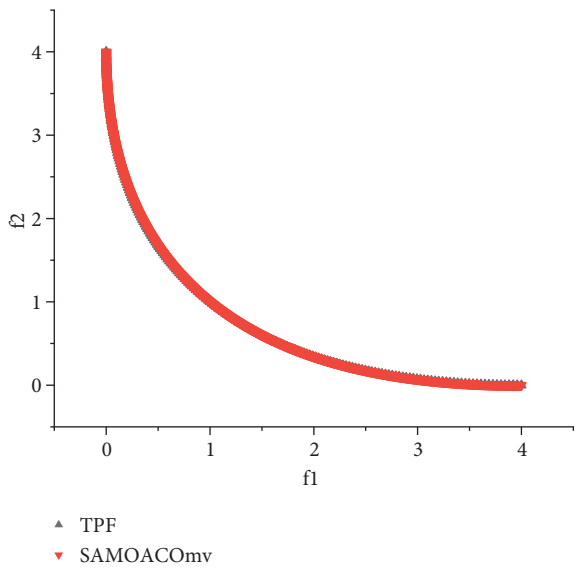


Figure 3: Pareto front for the Schaffer problem.

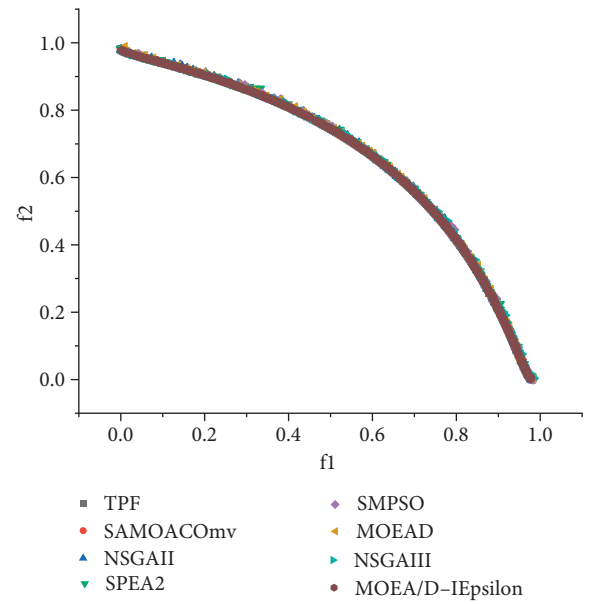


Figure 4: Pareto front for the Fonseca problem.

problem, the performance of each algorithm is very good, and the generated Pareto points right on TPF. For the Kursawe problem, the performance of each algorithm is also very good, except that some Pareto points generated by SMPSO and MOEAD deviate slightly from TPF.

3.5.2. Results Based on ZDT (ZDT1–ZDT3) Problems. From Tables 4 and 5, $SAMOACO_{MV}$ ranks 1 for *ZDT* problems, which means that the $SAMOACO_{MV}$ outperforms other algorithms on the performance metrics *GD* and IGD^{+}. From Table 6, $SAMOACO_{MV}$ performed slightly worse on generalized spread for *ZDT* problems, ranking 3.

From Figures 6–8, all the algorithms have good performance, and the obtained Pareto front is basically consistent with the TPF. Some algorithms do not perform well on certain problems, such as SPEA2 and MOEAD produce some points that deviate slightly from the TPF for *ZDT1* and *ZDT2* problems.

3.5.3. Results Based on Viennet2 and Viennet3 Problems. As shown in Table 4, the mean of *GD* of $SAMOACO_{MV}$ for Viennet2 and Viennet3 problems are about 0.000032 and

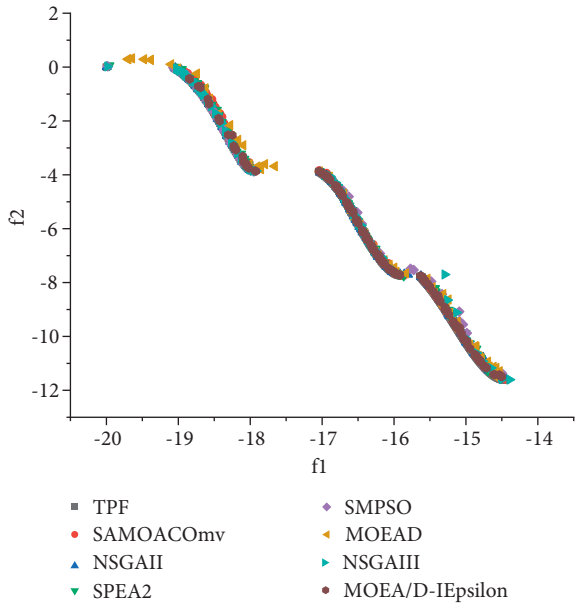


Figure 5: Pareto front for the Kursawe problem.

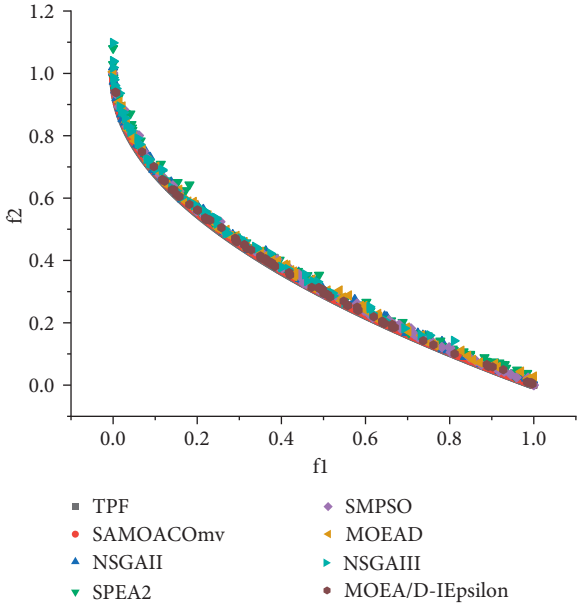


Figure 6: Pareto front for the ZDT1 problem.

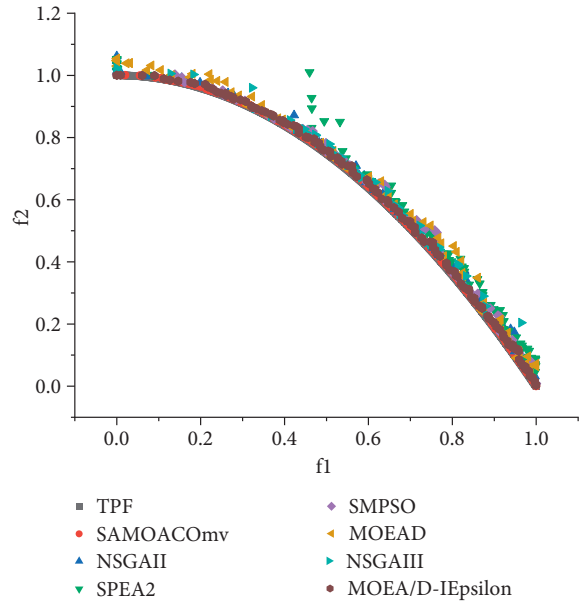


Figure 7: Pareto front for the ZDT2 problem.

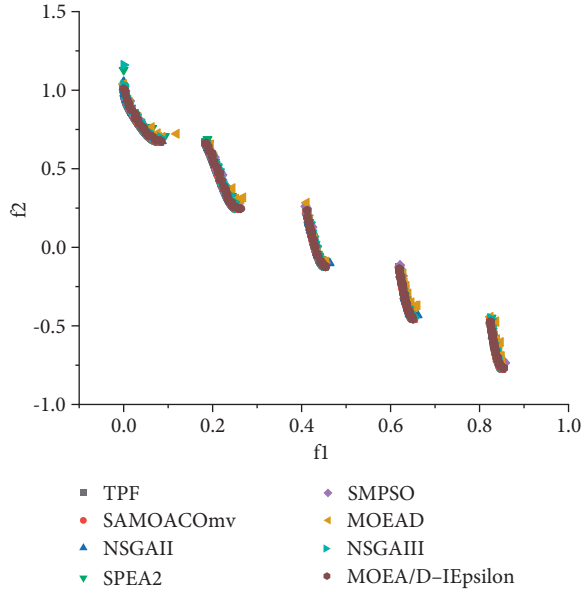


Figure 8: Pareto front for the ZDT3 problem.

0.000021, respectively, which are only slightly worse than the mean values of NSGAIII but far better than the corresponding performance metric values of other algorithms. It can be seen from Table 5 that similar to *GD*, the SAMOACO$_{MV}$ has almost the best IGD^+ mean for Viennet2 and Viennet3 problems, around 0.0007 and 0.0005, respectively, only slightly worse than the mean of NSGAIII. From Table 6, SAMOACO$_{MV}$ performed worse on generalized spread for Viennet2 and Viennet3 problems, ranking 4 and 3, respectively.

In Figures 9 and 10, the approximated Viennet2 and Viennet3 fronts of each algorithm are shown. It is clear that SAMOACO$_{MV}$ obtained much more Pareto points, they converge well to the TPF, and they widely and uniformly distribute along the TPF, which illustrates that it has better convergence and diversity compared with the other algorithms.

In summary, with *GD*, IGD^+, and generalized spread taken into consideration, SAMOACO$_{MV}$ is quite a competitive algorithm in terms of the convergence of the generated Pareto solution set; the overall rank is 1. But SAMOACO$_{MV}$ is slightly weaker than other algorithms in the coverage performance; the overall rank is 3.

4. Experiment Results on Spring Design Problem

The spring design problem is a common engineering practice problem and widely used MOO algorithm

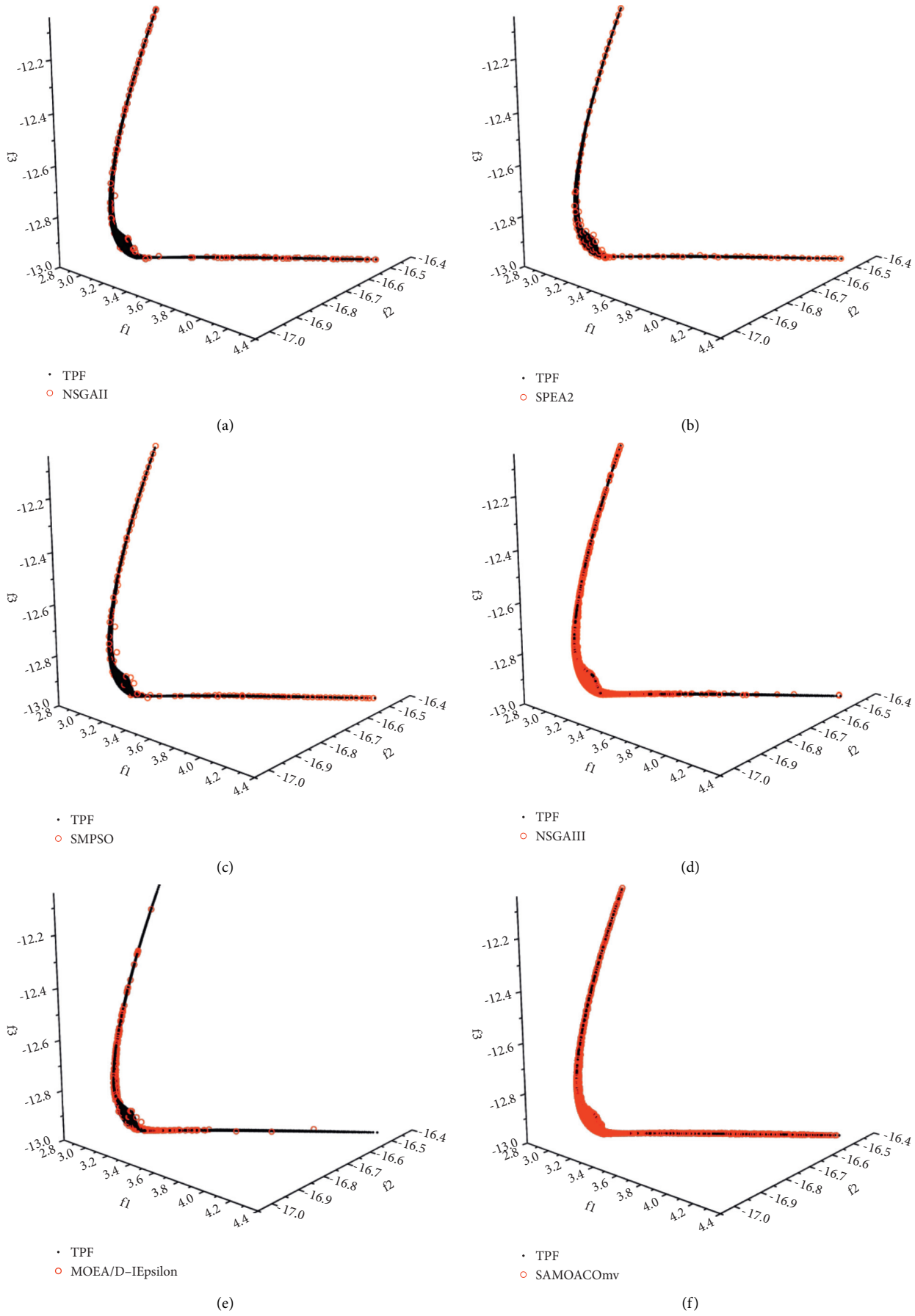


Figure 9: Pareto front for the Viennet2 problem.

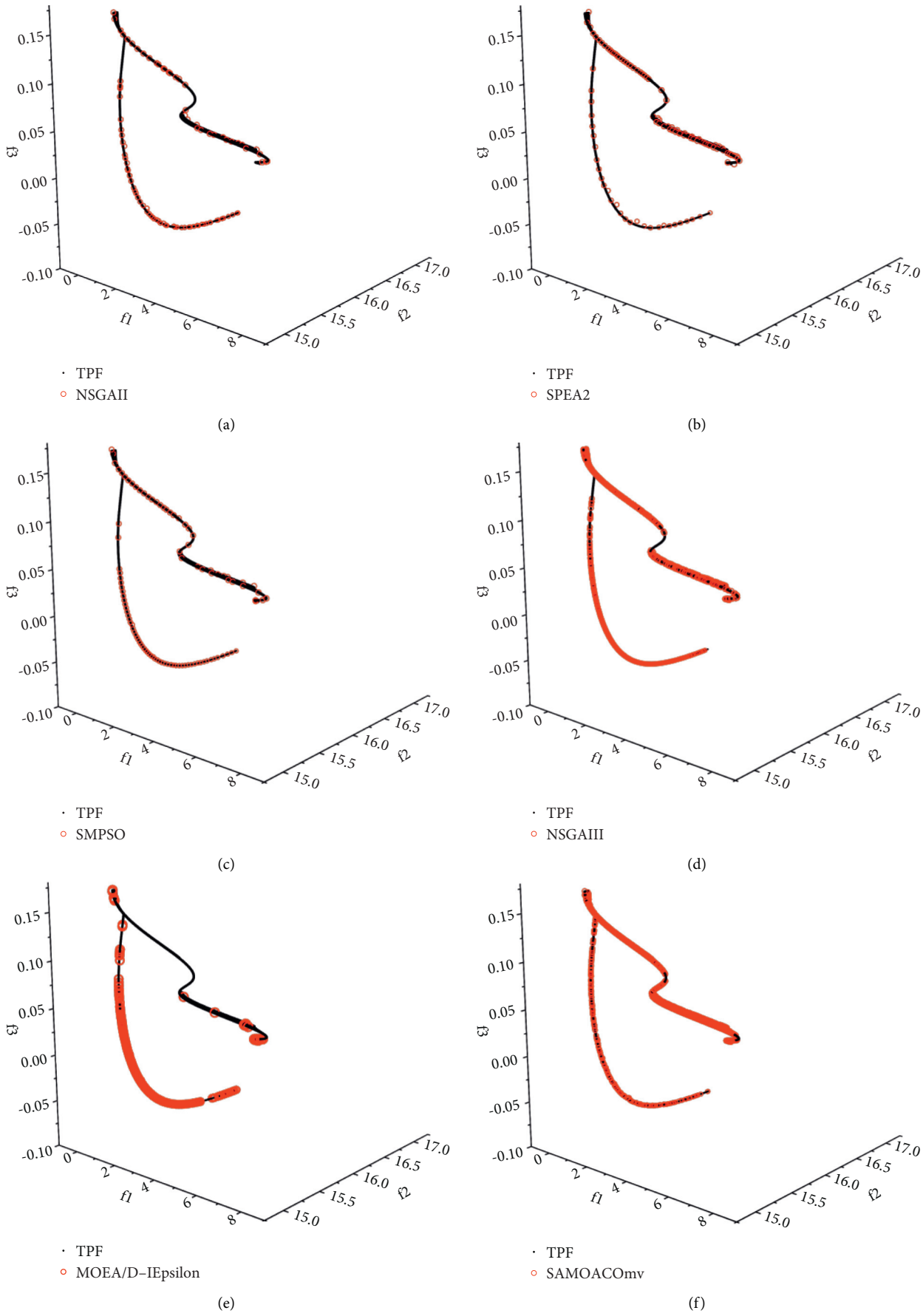


Figure 10: Pareto front for the Viennet3 problem.

performance verification example [37, 38], and it is a mixed-variable MOO problem containing continuous and discrete variables. We use the spring design problem to test the performance of the SAMOACO$_{MV}$ algorithm in this paper.

4.1. Problem Description. The spring design problem consists of two discrete variables and one continuous variable. The objectives are to minimize the volume of the spring and minimize the stress developed by applying a load. Variables are the diameter of the wire (d), the diameter of the spring (D), and the number of turns (N). Denoting the variable vector $\vec{x} = (x_1, x_2, x_3) = (N, d, D)$, formulation of this problem with two objectives and eight constraints is as follows [38]:

$$\begin{aligned}
&\text{minimize } f_1(\vec{x}) = 0.25\pi^2 x_2^2 x_3 (x_1 + 2)\\
&\text{maximize } f_2(\vec{x}) = \frac{8KP_{\max}x_3}{\pi x_3^2}\\
&g_1(\vec{x}) = l_{\max} - \frac{P_{\max}}{k} - 1.05(x_1 + 2)x_2 \geq 0\\
&g_2(\vec{x}) = x_2 - d_{\min} \geq 0,\\
&g_3(\vec{x}) = D_{\max} - (x_2 + x_3) \geq 0,\\
&g_4(\vec{x}) = C - 3 \geq 0,\\
&\text{Subject to}\\
&g_5(\vec{x}) = \delta_{pm} - \delta_p \geq 0,\\
&g_6(\vec{x}) = \frac{P_{\max} - P}{k} - \delta_w \geq 0,\\
&g_7(\vec{x}) = S - \frac{8KP_{\max}x_3}{\pi x_2^3} \geq 0,\\
&g_8(\vec{x}) = V_{\max} - 0.25\pi^2 x_2^2 x_3 (x_1 + 2) \geq 0,
\end{aligned} \tag{15}$$

where x_1 is an integer, x_2 is a discrete variable, and x_3 is a continuous variable.

The parameters used are as follows:

$$\begin{aligned}
K &= \frac{4C - 1}{4C - 4} + \frac{0.615x_2}{x_3},\\
P &= 300\text{lb},\\
D_{\max} &= 3\text{ in}, k = \frac{Gx_2^4}{8x_1x_3^3},\\
P_{\max} &= 1000\text{lb},\\
\delta_w &= 1.25\text{in}, \delta_p = \frac{P}{k},\\
l_{\max} &= 14\text{in},\\
\delta_{pm} &= 6\text{in},\\
S &= 189\text{ ksi},\\
d_{\min} &= 0.2\text{in},\\
C &= \frac{x_3}{x_2},\\
G &= 11,500,000\frac{\text{lb}}{\text{in}^2}, V_{\max} = 30\text{in}^3.
\end{aligned} \tag{16}$$

The 42 discrete values of d are given below:

$$d = \begin{pmatrix}
0.009, & 0.0095, & 0.0104, & 0.0118, & 0.0128, & 0.0132,\\
0.014, & 0.015, & 0.0162, & 0.0173, & 0.018, & 0.020,\\
0.023, & 0.025, & 0.028, & 0.032, & 0.035, & 0.041,\\
0.047, & 0.054, & 0.063, & 0.072, & 0.080, & 0.092,\\
0.105, & 0.120, & 0.135, & 0.148, & 0.162, & 0.177,\\
0.192, & 0.207, & 0.225, & 0244, & 0.263, & 0.283,\\
0.307, & 0.331, & 0.362, & 0.394, & 0.4375, & 0.5.
\end{pmatrix}. \tag{17}$$

5. Experiment Results

From Table 7, the mean of GD, IGD^+, and generalized spread of SAMOACO$_{MV}$ for the spring design problem are about 0.0014, 0.064, and 0.3532, respectively, much smaller than other algorithms. The values of the three performance metrics of SAMOACO$_{MV}$ for the spring design problem are all ranked first, and its overall rank is also the first, which shows that SAMOACO$_{MV}$ is optimal in convergence and coverage.

Table 7: The performance for spring design problem.

Spring design	Generational distance			Inverted generational distance			Generalized spread			Sum of ranks	Overall rank
	Mean	Stdev	Rank	Mean	Stdev	Rank	Mean	Stdev	Rank		
NSGAII	0.0119	0.0157	3	0.0225	0.0802	4	0.9629	0.1547	4	11	4
SPEA2	0.0043	0.0027	2	0.0173	0.0399	3	0.9742	0.1291	5	10	2
SMPSO	0.0161	0.0154	4	0.2132	0.1329	5	0.8366	0.1505	2	11	4
GDE3	0.0885	0.1489	5	0.0068	0.0053	2	0.8876	0.1654	3	10	2
$SAMOACO_{MV}$	0.0014	0.0007	1	0.0064	0.0027	1	0.3532	0.0338	1	3	1

Table 8: Pareto points for spring design problem.

Spring design	Number of archive points			Number of Pareto points			Percentage of Pareto points in archive			Sum of ranks	Overall rank
	Mean	Stdev	Rank	Mean	Stdev	Rank	Mean	Stdev	Rank		
NSGAII	43.95	30.1531	3	4.5	5.3852	3	0.0775	0.0656	3	9	3
SPEA2	100	0	1	8.15	7.6796	1	0.0815	0.0768	2	4	1
SMPSO	9.7	0.0897	5	0.15	0.4894	5	0.0131	0.0403	5	15	5
GDE3	16.3	13.0307	4	0.45	0.887	4	0.0254	0.0513	4	12	4
$SAMOACO_{MV}$	55.65	6.483	2	6.35	7.4288	2	0.1044	0.1099	1	5	2

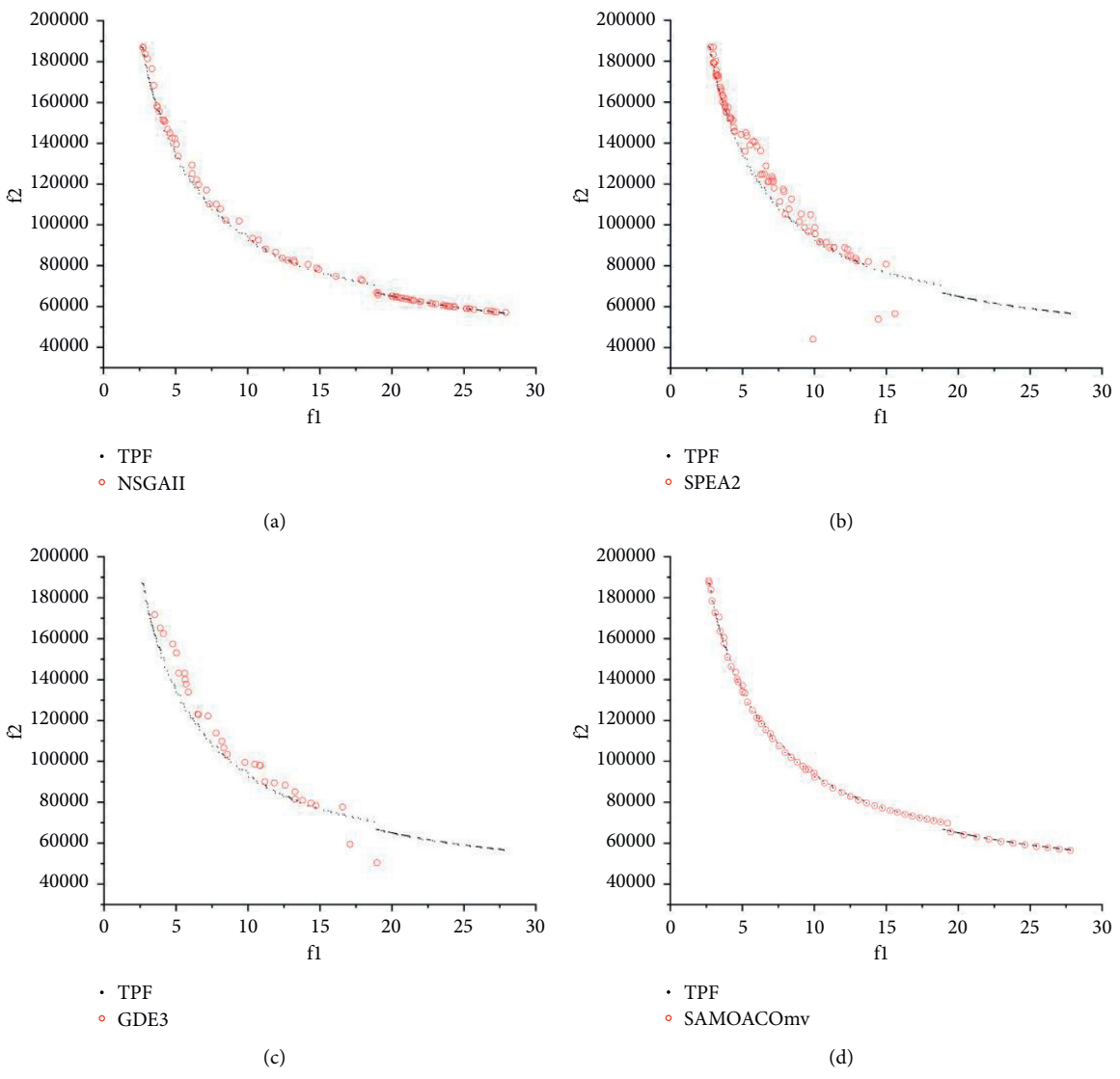


Figure 11: Pareto front for the spring design problem.

As shown in Table 8, for the spring design problem, the number of archive points and the number of Pareto points of $SAMOACO_{MV}$ rank 2, and the percentage of Pareto points in archive ranks 1, which means that $SAMOACO_{MV}$ has the highest comprehensive efficiency in finding Pareto points.

The obtained Pareto frontier is plotted in Figure 11. The TPF represents the set of non-inferior solutions obtained by merging all experimental results from all independent runs of all algorithms and removing the inferior solution. SMPSO can only obtain a few Pareto points, so it is not shown by the figure. It can be seen from Figure 11 that many points of NSGA-II, SPEA2, and GDE3 do not converge to the TPF, and some points of SPEA2 and GDE3 are far away from TPF. The Pareto points of NSGA-II, SPEA2, and GDE3 have poor distributions, and SPEA2 and GDE3 only cover part of TPF. In contrast, the Pareto points obtained by $SAMOACO_{MV}$ widely and uniformly distributed along the TPF, which illustrates that it has better convergence and diversity compared with the other algorithms.

6. Conclusion

In this work, we have modified the single-objective optimization algorithm ACO_{MV} to handle mixed-variable MOO problems and proposed a self-adaptive parameter-setting scheme. Then the performance of $SAMOACO_{MV}$ was thoroughly tested using a set of performance metrics with a well-designed benchmark test suite. Its performance was compared with the state-of-the-art multiobjective optimization algorithms. For all benchmark problems, the $SAMOACO_{MV}$ algorithm has good convergence performance, and its GD and IGD^+ are almost the best. However, the generalized spread of $SAMOACO_{MV}$ is slightly worse, which means that the coverage performance of $SAMOACO_{MV}$ is slightly weaker than other algorithms. For spring design problem, the $SAMOACO_{MV}$ algorithm can get widely and uniformly distributed Pareto front, and it has the best convergence and coverage performance.

In general, the $SAMOACO_{MV}$ algorithm is an excellent MOO algorithm, which adds a new choice for solving MOO problems.

Data Availability

Some or all data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

Disclosure

The approach proposed in this paper has been published at the 2020 IEEE International Congress on Cybermatics (iThings/GreenCom/CPSCom/SmartData/Blockchain-2020) [39]. Based on the conference paper, this paper mainly expands as follows: a new congestion degree of the solution is defined to rank the solutions in the archive, modified the self-adaptive strategy to set the parameters m and k of the $SAMOACO_{MV}$ algorithm, and designed some new mixed-variable MOO benchmark problems to test and compare the performance of the $SAMOACO_{MV}$ algorithm. New performance metrics such as *GD*, IGD^+, and *Generalized Spread* are used to evaluate the performance of the algorithms. All experiments are redone, and the corresponding described text, figure, and table of experimental results are updated.

Conflicts of Interest

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

Acknowledgments

This work was supported by National Key Research and Development Program of China (Grant no. 2018YFC1405700) and Industry University Research Cooperation Project of Jiangsu Province (Grant no. BY2019005).

References

[1] Q. Liu, R. Mo, X. Xu, and M Xu, "Multi-objective resource allocation in mobile edge computing using PAES for Internet of Things," *Wireless Networks*, pp. 1–13, 2020.

[2] M. Huang, Q. Zhai, Y. Chen, S. Feng, and F. Shu, "Multi-objective whale optimization algorithm for computation offloading optimization in mobile edge computing," *Sensors*, vol. 21, no. 8, p. 2628, 2021.

[3] G. Fan, L. Chen, H. Yu, and W. Qi, "Multi-objective optimization of container-based microservice scheduling in edge computing," *Computer Science and Information Systems*, vol. 18, 2020.

[4] X. Xu, R. Gu, F. Dai, L. Qi, and S. Wan, "Multi-objective computation offloading for internet of vehicles in cloud-edge computing," *Wireless Networks*, vol. 26, no. 11, 2020.

[5] C. García-Martínez, O. Cordón, and F. Herrera, "A taxonomy and an empirical analysis of multiple objective objective ant colony optimization algorithms for the Bi-criteria TSP," *European Journal of Operational Research*, vol. 180, no. 1, pp. 116–148, 2004.

[6] C. A. Coello, D. A. V. Veldhuizen, and G. B. Lamant, *Evolutionary Algorithms for Solving Multiobjective Problems*, Kluwer Academic Publishers, NY, USA, 2002.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: nsga," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[8] E. Zitzler and M. Laumanns, "SPEA2: improving the strength Pareto evolutionary algorithm," TIK-Report, vol. 103, 2001, https://doi.org/10.3929/ethz-a-004284029.

[9] J. Knowles and D. Corne, "The Pareto archived evolution strategy: a new baseline algorithm for multiobjective optimization," *NJ*, IEEE Press, in *Proceedings of the 1999 Congress on Evolutionary Computation,Piscataway*, pp. 9–105, IEEE Press, Washington, July 1999.

[10] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[11] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. Coello, and E. Alba, "SMPSO: a new PSO-based metaheuristic for multi-objective optimization," in *Proceedings of the Computational Intelligence in Miulti-Criteria Decision-Making*, March 2009.

[12] M. R. Sierra and C. C. A. Coello, "Improving PSO-based multi-objective optimization using crowding, mutation and e-dominance," in *International Conference on Evolutionary Multi-Criterion Optimization Springer*, Berlin, Germany, 2005.

[13] "Gde3, "The Third Evolution Step of Generalized Differential Evolution," in *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)*, pp. 443–450, Sept 2005, http://www.iitk.ac.in/kangal/papers/k2005013.pdf.

[14] H. Li and Q. F. Zhang, "Multi-objective optimization problems with complicated Pareto sets," *MOEA/D and NSGA-II*, 2008.

[15] Z. Fan, W. Li, X. Cai et al., "An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions," *Soft Computing*, vol. 23, 2017.

[16] C. E. Mariano, E. Morales, and P Cuauhnahuac, "A multiple objective ant-Q algorithm for the design of water distribution irrigation networks," *Technical Report HC-9904, Instituto Mexicano de Tecnología del Agua, Progreso, Mexico*, 1999, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.7622.

[17] K. Doerner, W. J. Gutjahr, R. F. Hartl, C Strauss, and C Stummer, "Pareto Ant Colony Optimization: A metaheuristic approach to multiobjective portfolio selection," *Annals of Operations Research*, vol. 131, 2004.

[18] B. Barán and M. Schaerer, "A multiobjective ant colony system for vehicle routing problem with time Windows. Proc," in *Proceedings of the 21st IASTED International Conference on Applied Informatics*, no. 2, pp. 97–102, Insbruck, Austria, January 2003.

[19] P. Cardoso and M. Jesús, "A.Márquez," *MONACO-Multi-Objective Network Optimisation Based on an ACO. Proc.X Encuentros de Geometría Computacional*, Spain, Seville, 2003.

[20] V. T'kindt, N. Monmarché, F. Tercinet, and D. Laügt, "An Ant Colony Optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem," *European Journal of Operational Research*, vol. 142, no. 2, pp. 250–257, 2002.

[21] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[22] J. I. N. Bingyao, "An introduction of some new optimum techniques," *Bulletin of Science and Technology*, vol. 16, no. 3, pp. 119–124, 2000.

[23] J. A. Manson, T. W. Chamberlain, and R. A. Bourne, "MVMOO: mixed variable multi-objective optimisation," *Journal of Global Optimization*, vol. 80, no. 4, pp. 865–886, 2021.

[24] H. Li, Z. Liu, and P. Zhu, "An improved multi-objective optimization algorithm with mixed variables for automobile engine hood lightweight design," *Journal of Mechanical Science and Technology*, vol. 35, no. 5, pp. 2073–2082, 2021.

[25] Z. O. Khokhar, H. Vahabzadeh, A. Ziai, G. G. Wang, and C. Menon, "On the performance of the PSP method for mixed-variable multi-objective design optimization." ASME. J. Mech. Des," vol. 132, no. 7, Article ID 071009, 2010.

[26] S. Liao, K. Socha, M. A. Montes de Oca, T. Stützle, and M. Dorigo, "Ant colony optimization for mixed-variable optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 503–518, 2014.

[27] V. Chankong and T Haimes, "Multio- bjective decision making: theory and methodology," North Holland, New York, 1983.

[28] Y. Fei, N. Li, and Z. Han, "Multi-objective optimization method and its application based on Pareto sets," *Hoisting and Conveying Machinery*, no. 9, pp. 13–15, 2006.

[29] F. Kursawe, "A Variant of Evolution Strategies for Vector Optimization," in *Parallel Problem Solving from Nature*, pp. 193–197, Springer, Berlin, 1991.

[30] J. Antonio and J. D Nebro, "jMetal 5.10 [CP]," 2021, https://github.com/jMetal/jMetal.

[31] M. K. Rahman, "An intelligent moving object optimization algorithm for design problems with mixed variables, mixed constraints and multiple objectives," *Structural and Multidisciplinary Optimization*, vol. 32, no. 1, pp. 40–58, 2006.

[32] Y. Xiong, "Mixed discrete fuzzy nonlinear programming for engineering design optimization," *Department of Mechanical Engineering*, pp. 109–111, University of Miami, Coral Gables, Florida, 2002.

[33] Z. Xuejun and P. Deng, "Multiobjective Optimization Design with Mixed- Discrete Variables in Mechanical Engineering via Pareto Genetic Algorithm," *Journal of Shanghai Jiaotong University*, vol. 34, no. 3, pp. 411–414, 2000.

[34] L. Yin, M. Zhuang, J. Jia, and H. Wang, "Energy saving in flow-shop scheduling management: an improved multi-objective model based on grey wolf optimization algorithm," *Mathematical Problems in Engineering*, vol. 2020, Article ID 9462048, 14 pages, 2020.

[35] D. A. Van, V. Gary, and B. Lamont, "Multiobjective evolutionary algorithm research: a history and analysis," *Evolutionary Computation*, vol. 8, no. 2, 1998.

[36] H. Ishibuchi, H. Masuda, and Y. Nojima, "A study on performance evaluation ability of a modified inverted generational distance indicator," pp. 695–702, 2015.

[37] Z. O. Khokhar, H. Vahabzadeh, A. Ziai, and W Gary, "On the performance of the PSP method for mixed-variable multi-objective design optimization," *Journal of Mechanical Design*, vol. 132, no. 7, Article ID 071009, 2010.

[38] K. Deb and A. Srinivasan, "Innovization: innovating design principles through optimization," in *Proceedings of the Genetic and evolutionary computation conference, GECCO*, Proceedings, Seattle, Washington, USA, July 2006.

[39] Y. Gong, W. Wang, and S. Gong, "Research of a self-adaptive mixed-variable multi-objective ant colony optimization algorithm," in *Proceedings of the 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, pp. 735–742, Rhodes, Greece, November 2020.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 1870779, 15 pages
https://doi.org/10.1155/2022/1870779

Research Article

A Lightweight Data Integrity Verification with Data Dynamics for Mobile Edge Computing

Haiyan Wang, Yi Lin, and Fu Xiao

School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China

Correspondence should be addressed to Haiyan Wang; wanghy@njupt.edu.cn

Received 4 November 2021; Accepted 7 February 2022; Published 4 March 2022

Academic Editor: Xiaolong Xu

As a special scenario of mobile cloud computing, mobile edge computing can meet the requirements of low latency of data integrity verification and support of mobility in mobile scenarios. However, most existing data integrity verification methods have relatively large computational overhead and few considerations of data dynamic update. To address the above problems, we propose a lightweight data integrity verification method that can support data dynamics in mobile edge computing scenarios. The proposed method is based on an algebraic signature and data integrity verification framework, which ensures security and reduces the computational overhead to achieve the requirement of lightweight. On this basis, analysis and proof of the feasibility, security, and privacy are given. At the same time, in order to support the dynamic update of the data, an optimized strategy based on matrix index is designed with low overhead. In comparison with other baseline methods, simulation experiments show that our method is superior in terms of computational overhead and has good performance in supporting data dynamics.

1. Introduction

With the development of cloud computing, mobile edge computing (MEC) has been proposed as a special scenario of mobile cloud computing (MCC), which plays an important role in mobile scenarios with low service latency [1, 2]. When verifying the integrity of data stored on remote servers, MEC can provide lower service latency and support mobility, which is more appropriate for data integrity verification in mobile scenarios than MCC.

Most of the existing data integrity verification methods are based on cryptography theory, signature strategy, and blockchain. Cryptography theory includes elliptic curve cryptography theory [3] and homomorphic verification [4], and signature strategy includes ZSS short signatures [5, 6] and aggregate signatures [7]. These methods can safely and reliably verify the data stored in remote servers. However, for mobile users, the computing performance and communication resources of their portable mobile devices are limited, such as when users are on a high-speed train or bus journey. Most of the existing methods for data integrity verification are based on mobile cloud computing scenarios, which consume a large computational overhead. Moreover, the methods have poor support for data dynamic update operations, which is not conducive to the dynamic update operation of user data in mobile scenarios. Therefore, data integrity verification in mobile scenarios to meet the rapidly growing needs of low latency, low computing overhead, and support for mobility has important research significance.

To solve the above problems, based on our previous work, combined with mobile edge computing and data integrity verification, we propose a lightweight data integrity verification method supporting data dynamics in a mobile edge computing scenario. The method can take into account both security and computing overhead to achieve lightweight requirements. In addition, considering the low performance of the user's mobile device and difficulty in independently verifying the integrity of the data, we introduce a third-party audit (TPA) and assume that it is not fully trusted to verify the security of our method. When the data block being queried is missing at the edge, cloud services may provide additional assistance, and our method will check all data blocks in the data set to eliminate any illegal operations that may exist. The main contributions of our work can be summarized as follows:

(i) Firstly, we design a system framework based on data integrity verification in mobile edge computing scenarios. In our framework, we use edge nodes to prestore the data blocks to be checked and use a semitrusted third-party auditor to verify the integrity of the data blocks that users need to query. Our method can ensure that the data blocks are securely protected, and users' privacy will not be disclosed.

(ii) Secondly, we propose a data integrity verification protocol based on algebraic signatures (ASDIV-MEC); this protocol can not only ensure security and reliable verification of data but also ensure low computational overhead in the case of verifying all data blocks to achieve lightweight requirements, allowing users to verify the integrity of data under acceptable computational and communication overhead. On this basis, the feasibility, security, and privacy of the algorithm and performance are analyzed and proved.

(iii) Thirdly, based on previous research, we propose an optimization strategy for data dynamic update, which uses the data dynamic operation based on matrix index to support dynamic update of user data in mobile scenarios and reduces the computing overhead of dynamic update.

(iv) Finally, we carry out a series of simulation experiments. Through simulation and comparison experiments, our method is superior to other methods in terms of computational overhead, which verifies the efficiency of the method. At the same time, we conduct comparative experiments on several data dynamic update operations to further verify that our optimization strategy has better performance than other methods.

The rest of the paper is organized as follows: Section 2 introduces the background of data integrity verification, mobile edge computing, and algebraic signatures. Section 3 expounds the system framework from the design goal and the architecture. Section 4 describes the content of the ASDIV-MEC agreement in detail. Section 5 analyzes and proves the performance of the proposed method. Section 6 gives the specific content of the dynamic update optimization strategy. In Section 7, a simulation experiment is carried out, and the experimental results are given. Finally, Section 8 summarizes this paper.

2. Background

In this section, we describe the related work. Firstly, the development of data integrity verification methods and the shortcomings of each method are introduced. Secondly, the application of data integrity verification in the mobile edge computing scenario is given. Finally, we introduce the algebraic signatures used in this paper.

2.1. Data Integrity Verification. Data integrity verification was first proposed by Deswarte et al. [8], who proposed two data integrity verification methods using hash operation and Diffie–Hellman key exchange protocol. However, the method based on hash operation requires a lot of calculation and communication overhead. Subsequently, Venkatesh et al. [9] considered using homomorphic encryption technology based on RSA signatures for integrity verification, but this method also requires a lot of computational overhead. On this basis, Ateniese et al. [10] proposed a probabilistic verification method using message authentication codes to reduce communication overhead. The work of Shacham and Waters [11] used the Boneh–Lynn–Shacham (BLS) signatures mechanism to construct a homomorphic encryption verifiable label and proved the security and reliability of the mechanism. Wang et al. [12] considered the characteristics of the Merkle hash tree and proposed using the Merkle hash tree to verify the correctness of the data block.

In recent years, the combination of data integrity verification and cloud computing has also been studied. Zhu et al. [6] proposed a cloud-IoT data integrity verification method combined with ZSS signatures. However, they use ZSS signatures-based data integrity verification, which is limited by the large computational overhead required and cannot verify all data blocks to ensure that 100% of the data blocks are complete. Shen et al. [13] proposed the use of algebraic signatures for data integrity verification, but this method did not achieve faster data processing and analysis in mobile edge scenarios to reduce latency and support mobility. Ren et al. [14] proposed a sensor data integrity verification mechanism based on bilinear mapping accumulators. Compared with other works, this method is considered to verify the integrity of the entire data set, but bilinear mapping-based methods require relatively high computational overhead, and they do not consider whether the verifier is secure and reliable, so they cannot completely ensure the correctness of data verification results. Fan et al. [7] used aggregate signatures to verify data integrity. X. Lu and Pan [15] proposed a security and lightweight integrity verification method for IoT mobile terminal devices, which ensures the privacy and efficiency of data sharing in the cloud and achieves relatively lightweight operations for data owners.

At the same time, based on the key characteristics of blockchain decentralization, many researchers have studied the data integrity verification scheme based on blockchain. Aiming at the shortcomings of existing data integrity verification schemes, Wang et al. [16] proposed a data integrity verification scheme based on blockchain, which greatly improved the efficiency and security of the verification process. In order to avoid overreliance on TPA, Yue et al. [17] proposed a data integrity verification framework for distributed edge cloud storage (ECS) based on blockchain, which adopted a Merkle tree with random challenge number to verify data integrity without relying on TPA. Similarly, Liu et al. [18] believed that the reliability of the framework

based on TPA was not satisfactory and then proposed a data integrity verification framework based on blockchain. While existing blockchain-based data integrity verification schemes can avoid the trust issues of TPA, they must face another challenge, the issue of huge computing and communication overhead.

2.2. Mobile Edge Computing. Mobile edge computing (MEC) is a special scenario of mobile cloud computing (MCC). It provides lower service latency than MCC. It is used to meet the low latency and high mobility requirements of data integrity verification in mobile scenarios to enhance the ability of IoT terminal devices to process data. With the development of 5G technology and the widespread application of mobile networks, MEC has received widespread attention. MEC has been applied to many fields, such as healthcare, education, and public services [2]. In recent years, the security issues of MEC have also attracted some attention. For example, Tong et al. [19] first studied the data integrity verification of mobile edge computing. They proposed two methods, which are suitable for users who want to verify the integrity of unilateral or multilateral data. On the basis of Zhu et al. [6], Wang et al. [5] proposed integrity verification based on ZSS signatures [20] in mobile edge scenarios. This method transfers the data integrity verification to the edge node closer to the user to provide lower delay and meet the user's strong mobility, but the consideration of computational overhead is relatively insufficient. Liu and Shen [4] proposed using homomorphic verification technology to ensure data integrity, but this method is similar to ZSS short signatures, which are based on bilinear mapping, and the problem is also that the computational cost is relatively high.

2.3. Algebraic Signatures. The algebraic signatures are a signature defined on the Galois field. It is a kind of hash function with algebraic properties; that is, the signature of taking the sum of a certain file block is the same as the signature of taking the sum of the corresponding block. Therefore, the algebraic signatures can be regarded as a kind of algebraic hash function, which can return a part of the data signature for data integrity verification, thereby saving the computational overhead of data integrity verification at the edge node. The algebraic signatures method in this paper is based on Mokadem and Litwin [21] and Schwartz and Miller [22]. Algebraic signatures are similar to cryptographically secure hash functions such as MD5 and SHA-1. But MD5 and SHA are not secure in terms of encryption because it is easy to deliberately construct two strings with the same signature [22].

Mokadem and Litwin [21] first used algebraic signatures in Scalable Distributed Data Structures (SDDSs) to check distributed files stored in distributed networks. Schwarz and Miller [22] used algebraic signatures to check data in remote servers. Later, Luo et al. [23] used algebraic signatures in the cloud to check the data possession in the cloud. In the method of Luo et al., a trusted third-party auditor is used to check data in the cloud. In addition, this method uses an index table method to support dynamic operations of data. Ping et al. [3] and Shen et al. [13] based on the work of [22, 23] further proposed the use of algebraic signatures in cloud computing to verify the integrity of data. However, these tasks are based on data integrity verification in cloud computing scenarios, which cannot meet the needs of low latency and strong user mobility.

3. System Model

3.1. Design Objective. Compared with the data integrity verification in the traditional MCC environment, the data integrity verification in the MEC environment faces additional and complex interaction problems, which are more challenging. Although previous work has conducted research on scenarios and data integrity verification, the consideration of low latency and low computing overhead is relatively insufficient, and the research on data dynamic update operations needs to be improved. Therefore, in this paper, we propose a lightweight data integrity verification method that supports data dynamics in mobile scenarios. Our goals are as follows:

(i) Design a data integrity verification framework in a mobile scenario and design an algebraic signature-based data integrity verification protocol (ASDIV-MEC) under this framework. The verification protocol will verify all required data blocks instead of randomly selecting data block verification and, on the premise of guarantee security, maintain a low computational overhead to meet the requirements of lightweight. At the same time, the protocol ensures that the semitrusted third-party auditor will not obtain the user's private information from the verification, ensuring the security and privacy of the entire verification process. On this basis, in order to ensure the complete verification of the data, the situation where a single edge node, multiple edge nodes, and the cloud collaborate together is considered.

(ii) A data dynamic update optimization strategy with a lower computational cost is designed for mobile users to dynamically update data. This strategy is based on an index matrix to support the correct update of mobile user data and ensure low computational cost.

3.2. Architecture of ASDIV-MEC. The overall architecture of our method is shown in Figure 1, which consists of four types of entities: users, edge, cloud, and semitrusted third-party auditor (semi-TPA). The following is a detailed functional description of these four main components:

User: the user is the initiator who queries the data block and requests verification of integrity. Each user has a different fixed or mobile device, and the location is also different. Through our method, the user quickly signs the uploaded data with low computational overhead, stores the data in the cloud, and sends the signatures to

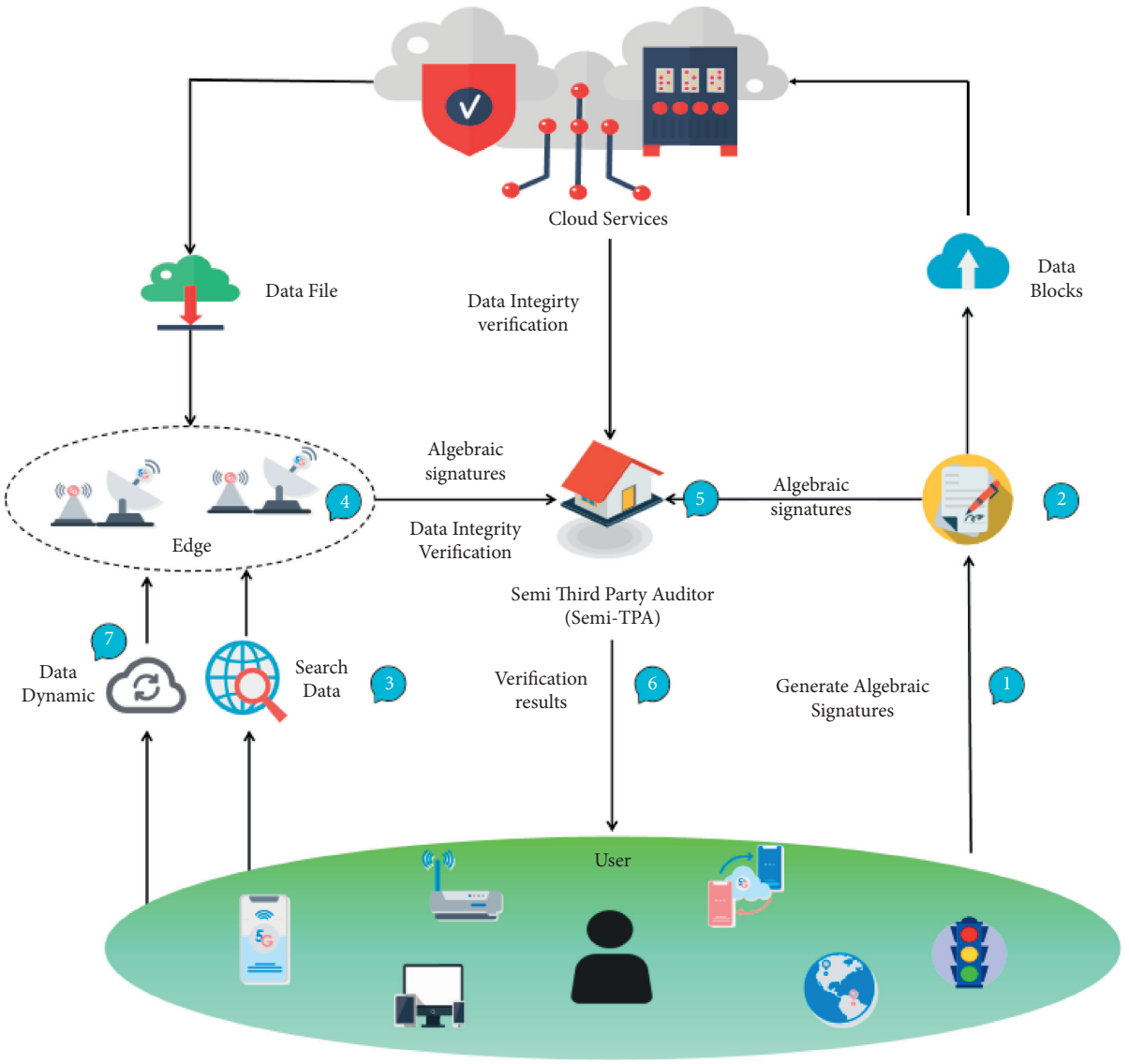


Figure 1: The architecture of ASDIV-MEC.

TPA. When the user needs to query a data block, a request is sent to the edge node through the method, and the verification result is obtained with lower latency and computational overhead so as to achieve a lightweight effect. User can also update data dynamically according to the data update optimization strategy, which consumes less computing overhead.

Edge: it is deployed at the edge node of the network close to the terminal device and the user. The edge node has the characteristics of miniaturization, distribution, and being closer to the user, which can realize the processing of data on the edge of the network, reduce the request and response time and computing overhead, and support users' strong mobility. After the user sends the integrity verification request, the edge node finds the corresponding data block from the predownloaded data, uses our method to quickly generate the corresponding proof with low overhead, and returns the result to the TPA for verification.

Cloud: it is a server that stores and processes user data. Each cloud service provider has powerful capabilities and huge storage space to provide comprehensive services with a short execution time, which greatly saves local storage space. In addition to storing the entire data file for the user, the cloud can also generate a certificate for the missing data block in the edge server and send it to the TPA.

Semi-TPA: it has powerful computing and storage capabilities and performs data integrity verification. Our method introduces TPA to reduce the user's computing overhead and the communication overhead between the user and the edge node and replace the user to perform integrity verification, which can meet

the lightweight requirements. TPA collects and verifies the proof sent from the edge node and returns the verification result to the user.

As shown in Figure 1, in the MEC environment, due to the low computing and storage performance of mobile devices, mobile users use our method to quickly divide data into multiple data blocks with a small amount of overhead and generate signatures for each data block and then upload the data blocks to the cloud storage device and delete the data on the local device. After that, the signature generated by the corresponding data is sent to TPA. In many studies, it is generally assumed that TPA is completely reliable. This is unrealistic because TPA may be attacked by external or internal attacks and return wrong results to users, posing a threat to data security. Our model system can handle this situation well, so we assume that TPA is semireliable. Edge nodes deployed near mobile terminals periodically predownload data blocks from remote clouds to provide data integrity services with low latency and computing overhead to achieve lightweight requirements. When a user requests to verify data blocks, the user sends the request to the edge node, which generates a proof of the data blocks to be verified with low computational overhead and sends it to the TPA. After that, TPA verifies whether the received data block proof and the data block signature uploaded by the user are correct according to the characteristics of the algebraic signature and returns the result to the user.

Users usually verify the integrity of data stored on edge nodes before querying the stored data. When the queried data is not predownloaded on the edge nodes, the cloud server will provide a certificate to help return the stored data to the TPA for verification. When a user needs to update stored data, he can send an update request through our method. The method adopts the data dynamic update optimization strategy to dynamically update the user's data with a low computational overhead to ensure lightweight operations.

4. Protocol of ASDIV-MEC

4.1. Preliminaries. This paper introduces a secure and efficient data integrity verification signature, which is a hash function with algebraic properties, that is, algebraic signatures, which can quickly sign data blocks to improve signature efficiency. Algebraic signature is a hash function with algebraic properties proposed by Schwarz and Miller [22]; that is, the sum of the signatures of a certain file block is the same as the signature of the sum of the corresponding block. Algebraic signatures have homogeneity and algebraic properties, consume less computational overhead when signing and verifying data, and can be used for lightweight integrity verification of remote data [22].

Assuming that a block of data in data file F is composed of $F_1, F_2, \ldots, F_n$, then the algebraic signatures have the following properties.

4.1.1. Compressibility.

$$AS_\alpha(F_1, F_2, \ldots, F_N) = \sum_{i=1}^{N} F_i \alpha^i. \tag{1}$$

It can be seen from equation (1) that algebraic signatures are compressible and can compress a file into a small string, where AS is the algebraic signature method and α is the algebraic signature parameter. When the original file is modified, the corresponding algebraic signatures value will also change accordingly. This attribute is similar to the hash functions MD5 and SHA in cryptography. However, when using MD5 and SHA, users need to keep all hash values and must retrieve all their own data, which causes huge communication and computational overhead. Therefore, MD5 and SHA are not suitable for remote data integrity verification [22]. The algebraic characteristics and compressibility of algebraic signatures enable users to check all data stored remotely with less overhead. Therefore, algebraic signatures are an ideal way to verify whether remote data is stored intact.

4.1.2. Algebraic Property. In addition, the technology has low communication and computing overhead. Suppose two large data files X and Y are composed of n subblocks, which are, respectively, expressed as $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_n$. The algebraic properties of algebraic signatures can be observed as follows:

$$\begin{aligned} AS_\alpha(X) + AS_\alpha(Y) &= AS_\alpha(x_1, x_2, \ldots, x_N) + AS_\alpha(y_1, y_2, \ldots, y_N) \\ &= \sum_{i=1}^{N} x_i \alpha^i + \sum_{i=1}^{N} y_i \alpha^i = \sum_{i=1}^{N} (x_i + y_i)\alpha^i = AS_\alpha(X + Y). \end{aligned} \tag{2}$$

Therefore, algebraic signatures enable edge nodes to return a portion of the data signature for data integrity checking, thus saving bandwidth for edge node data integrity checking applications. In the face of the diversity of data, the following expansion can also be carried out:

$$\begin{aligned} &AS_\alpha(X) + AS_\alpha(Y) + \cdots + AS_\alpha(Z) \\ &= AS_\alpha(x_1, x_2, \ldots, x_N) + AS_\alpha(y_1, y_2, \ldots, y_N) + \cdots + AS_\alpha(z_1, z_2, \ldots, z_N) \\ &= \sum_{i=1}^{N} x_i \cdot \alpha^i + \sum_{i=1}^{N} y_i \cdot \alpha^i + \cdots + \sum_{i=1}^{N} z_i \cdot \alpha^i = \sum_{i=1}^{N} (x_i + y_i + \cdots + z_i) \cdot \alpha^i = AS_\alpha(X + Y + \cdots + Z). \end{aligned} \tag{3}$$

Algebraic signatures consist mainly of the following three functions:

KeyGen: in the KeyGen phase, it generates some initialization parameters, such as the master key k, the signatures parameter, and some random parameters.

Sign: the algebraic signatures of a data block are as follows: $AS_\alpha(F_i) = F_i \cdot \alpha^i$.

Verify: given a fixed key k, data block F_i', and signature Sig, the verifier needs to verify that the signature of the stored data block is equal to the original signature. If they are equal, then the signature is generated by the user who has the signature, and verification can be confirmed as follows: $AS_\alpha(F_i') = \text{Sig}$.

For the sake of clarity, we list some notations and their descriptions in Table 1, which will be used throughout the paper.

4.2. ASDIV-MEC. ASDIV-MEC includes five stages: parameter generation stage, signature generation stage, challenge generation stage, proof generation stage, and verification proof stage. We use **Paragen**, **SigGen**, **ChallGen**, **ProofGen**, and **VerifyProof** to represent these five stages. These five stages are described in detail as follows.

Step 1. KeyGen() $\longrightarrow$, (k, a, r_1, r_2): first, some parameters need to be generated in the initial stage. The user needs to generate a master key k and signature parameter a from the secure hash function. Meanwhile, the user and the TPA generate two security parameters r_1 and r_2, according to the secure hash function. Finally, the TPA sends r_2 to the user over a secure channel between the client and the TPA.

Step 2. SigGen $(F, k, r_1, r_2) \longrightarrow$ (Sig): this stage generates an algebraic signature for each data block. The user divides the data file F into blocks: $\{m_1, m_2, \ldots, m_n\}$. Since TPA is not necessarily honest, in order to ensure the security of data blocks, data owners need to preprocess data blocks $m_i \in F$ to prevent the disclosure of user data when TPA is attacked.

$$F_i = m_i \oplus H(r_1 \| i), \tag{4}$$

where H is a one-way hash function with collision resistance, used to protect security parameter r_1 and block of the confidential nature of ID_i, and $\|$ is a concatenation operation.

From the above operation, the data block of data file $F = \{F_1, F_2, \ldots, F_n\}$. According to our method, the signature of block i can be generated with little overhead.

$$\text{Sig}_i = F_i \cdot a^{i \cdot r_2}. \tag{5}$$

On this basis, the signature of the data file F is $\text{Sig} = \{\text{Sig}_1, \text{Sig}_2, \ldots, \text{Sig}_n\}$, which reduces communication overhead and facilitates support for TPA auditing.

After that, the user sends the data file block $F = \{F_1, F_2, \ldots, F_n\}$ that is uploaded to the cloud server for storage, and the edge node periodically downloads the data block that the user needs to query in advance and signs the data block $\text{Sig} = \{\text{Sig}_1, \text{Sig}_2, \ldots, \text{Sig}_n\}$ that is uploaded to TPA.

Step 3. ChallGen (r_1), $\longrightarrow$, (chall): when a user wants to check whether some of his data is stored intact in the cloud, the user initiates a data integrity request. The user first generates a random x in the Galois field and calculates $c_k = f_k(r_1 + x)$, generates the challenge $\text{chall} = \{(c_k, x)\}$, and sends it to the edge node and TPA.

Step 4. ProofGen (chall, r_2), $\longrightarrow$ $(proof)$: the edge node stores all the data that users need to query. After receiving the chall, our method requires the edge node to generate a certificate based on the algebraic signatures, generate a certificate for the data block to be queried with a lower computational cost, and reply to the TPA with a storage certificate. Considering the possible lack of data transmission between the cloud and edge nodes, we consider three cases in Figure 2 in this step, which are described in detail as follows:

Case 1 (Single Edge). The edge node first calculates the position of the selected block, as shown as follows:

$$l_i = \sigma_k(r_2 + i), \quad 0 \le i \le c. \tag{6}$$

And let $L = \{l_i\}_{0 \le i \le c}$, c is the number of blocks per chall.

Then the edge node computes the algebraic signatures of the sum of the selected blocks:

Table 1: Summary of main notations.

Notations	Descriptions
F	Data file
m_i	i^{th} data block of data file F
n	Number of data blocks
k	Master key
a	Signature parameter
r_i	Security parameter
H	Hash function
c_k	Pseudorandom number
x	Random number generated in Galois field
Chall	Challenge request chall = $\{(c_k, x)\}$
Proof	Proof of data blocks
l_i	The position of i^{th} data block
c	The number of blocks per chall
t	Number of edge nodes working together in Case 2
I	Queried data block index set

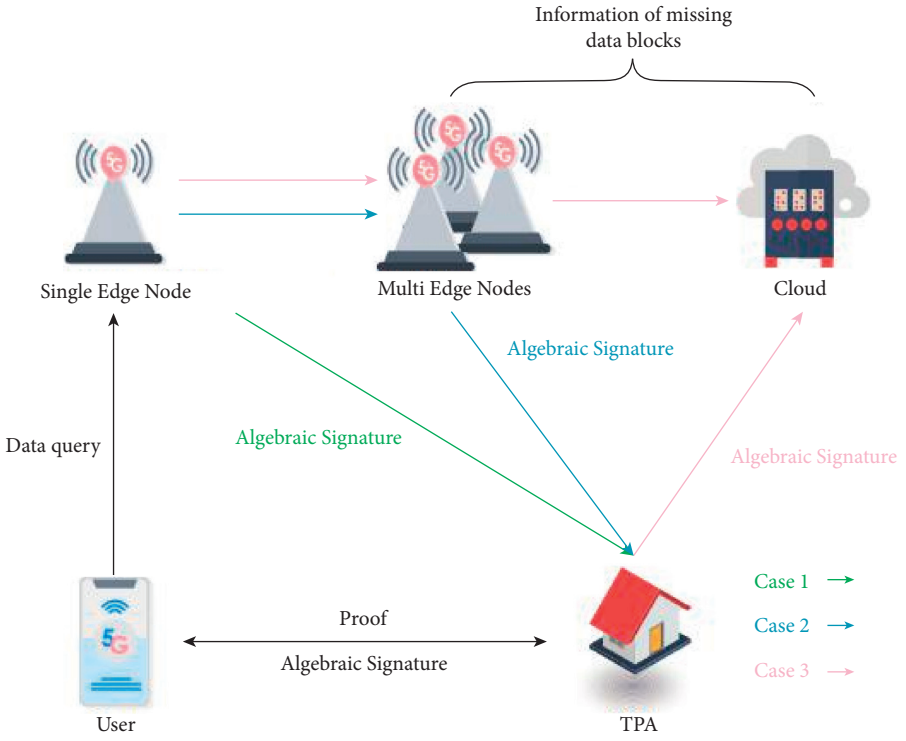


Figure 2: Three cases in our verification protocol.

$$\beta = AS_{\alpha}\left(\sum_{l \in L} F_l\right). \tag{7}$$

The signature of the sum of the computed data blocks is sent to TPA as proof.

Case 2 (Multiple Edges). In this case, a single edge node cannot provide enough data blocks, and it queries nearby edge nodes for help. Let us assume that there are T edge nodes working together to solve this tricky problem, and each edge node has its own index set of data blocks O_j, $j \in \{1, 2, 3, \ldots, t\}$; on this basis, for

every $j \in \{1, 2, 3, \ldots, t\}$, edge node E_j can extract the valid data block index set for the missing data block, as shown as follows:

$$\begin{aligned} I_j &= \left(I - (I \cap O_1) \cup (I \cap O_2) \cup \ldots \cup \left(I \cap O_{j-1}\right)\right) \cap O_j \\ &= \left(I - I \cap \left(O_1 \cup O_2 \cup O_3 \cup \ldots \cup O_{j-1}\right)\right) \cap O_j \\ &= I \cap O_j - I \cap O_j \cap \left(O_1 \cup O_2 \cup O_3 \cup \ldots \cup O_{j-1}\right) j \in \{1, 2, 3, \ldots, t\}. \end{aligned} \tag{8}$$

We define the edge node E_t as the last one if $O_1 \cup O_2 \cup O_3 \cup \ldots \cup O_{t-1} \cup O_t = I$. Assume that $I_j = \left\{s_1, s_2, \ldots, s_{e_j}\right\}$ and $e_j \in \{1, 2, 3, \ldots, n\}$; according to formula (8), we can get

$$\beta = AS_\alpha \left(\sum_t \sum_{l \in L} F_l \right). \tag{9}$$

The last edge node sends the signature of the sum of the computed data blocks to TPA as proof.

Case 3 (A Joint of Multiple Edges and Cloud). In this case, a single edge node and its nearby edge nodes cannot provide enough data blocks and will eventually seek help from the central cloud. The last edge node communicates with the central cloud for the missing data block. The calculation is similar to Case 2. Finally, the central cloud returns the integrated evidence to the TPA.

Step 5. Verify(*chall*, *proof*) $\longrightarrow$ \{TRUE, FALSE\}: when TPA receives the storage proof *proof* from the edge node, it verifies the correctness of *proof*

$$\text{proof} = \sum_n \text{Sig}_i. \tag{10}$$

If the above equation is true, TPA will return true to the user to confirm that the outsourced data is complete; otherwise, it will return false to notify the user that the outsourced data is corrupted.

5. Performance of Analysis

In this section, we analyze the performance of the proposed ASDIV-MEC model system from three aspects, namely, feasibility, security, and privacy. Both edge nodes and TPA are semitrusted in our model. For the sake of description, we examine the case of Case 1 in detail.

5.1. Feasibility

Lemma 1. *If both TPA and edge nodes can carry out data transmission and communication normally, then the data integrity verification scheme ASDIV-MEC proposed in this paper is feasible.*

Proof. According to the protocol described earlier, if the data block on the edge node is stored intact, then TPA can verify the correctness through the proof generated by the edge node and the user-generated signature. We can accurately infer and verify the feasibility of the scheme through the following calculations:

$$\begin{aligned} AS_\alpha(X) + AS_\alpha(Y) &= AS_\alpha(x_1, x_2, \ldots, x_N) + AS_\alpha(y_1, y_2, \ldots, y_N) \\ &= \sum_{i=1}^{N} x_i \cdot \alpha^i + \sum_{i=1}^{N} y_i \cdot \alpha^i = \sum_{i=1}^{N} (x_i + y_i) \cdot \alpha^i = AS_\alpha(X + Y). \end{aligned} \tag{11}$$

From the above derivation that the sum of the user's signatures for each data block is equal to the sum of the signatures of the data block stored in the edge node, it can be concluded that our verification algorithm is feasible. □

5.2. Security and Privacy

Lemma 2. *If our model system is maliciously attacked, TPA can detect data file corruption.*

Proof. Assume that the edge node is attacked and tamper with the data block F_i stored in the edge node. If an attacker wants to pass TPA authentication, then they need to build an alternative signature:

$$\text{Sig}' = AS_\alpha(X^*) = \text{Sig}. \tag{12}$$

Make

$$AS_\alpha(X^*) + AS_\alpha(Y) + \cdots + AS_\alpha(Z) = AS_\alpha(X + Y + \cdots + Z). \tag{13}$$

In order to ensure the security of the original data, this paper uses the hash function and random parameters to blind the original data block and uses the hash function to encrypt the security parameters r_1 and i, which can not only protect the security parameters but also resist forgery attack. The data file is hidden in $F_i \cdot a^{i \cdot r_2}$, and a is the security parameter randomly generated by the data owner in the *KeyGen* stage. In addition, a is masked by the data block i and the user identifier r_1, enhancing the privacy of the data.

Therefore, if the attacker modifies the block m_i, but cannot obtain the user's security parameter r_1, then the following cannot be constructed:

$$F_j^* = m_j^* \oplus H(\cdot) = F_i = m_i \oplus H(\cdot). \tag{14}$$

Thus, (13) cannot be verified through TPA, so the attacker cannot pass TPA verification by constructing a new signature. □

Theorem 1. *When the attacker attacks the TPA or the edge node, the attacker cannot obtain the user's private information by intercepting the signature information.*

Proof. In our approach, TPA has a signature set $Sig_a = \{Sig_i\}$, and the signature set is the hidden user data information generated by $F_i \cdot a^{i \cdot r_2}$. Even in batch authentication, the information of multiple users will be hidden in the data block. Therefore, the security parameters block ID and signature parameters to hide the user's private information. □

Theorem 2. *If the TPA and edge nodes are honest, the integrity verification process is feasible and secure.*

Proof. On the basis of Lemmas 1 and 2, we can prove the theorem directly. □

6. Data Dynamics Optimization Strategy

In mobile scenarios, users need to update the data stored on the server, which consumes a large amount of overhead. However, the limited computing resources of the devices carried by users are not conducive to the data dynamic update in mobile scenarios. Therefore, it is necessary to develop corresponding optimization strategies for the data dynamic update to reduce the computing cost of data update and support the data dynamic update operation of users in mobile scenarios, which has important research significance.

According to the validation method proposed in Section 4, if all entities can honestly communicate and transmit data, we design a data dynamic update optimization strategy that allows users to dynamically insert, delete, and modify data blocks with low computational overhead. We propose two functions: UpdateReq, an update request algorithm, and UpdateExec, an update execution algorithm, to implement dynamic update operations on data blocks. Update operations include adding, deleting, and modifying data blocks. Some existing methods to realize data dynamics mainly focus on using linked lists [3], index tables, and trees [5]. Reference [13] proposed using the matrix to realize the dynamic change of data, and through simulation experiments, it is concluded that the calculation cost of using the matrix method is lower than that of using the index table and tree method. Therefore, this paper uses a matrix-based approach to achieve dynamic data change.

First, we represent the file block in the form of a matrix index. Each row index of the matrix corresponds to a data block, as shown as follows:

$$F = \begin{bmatrix} F_1 & f_{11} & \cdots & f_{1k} \\ F_2 & f_{21} & \cdots & f_{2k} \\ F_3 & f_{31} & \cdots & f_{3k} \\ \vdots & \vdots & \ddots & \vdots \\ F_n & f_{n1} & \cdots & f_{nk} \end{bmatrix}. \tag{15}$$

$F_1, F_2, \ldots, F_n$, respectively, represent the index value of the data block; f_{ij} represents the data subblock j under data block i. If the data subblock exists in the edge node, it is represented by 1; if it does not exist, it is represented by 0. The following matrix index can be obtained:

$$M_I = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 2 & 1 & \cdots & 1 \\ 3 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ n & 1 & \cdots & 1 \end{bmatrix}. \tag{16}$$

When a data block is modified, the index matrix needs to be changed accordingly. Because of the hierarchical structure of data, we can modify both a data block and a data subblock, which is very suitable for the operation of a large amount of data on edge nodes. The index matrix is managed by the user, and the edge nodes and cloud services process the data according to the index matrix provided by the user.

UpdateReq: a function that handles user requests. The input is $\langle \text{BlockOperation}, \text{Index}, M \rangle$, where BlockOperation is based on the specific data block requested by the user, *In de x* is the index of the updated data block, and M is the index matrix. The output is $\langle \text{BlockOperation}, \text{Index}, \text{block}', t', M' \rangle$, where $block'$ is the updated data block, t' is the updated signature, and M' is the updated matrix index.

UpdateExec: it handles functions executed on the edge server. The input is the output of UpdateReq, which is a new copy of the file. After each update, the user can perform challenge validation to ensure that the update operation is correct.

6.1. Insertion Operations. Data insertion includes data block insertion and data subblock insertion. When a user wants to insert a data block after a data index, he finds the row in the index matrix that needs to be inserted and then inserts the new row in the next row. The change process of the index matrix of data block insertion is shown in Figure 3.

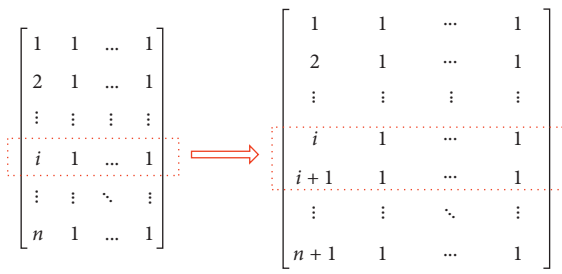


Figure 3: Data block insertion.

The insert operation also supports the insertion of data subblocks. When the user wants to insert a new data block $f_{i,i+1}$ after the data subblock $f_{i,i}$, the system needs to determine the index position of the inserted block. After that, the system will add a column to the index matrix, move the 1 after the index i backward, and finally insert the data subblock at the position of the index $i + 1$. The changing process of the index matrix is shown in Figure 4. No subblocks are inserted except for the index i row, and the other data blocks remain unchanged.

6.2. Deletion Operations. Similarly, data deletion operations include two parts: block deletion and subblock deletion. During block deletion, when you need to delete data block F_i, set the corresponding row matrix to -1, and the row index changes accordingly. The M_{BID} block index deletion matrix is used to assist in data deletion as follows:

$$M_{\text{BID}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -i & -1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{17}$$

During the subblock deletion process, if the subblock $f_{i,i}$ needs to be deleted, the value of the relative position of index matrix S in the deletion matrix can be changed to -1. Therefore, we can write the subblock index deletion matrix M_{SID} as the following matrix:

$$M_{\text{SID}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -1 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{18}$$

As mentioned above, the edge node has an index matrix and a modification matrix for the data. If we need to delete the data block F_i, we can use the index matrix plus the corresponding block index delete matrix to get a new index matrix NM_I. The block F_i deletion process can be written as follows:

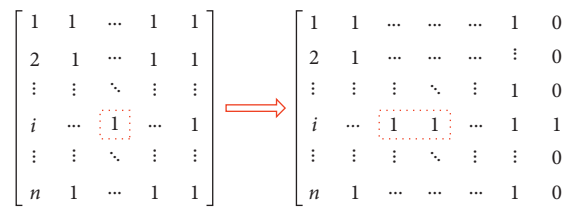


Figure 4: Data subblock insertion.

$$NM_I = M_I + M_{\text{BID}}. \tag{19}$$

If the subblock $f_{i,i}$ needs to be deleted, the index can be modified using the index matrix plus the corresponding subblock index deletion matrix. The new index matrix NM_I can be obtained by

$$NM_I = M_I + M_{\text{SID}}. \tag{20}$$

Using matrix addition operation, the proposed data dynamics method can reduce the deletion operation overhead [13].

6.3. Update Operations. Another common operation is data update. When a block or subblock is updated, the index is first found and deleted, and then the new block or subblock is inserted into the original block location.

Assume the data block F_i is updated to F_i'. First, find the i-th row of the index matrix. Second, delete the values of the matrix rows. Again, after the new data is inserted into the original data location, the corresponding index value is updated. The block update process in an index operation is shown in Figure 5.

Similarly, our scheme supports subblock updates, as well as the other two dynamic operations. To update the data subblock f_{ii} to f_{ii}', the corresponding index value needs to be found and deleted. After the new subblock is inserted, the value of the index matrix is updated. The operation method is similar to data block update. The detailed process of index matrix operation is shown in Figure 6.

Through the above analysis, we can prove the dynamic nature of our integrity verification algorithm. None of these changes breaks the signature policy technology, so the privacy of the data is still protected.

For Cases 2 and 3, we can get similar results because the main difference in performance compared to Case 1 is the additional communication costs from the edge nodes and the cloud.

7. Performance Evaluation

7.1. Experimental Settings. In the experiment, we carried out experiments on the proposed prototype system. For the four entities involved in the system, namely, users, edge nodes, CSP, and TPA, we deployed these entities using machines with different configurations according to the roles and functional requirements in the system model. CSP and TPA are deployed on two Dell Precision T9720 tower servers. The user and edge nodes are deployed on two different laptops.

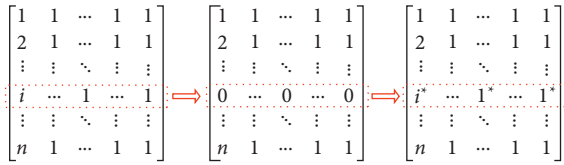


Figure 5: Data block update.

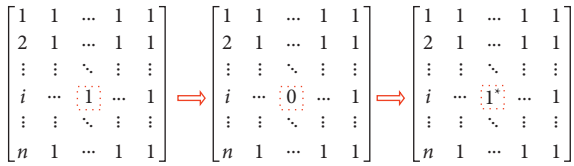


Figure 6: Data subblock update.

Table 2 shows the detailed parameters of our prototype system.

The experiment was carried out under the PBC library-0.5.14, GMP library-6.1.2, and VC ++ 6.0 software environment. In the experiment, the key size is 160 bits, and the pseudorandom number size is 80 bits. These data were the mean values of 40 replicates.

7.2. Baseline Methods. In order to prove the efficiency and effectiveness of our ASDIV-MEC method, two traditional signature methods (RSA signatures and BLS signatures) and two recently worked signature methods (aggregate signatures and ZSS short signatures) are compared with our method.

(1) RSA signatures: the RSA signatures bit ranges from 1024 to 4096. Under the same security condition, the BLS signatures are shorter than the RSA signatures.

(2) BLS: because the RSA algorithm mainly relies on the difficulty of factorization of a large integer, the calculation cost of the RSA-based method is high.

(3) Aggregate signatures scheme: Fan et al. [7] used aggregate signatures to verify data integrity. Aggregation signature is a variant signature method used to aggregate any multiple signatures into a signature. It can combine the public key and signature of each participant in a multisign transaction into a single public key and signature. The entire merge process is invisible, the premerge information cannot be derived from the merged public key and signature, and the verification only needs to be done once.

(4) ZSDIV-MEC: ZSS signature is a bilinear pin-based short signature proposed by Zhang et al. [20], which is based on encrypted hash functions such as SHA-2 or SHA-3. While the overhead of a ZSs-based signatures system is smaller than that of BLS and RSA, the user must store the hash of SHA and then retrieve the entire data file from the edge node or cloud to verify its integrity, resulting in significant communication and computing costs.

Table 2: Experiment environments.

Entity	Experiment environments		
	Device	CPU	RAM (GB)
TPA	Server	Intel XEON silver 4210 @2.20 GHz × 20	64
CSP	Server	Intel XEON silver 4210 @2.20 GHz × 20	128
Edge	Laptop	Intel Core i7 @2.70 GHz × 4	16
User	Laptop	Intel Core i7 @2.70 GHz × 4	16

7.3. Performance Comparisons

7.3.1. Response Time for Different Cases. Firstly, we evaluate the calculation cost of each signature policy in the edge node to compare the performance of each signature policy. We choose the time cost as the index; that is, the time cost is related to the number of data blocks queried. In this experiment, we set a total of 200 data blocks, and the abscissa represents the number of data blocks queried and validated by the user. The size of each data block is 64 kb. The ordinate indicates the query time. At the same time, since there are three cases of our strategy, experiments are conducted on the calculation overhead of queries in different cases.

Figure 7(a) shows the experimental results of Case 1, where the edge node has predownloaded all the data blocks requested by the user for verification. As shown in Figure 7(a), with the increase of the number of queried data blocks, the time cost of the five schemes is getting higher and higher. The RSA-based scheme has the highest time cost, and the time cost increases exponentially. The performance of the scheme based on BLS is slightly worse than that based on aggregate signatures scheme and ZSDIV-MEC, but better than that based on RSA. At the same time, the response time of our method is better than that of the baseline, so the ASDIV-MEC policy is better than the three baseline policies, which illustrates the low computational cost and effectiveness of algebraic signatures in data integrity verification, and meets the requirements of lightweight.

Figure 7(b) shows the experimental results of Case 2. In this experiment, we set a total of 5 edge nodes. It can be seen from the experimental results that our scheme also has the lowest cost, and the RSA-based scheme has the highest cost. In addition, in the case of multiple edge nodes, the response time of each scheme increases significantly compared with the case of a single edge node, which is due to the additional communication costs between the edge nodes. Figure 7(c) shows the experimental results of Case 3. Obviously, the increasing trend is very similar to Case 2. However, for each scenario, the overall response time of Case 3 is slightly greater than that of Case 2 because of the residual costs due to increased communication between the edge nodes and the cloud.

As can be seen from the experimental results, compared with the other three baselines, the proposed ASDIV-MEC solution has lower latency and low computational overhead and can better meet the requirements of lightweight.

7.3.2. Computation Overhead for KeyGen and SigGen. As we all know, *KeyGen* and *SigGen* are two important steps in data integrity verification. And depending on the protocol,

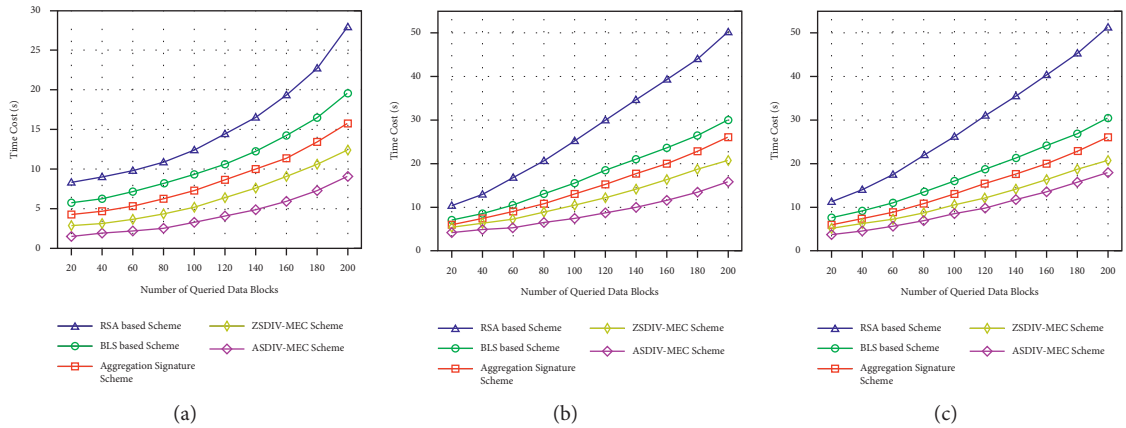


Figure 7: Comparison of response time. (a) Single edge. (b) Multiple edges. (c) A joint of multiple edges and the cloud.

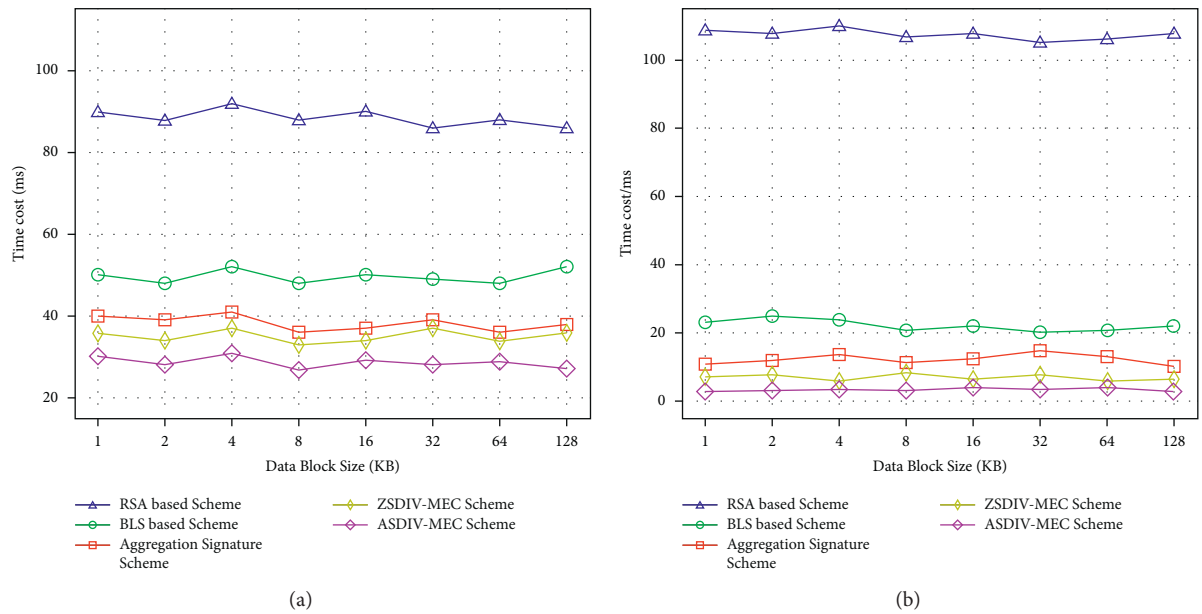


Figure 8: Computation overhead for (a) *KeyGen* and (b) *SigGen*.

the process is very different. Therefore, after comparing the overhead on the edge nodes, we further conducted experiments on the other two important steps in the strategy, *KeyGen* and *SigGen*, for different schemes and calculated the computational overhead of each scheme.

In this experiment, we evaluated the time overhead of key generation and signature generation for data block sizes ranging from 1 kB to 128 kB. As shown in Figure 8(a), we can see that, with the increase of data block size, the proposed time cost plan of ASDIV-MEC is about 20–30 ms, while the time cost plan based on aggregate signatures scheme, ZSDIV-MEC, and BLS is about 30–40 ms. The time cost of the RSA-based solution was much higher than the other three solutions at about 90 milliseconds, nearly three times the cost of the plan. As shown in Figure 8(b), it is clear that our proposed scheme always outperforms the baseline during the signature generation phase. This is because the computational security of the RSA algorithm depends on the difficulty of factorizing large integers, while the BLS signatures require a specific hash function, which is especially efficient for large-scale data. The strategy based on aggregation signatures is based on bilinear mapping, which leads to high overhead. Although ZSS signatures use general hash functions (such as SHA-2 and SHA-3), the calculation of its short signatures is complicated and increases the time cost, while the proposed ASDIV-MEC is relatively simple. Therefore, in terms of *KeyGen* and *SigGen* for data integrity verification, our ASDIV-MEC has better performance in terms of computational overhead, which can improve the efficiency of signature and meet the requirement of lightweight.

7.3.3. Computation Overhead Comparison. In order to better explain the reasons for the low latency and low computational overhead of ASDIV-MEC, we compared the

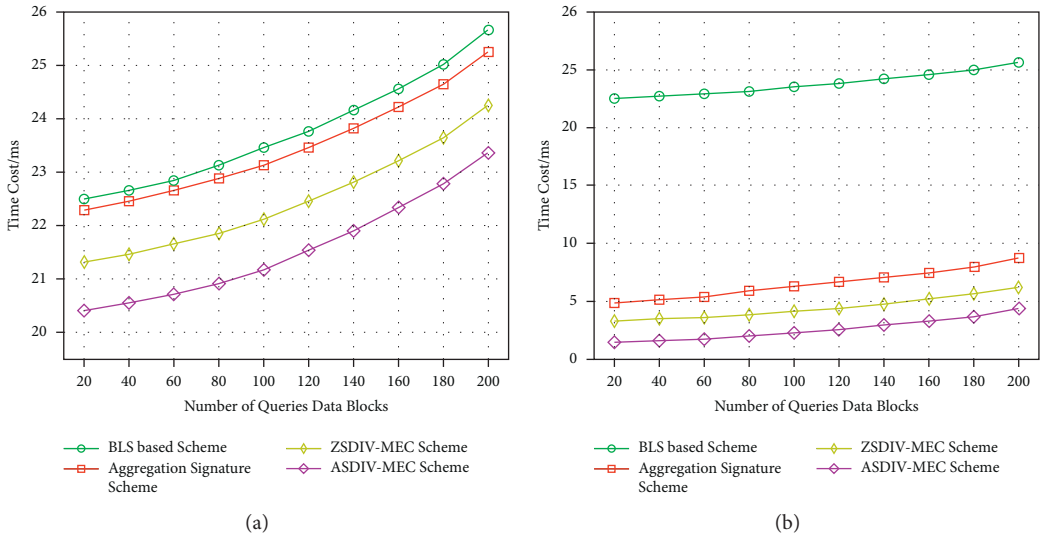


Figure 9: Computation overhead on (a) *User* and (b) *TPA*.

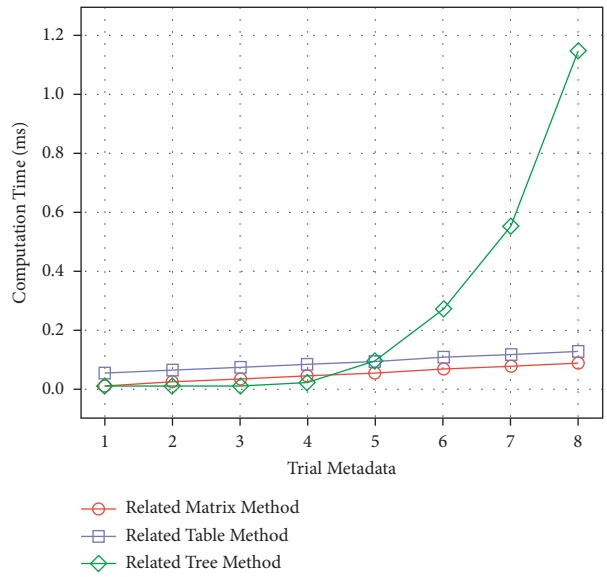


Figure 10: Comparison of different methods.

computational complexity of the BLS-based method, the ZSDIV-MEC method, and our method. Figure 9 shows the calculation times for user and TPA.

As can be seen from Figure 9(a), with the same number of data blocks, the ASDIV-MEC scheme spends less time on users than the scheme based on BLS, aggregate signatures scheme, and ZSDIV-MEC. This is because an algebraic signature is a signature similar to a hash function, but with relatively low computational complexity. As shown in Figure 9(b), due to the algebraic signatures, our protocol also takes less time than the schemes based on BLS, aggregate signatures scheme, and ZSDIV-MEC. By comparing the computational overhead of the BLS-based method, the aggregate signatures scheme-based method, the ZSDIV-MEC method, and our ASDIV-MEC method, the lightweight performance of the method is further proved.

7.3.4. Comparison of Computational Overhead for Data Dynamics. To verify the low computational overhead of our proposed data dynamics operation, we compare three main data dynamics strategies. Figure 10 shows three data dynamic update schemes. It can be seen from Figure 10 that the computing cost of the tree-based scheme increases exponentially, while that of the other two schemes increases linearly. This is because the tree-based scheme needs to calculate split subtrees, making its cost much higher than the other two.

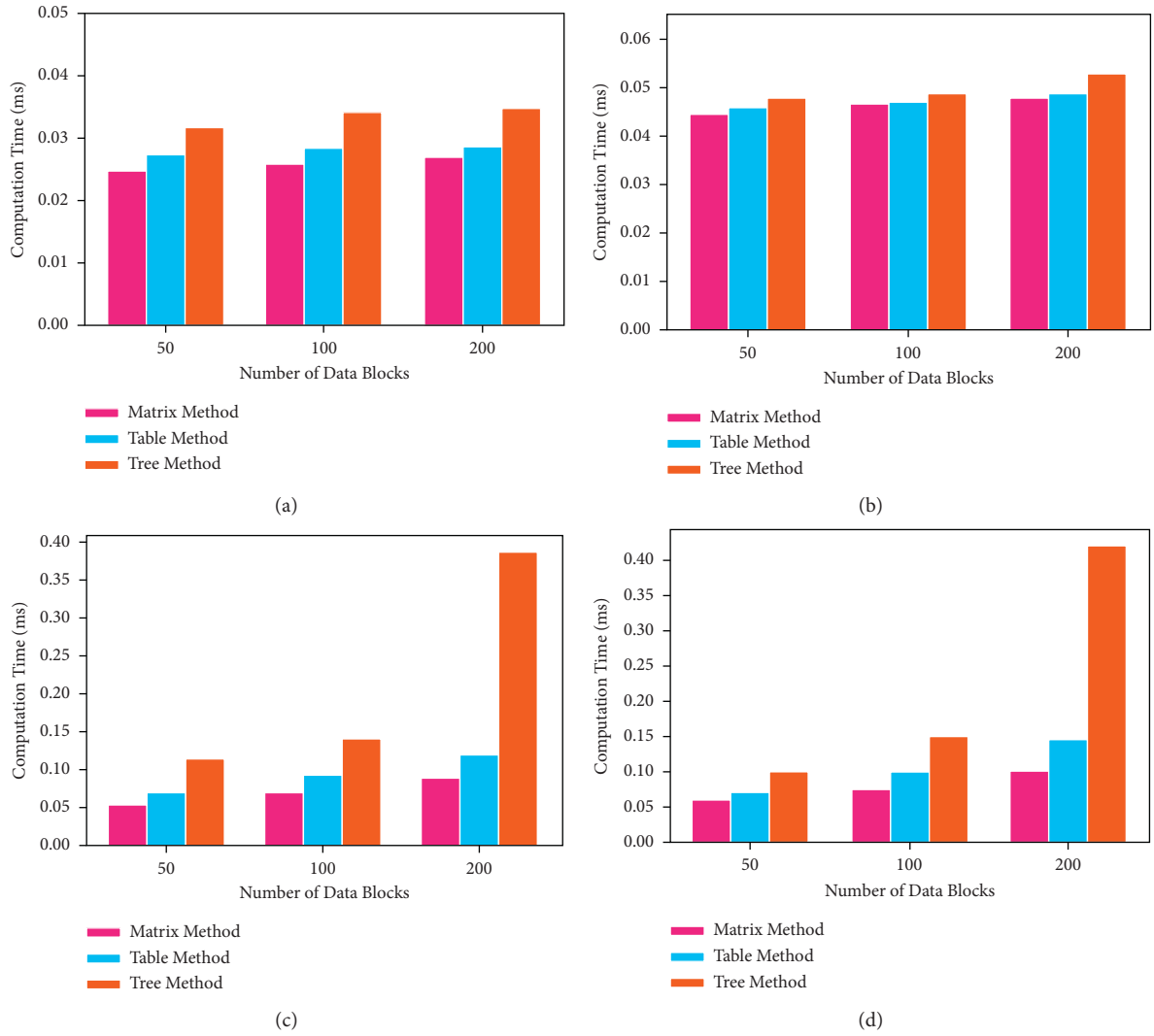


Figure 11: Comparison of computation time in insertion and deletion. (a) Block insertion. (b) Subblock insertion. (c) Block deletion. (d) Subblock deletion.

Further, we simulate the dynamic operation on the MATLAB platform and construct the index matrix M_I in MATLAB, where all the matrix values are 1, and the first column of the matrix is numbered 1 to n in MATLAB. Each operation is implemented three times according to different data blocks. The number of blocks is 50, 100, and 200. In order to reflect the rule of time overhead, we simulate three data block operations with different subblock numbers, namely, 50, 100, and 200.

Figure 11 shows the computing cost of data insertion and deletion in three dynamic operation methods. Figures 11(a) and 11(b) are the insertion operations of blocks and subblocks. The computing cost of the matrix-based method used in our dynamic update strategy is slightly lower than that of the table-based method, while that of the tree-based method is much higher than that of the matrix and table-based method. Figures 11(c) and 11(d) are block and subblock deletion operations. Therefore, the computational cost of dynamic operations based on matrix indexes is the lowest.

8. Conclusion

In this paper, data integrity verification in a mobile edge computing environment is studied. We propose a lightweight and dynamic data integrity verification method in an MEC environment. In order to achieve low latency and acceptable computational overhead, we design a data integrity verification protocol based on algebraic signatures. Through detailed performance analysis, the feasibility, security, and privacy of the proposed method are proved. At the same time, a data dynamic update optimization strategy is proposed to further reduce the computing cost. We have conducted a series of experiments to compare the computational overhead of the proposed method with other

methods at various stages. Simulation results show that the performance of our method is better than the baseline methods.

In future work, we will study how to ensure that the algebraic signatures-based integrity verification method is more secure and efficient, and we will also consider the problem of data integrity verification for multiple mobile users.

Data Availability

The data supporting the results of this study can be obtained from the corresponding author.

Disclosure

This work is extended from publication [5].

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 (61772285), Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, and Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks.

References

[1] Y. Liu, Y. Li, Y. Niu, and D. Jin, "Joint optimization of path planning and resource allocation in mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 9, pp. 2129–2144, 2020.

[2] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[3] Y. Ping, Y. Zhan, K. Lu, and B. Wang, "Public data integrity verification scheme for secure cloud storage," *Information*, vol. 11, Article ID 409, 2020.

[4] D. Liu and J. Shen, "Dang with corrupted data recovery for edge computing in enterprise multimedia security," *Multimedia Tools and Applications*, vol. 79, pp. 10851–10870, 2020.

[5] H. Wang, J. Zhang, Y. Lin, and H. Huang, "ZSS signature based data integrity verification for mobile edge computing," in *Proceedings of the 2021 IEEE/ACM 21st international symposium on cluster, Cloud and Internet Computing (CCGrid)*, pp. 356–365, Melbourne, Australia, May 2021.

[6] H. Zhu, Y. Yuan, Y. Chen et al., "A secure and efficient data integrity verification scheme for cloud-IoT based on short signature," *IEEE Access*, vol. 7, pp. 90036–90044, 2019.

[7] Z. Fan, X. Lin, G. Tan, Y. Zhang, and W. Dong, "One secure data integrity verification scheme for cloud storage," *Future Generation Computer Systems*, vol. 96, pp. 376–385, 2019.

[8] Y. Deswarte, J. J. Quisquater, and A. Saidane, "Remote integrity checking," in *Proceedings of the 6th Working Conf. Integr. Internal Control Inf. Syst. (IICIS)*, pp. 1–11, Fairfax, Virginia; USA, November 2004.

[9] M. Venkatesh, M. R. Sumalatha, and C. Selva Kumar, "Improving public auditability, data possession in data storage security for cloud computing," in *Proceedings of the Int. Conf. Recent Trends Inf. Technol.*, pp. 463–467, Udiapur, India, April 2012.

[10] G. Ateniese, R. C. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," in *Proceedings of the CCS*, pp. 598–610, Alexandria, VA, USA, October 2007.

[11] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proceedings of the ASIACRYPT*, pp. 90–107, Melbourne, Australia, December 2008.

[12] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847–859, 2011.

[13] I. Shen, D. Liu, D. He, X. Huang, and Y. Xiang, "Algebraic signatures-based data integrity auditing for efficient data dynamics in cloud computing," *IEEE Transactions on Sustainable Computing*, vol. 5, no. 2, pp. 161–173, 2020.

[14] Y. Ren, J. Qi, Y. Liu, J. Wang, and G.-J. Kim, "Integrity verification mechanism of sensor data based on bilinear map accumulator," *ACM Transactions on Internet Technology*, vol. 21, Article ID 19, 2021.

[15] J. Z. Lu and H. X. Pan, "An integrity verification scheme of cloud storage for internet-of-things mobile terminal devices"," *Computers & Security*, vol. 92, Article ID 101686, 2020.

[16] H. Wang, D. He, J. Yu, N. N. Xiong, and B. Wu, "RDIC: a blockchain-based remote data integrity checking scheme for IoT in 5G networks," *Journal of Parallel and Distributed Computing*, vol. 152, pp. 1–10, 2021.

[17] D. Yue, Y. Li, Y. Zhang, W. Tian, and Y. Huang, "Blockchain-based verification framework for data integrity in edge-cloud storage," *Journal of Parallel and Distributed Computing*, vol. 146, pp. 1–14, 2020.

[18] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain based data integrity service framework for IoT data," in *Proceedings of the 2017 IEEE International Conference on Web Services (ICWS)*, pp. 468–475, Honolulu, HI, USA, June 2017.

[19] W. Tong, B. Jiang, F. Xu, Q. Li, and S. Zhong, "Privacy-preserving data integrity verification in mobile edge computing," in *Proceedings of the ICDCS*, pp. 1007–1018, Dallas, TX, USA, July 2019.

[20] Y. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *Proceedings of the Int. Workshop Public Key Cryptogr*, pp. 277–290, Berlin, Germany, March 2004.

[21] R. Mokadem and W. Litwin, "String-matching and update through algebraic signatures in scalable distributed data structures," in *Proceedings of the International Workshop on Database and Expert Systems Applications*, pp. 708–711, Krakow, Poland, September 2006.

[22] S. S. J. Schwarz and E. L. Miller, "Store, forget, and check: using algebraic signatures to check remotely administered storage," in *Proceedings of the International Conference on Distributed Computing Systems*, pp. 12–21, Lisboa, Portugal, July 2006.

[23] Y. Luo, S. Fu, M. Xu, and D. Wang, "The Enable data dynamics for alge signatures based on remote data possession checking in cloud storage," *China Communications*, vol. 11, 2014.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 1615596, 9 pages
https://doi.org/10.1155/2022/1615596

Research Article

A Knowledge Representation Method for Question Answering Service in Mobile Edge Computing Environment

Rong Qian and Xia Hou

Beijing Information Science and Technology University, Beijing, China

Correspondence should be addressed to Xia Hou; houxia@bistu.edu.cn

Received 4 November 2021; Revised 11 January 2022; Accepted 20 January 2022; Published 17 February 2022

Academic Editor: Xiaolong Xu

Driven by the rapid development of mobile computing and the Internet of Things, the number of devices connected to Internet has increased dramatically in recent years. Such development has generated massive amounts of data and highlighted the importance and urgency of using the accumulated big data to improve frequently used services. Deploying the question answering service in the mobile edge computing environment is considered a good way to make efficient use of the data and improve user experiences. Powered by the breakthroughs of deep learning technologies, question answering system based on knowledge graph (KBQA) has flourished in recent years. Knowledge representation, as a key technology of KBQA, can express the knowledge graph as the vectors containing more semantic information and thereby improve the accuracy of the question answering system. This paper proposes a knowledge representation method that integrates more features than the traditional methods. In our method, knowledge is represented as a combination of a structured vector reflecting the target triple and the domain information around the entity. By representing richer semantic vectors, our method outweighs TransE, ConvE, and KBAT, in terms of link prediction.

1. Introduction

The explosive growth of smart phones and other mobile terminals and the emergence of many new applications have made a great impact on mobile and wireless networks [1–3]. The traditional centralized network cannot meet the needs of mobile users due to heavy load and long delay [4–6]. Therefore, a new architecture is proposed to open network capabilities from the core network to the edge network, i e., mobile edge computing [7–9]. Mobile edge computing deploys the services and functions originally located in the cloud data center to the edge of the mobile network and provides computing, storage, network, and communication resources at the edge of the mobile network [10, 11].

Question answering system is an advanced form of information retrieval system, which can use accurate and concise natural language to answer users' questions. KBQA is different from other types of question answering systems in that KBQA uses knowledge graphs to provide a highly structured knowledge source for the question answering system. Knowledge graph (KG) is a large scale multi-relationship graph, consisting of entities and their relationships. A few of the existing large knowledge graphs include Freebase, DBpedia, YAGO, and XLORE. Although these knowledge graphs are large in scale, they are far from complete. In order to address this problem, link prediction is proposed. The common way to solve this task is to learn a low-dimensional representation of all entities and relationships and use them to predict new facts, also known as knowledge representation learning [12].

In recent years, many knowledge representation models have been proposed. The most classic model is TransE [13], which has been inspired by Word2vec. TransE can only be used in one-to-one relationships and cannot express polysemous words. Based on TransE, researchers have proposed many extended models. TransH [14] introduces a specific relationship Hyperplane, so that different entities have different representations under different relationships. TransR [15] makes different relationships to have different semantic spaces. Neural networks have also been widely

used recently to build knowledge representation models. R-GCN [16] uses graph convolutional networks [17] (GCN) to represent relationships. ConvKB [18] combines the triple vector into a 3-column matrix and inputs it into the convolutional layer, which is represented by one-dimensional convolution. ConvE [19] uses two-dimensional convolution and multiple nonlinear features to model triples. Whether it is a traditional Trans model or a neural network-based model, each triple is processed independently, and the rich semantic information around the entity cannot be used.

In this paper, we propose KRDGC, a knowledge representation method that integrates multiple features. In order to learn the rich semantic information of nodes in the knowledge graph, KRDGC not only uses the structural information of the triple but also considers the neighborhood information around the entity. KRDGC uses TransD to represent structure information, so that different head and tail entities can be mapped to different relationship spaces according to their own attributes and relationship characteristics. The improved graph attention networks (GATs) can give different weights to adjacent entities according to different relationships between entities. We use improved GAT to obtain the neighborhood information within two hops of the entity. Finally, KRDGC uses the capsule network as a decoder. Our contributions in this paper are as follows:

(1) We propose an end-to-end knowledge representation model KRDGC that combines the advantages of TransD and GAT.

(2) We use the capsule network as a decoder to extract features of the same dimension in multiple feature maps.

(3) We evaluate our KRDGC for link prediction on benchmark datasets FB15K-237 and WN18RR. KRDGC obtains the best mean rank and highest Hits@3.

The rest of the paper is organized as follows. Section 2 provides a review of background. In Section 3, we introduce the classic knowledge representation model. Section 4 reports experimental results and datasets' descriptions followed by our conclusion and future research directions in Section 5.

2. Background

Mobile edge computing provides content storage, computing, and distribution services near the mobile user side through in-depth cooperation with content providers and application developers [20, 21]. This enables applications, services, and content to be deployed in a highly distributed environment to better meet low latency and high bandwidth requirements [22–24]. For our system, the overall deployment framework is shown in Figure 1.

Recently, knowledge representation learning models can be divided into three categories: (1) using the structural information of the triple itself, (2) utilizing external information such as text, images, or rules, and (3) fusion of entity neighborhood information.

Models based on structured information can be divided into two categories, traditional translation models and neural network-based models. Traditional models based on translation include TransE, TransH, TransR, and TransD [25]. These models do not consider any information other than triples. TransH and TransR focus on the multiple representations of entities in different relations, improving the performance on knowledge completion and triple classification. However, both models only project entities according to the relations in triples, ignoring the diversity of entities. To address this problem, TransD proposes a novel projection method with a dynamic mapping matrix depending on both entities and relations, which takes the diversity of entities as well as relations into consideration. Inspired by the fact that concentric circles in the polar coordinate can naturally reflect the hierarchical structure, HAKE [26] was proposed. HAKE can effectively model the semantic levels in the knowledge graph and has a good performance in link prediction tasks.

Models based on neural networks include ConvKB and ConvE, as well as CapsE [27]. CapsE uses the capsule network to model triples. It has a "deep" architecture for modeling the entries in a triple at the same dimension. The number of interactions that ConvE can capture is limited, so InteractE [28] was proposed. InteractE increases the number of interactions between entities and relationships. It proves that increasing the number of interactions can improve the performance of link prediction.

DKRL [29] utilizes text information to model the corresponding triples and entity description information. TKRL [30] makes good use of the hierarchical information of entities. Compared with TransE and TransR, its performance has been improved by 11.3% and 6.2%, respectively. KALE [31] combines knowledge graphs and logic rules and then expresses and models them in a unified framework. This method can make better predictions outside the scope of pure logical reasoning, but the logic rules are more limited. IKRL [32] is the first attempt to combine images with knowledge graphs for KRL. Its promising performances indicate the significance of visual information for KRL. Although the use of information outside the knowledge graph can add semantics to entities, not all entities have access to additional information, and it is not universally applicable.

PTransE [33] is a path-based model. It combines multiple relationships of entities semantically to obtain a vector representation of the path. GAKE [34] defines three contexts with entities and relationships as subjects and uses these kinds of contextual information for modeling. TCE [35] improves on GAKE and proposes two types of context information, path context and neighborhood context. KBAT [36] is a novel attention-based feature embedding model that captures both entity and relation features in any given entity's neighborhood.

In order to make the model more effective, we refer to KBAT as the basic model. On the basis of making full use of entity neighborhood information, combined with triple structured information, the capsule network is used to extract more in-depth information.

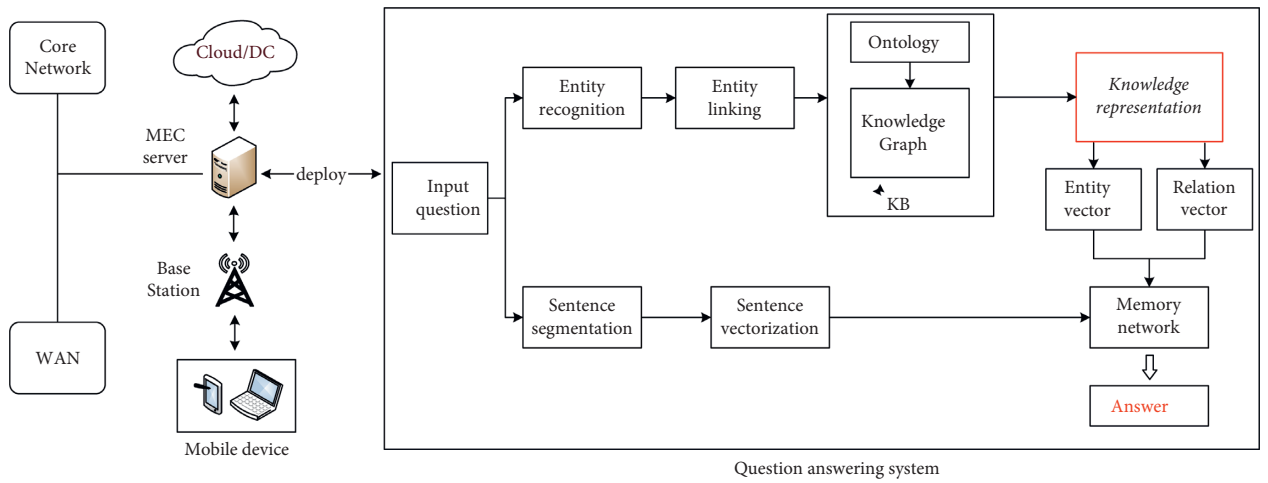


Figure 1: The overall structure and deployment of the question and answering system.

3. The Proposed KRDGC Model

In this section, we describe the proposed method. We first define notations. A knowledge graph $\mathcal{G} \in (\mathcal{E}, \mathcal{R})$ is a collection of valid factual triples in the form of (head entity, relation, tail entity) denoted as (h, r, t) such that $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$ where $\mathcal{E}$ is a set of entities and $\mathcal{R}$ is a set of relations. Embedding model aims to define a score function giving a score for each triple, such that valid triples receive higher scores than invalid triples. The overall structure is shown in Figure 2.

3.1. Structured Features. We use TransD to model the structural features of triples. As illustrated in Figure 3, TransD sets up two projection matrices $\mathbf{M_{rh}}$ and $\mathbf{M_{rt}}$ that, respectively, project the head entity and the tail entity into the relational space. The specific definitions are as follows:

$$\begin{aligned} \mathbf{M_{rh}} &= \mathbf{l_{r_p}} \mathbf{l_{h_p}} + \mathbf{I^{d\times k}}, \\ \mathbf{M_{rt}} &= \mathbf{l_{r_p}} \mathbf{l_{t_p}} + \mathbf{I^{d\times k}}, \end{aligned} \tag{1}$$

where $\mathbf{l_{h_p}}, \mathbf{l_{t_p}} \in \mathbf{R^d}, \mathbf{l_{r_p}} \in \mathbf{R^k}$, and the subscript p represents that the vector is a projection vector. Therefore, the mapping matrices are determined by both entities and relations. Compared with other Trans models, TransD makes the two projection vectors interact sufficiently because each element of them can meet every entry coming from another vector.

3.2. Graph Attention Networks with Relations. In the KG, entities and relationships are not independent, and they all have an impact on each other. The local neighbor nodes of the entity contain a lot of important hidden semantic information. In this paper, we use GAT to extract hidden features in the neighborhood of an entity. The original GAT only considers entities, ignoring edge information. We use the method proposed by KBAT to redefine an attention layer.

The structure of the graph attention network is shown in Figure 4. In order to update the vector of entity e_i, a linear transformation layer is used to learn the vector representation of the combination of entities and relations in a specific triple $t_{ijk} = (e_i, r_k, e_j)$. The corresponding vector after the combination is

$$c_{ijk} = \mathbf{W}_1 [e_i] [r_k] [e_j], \tag{2}$$

where $\mathbf{W}_1$ denotes the linear transformation matrix. Then, we obtain the absolute attention value of the triple through another linear transformation matrix $\mathbf{W}_2$ and the LeakyReLU nonlinearity.

$$b_{ijk} = \text{Leaky Rule}\left(\mathbf{W}_2 c_{ijk}\right). \tag{3}$$

We use softmax to normalize b_{ijk} to get the relative attention value.

$$\begin{aligned} \alpha_{ijk} &= \text{softmax}\left(b_{ijk}\right) \\ &= \frac{\exp\left(b_{ijk}\right)}{\sum_{t\in N_i} \sum_{r\in R_{it}} \exp\left(b_{itr}\right)}, \end{aligned} \tag{4}$$

where N_i is the neighborhood of entity e_i and R_{it} is the set of relations between entities e_i and e_j. The new vector representation of entity e_i is obtained by weighted summation of all neighborhood according to the relative attention value and is stabilized through the multihead attention mechanism.

$$e_i' = \sigma\left(\sum_{m=1}^{2} \sum_{j\in N_i} \sum_{k\in R_{ij}} \alpha_{ijk}^m c_{ijk}^m\right), \tag{5}$$

where m represents the m-th attention head and σ represents any nonlinear function.

In order to keep the relationship dimension and entity dimension consistent, the relationship vector is updated through linear transformation in the GAT. In the last layer, the new vector and the original vector are linearly combined through the weight matrix to prevent the loss of the original information of the entity.

3.3. Capsule Network. After GAT training, we use improved CapsE [27] as a decoder in our model. It uses a three-column matrix to represent each triple. First, we use CNN to perform

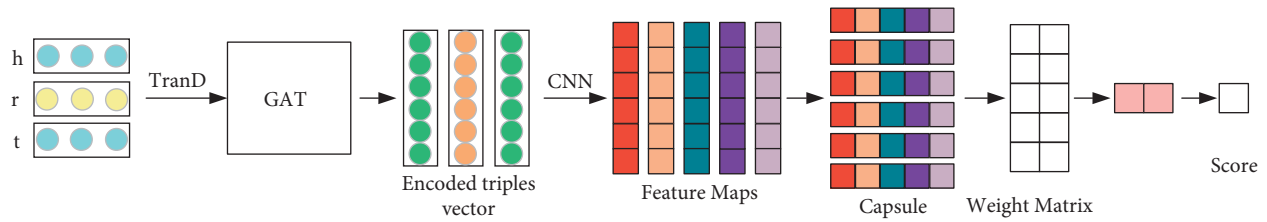


Figure 2: The overall structure of our model.

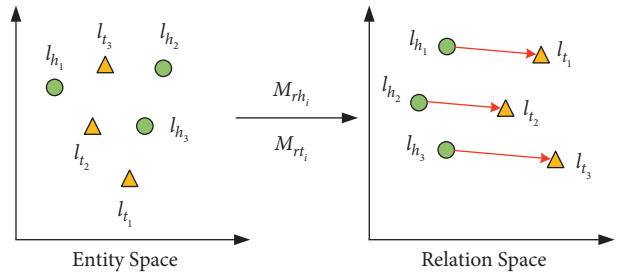


Figure 3: TransD model (entities and relationships' mapping matrix).

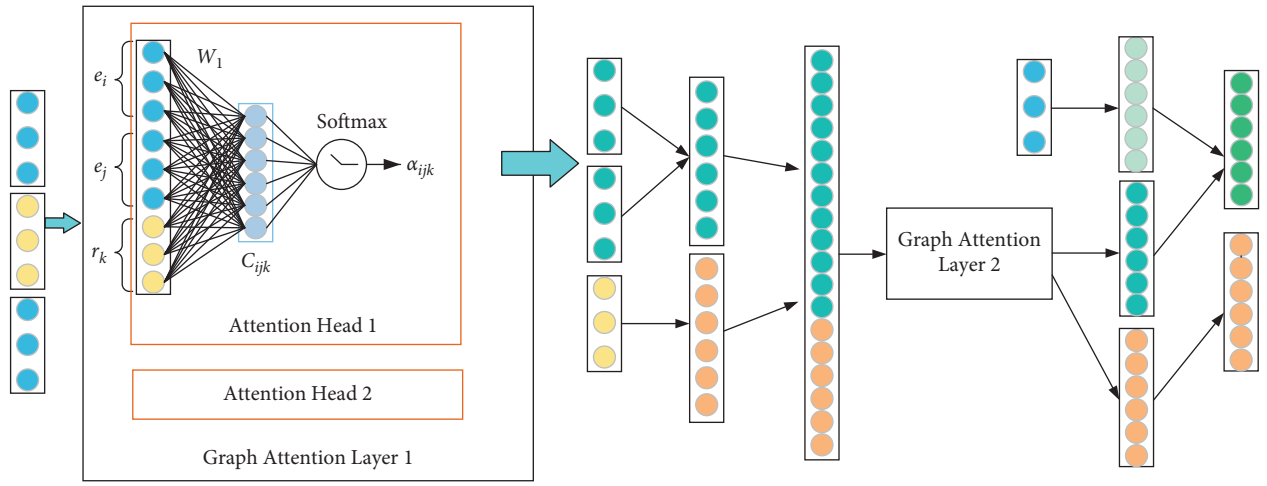


Figure 4: The structure of GAT with relations.

convolution operation on the triple vector to generate multiple different feature maps. All feature maps with the same dimension features are encapsulated into corresponding capsules. Therefore, each capsule can capture the different characteristics of the corresponding dimensions of the embedded triples. The products of these capsules and different weights generate smaller-dimensional capsules, and one continuous vector is obtained. The vector and the weight vector perform dot product operation again to obtain the corresponding score, and the result of the sum of all the scores is used to judge the correctness of the given triple. The score function for the triple is as follows:

$$f(h,r,t) = \left\| \text{caps}\left(g\left(\left[v_h, v_r, v_t\right] * \Omega\right)\right)\right\|, \tag{6}$$

where caps denotes a capsule network operator and g is an activation function. Because we use ReLU in this paper, Ω is the shared parameter in the convolution layer. The model is trained using the loss function as follows:

$$\mathscr{L} = \sum_{(h,r,t)\in\{\mathscr{T}\cup\mathscr{T}'\}} \log\left(1 + \exp\left(\left(-t_{(h,r,t)} \cdot f(h,r,t)\right)\right)\right) + \lambda\|\omega\|_2^2, \tag{7}$$

in which

$$t_{(h,r,t)} = \begin{cases} 1, & \text{for } (h,r,t) \in \mathscr{T}', \\ -1, & \text{for } (h,r,t) \in \mathscr{T}'. \end{cases} \tag{8}$$

The construction of negative triples is to replace the head entity and tail entity of the correct triple with all the entities in the dataset.

4. Experiments and Analysis

4.1. Datasets. In our experiments, we use two widely used benchmark datasets FB15K-237 [37] and WN18RR [19] for evaluation of the performance of link prediction.

FB15K-237 is extracted from Freebase. It contains 14,541 entities and 237 relations. It is an improved version of FB15K dataset where all inverse relations are deleted to prevent direct inference of test triples by reversing train triples. WN18RR is created from WN18, which is a subset of WordNet. WN18 consists of 18 relations and 40,943 entities. Similar to FB15K dataset, all inverse relations are deleted to prevent direct inference of test triples by reversing train triples. WN18RR contains 40,943 entities and 11 relations. Details of the datasets are summarized in Table 1.

4.2. Link Prediction. In the link prediction task, the purpose is to predict a missing entity given a relation and another entity. In a specific experiment, for each triple in the test set, we remove the head or tail entity and then replace it with all the entities in dictionary in turn. We first compute scores of those corrupted triplets and then rank them by descending order; the rank of the correct entity is finally stored. The task emphasizes the rank of the correct entity instead of only finding the best one.

4.3. Evaluation Protocol. We use the filtered setting protocol, i e., not taking any corrupted triples that appear in the KB into accounts. We rank the valid test triple and corrupted triples in descending order of their scores. We employ evaluation metrics: MR, MRR, Hits@1, Hits@3, and Hits@10 (i e., the proportion of the valid test triples ranking in top 1, 3, and 10 predictions).

MR means mean rank, and its specific calculation method is as follows:

$$\mathrm{MR} = \frac{1}{|S|}\sum_{i=1}^{|S|}\mathrm{rank}_i = \frac{1}{|S|}\left(\mathrm{rank}_1 + \cdots + \mathrm{rank}_{|S|}\right), \tag{9}$$

where S is a set of triples, $|S|$ is the number of triple sets, and rank_i refers to the link prediction ranking of the i-th triple. The smaller the indicator is, the better the performance is.

MRR means mean reciprocal ranking. The specific calculation method is as follows:

$$\mathrm{MRR} = \frac{1}{|S|}\sum_{i=1}^{|S|}\frac{1}{\mathrm{rank}_i} = \frac{1}{|S|}\left(\frac{1}{\mathrm{rank}_1} + \cdots + \frac{1}{\mathrm{rank}_{|S|}}\right). \tag{10}$$

The symbols involved in the above formula are the same as those involved in the MR calculation formula. The bigger the indicator is, the better the performance is.

HITS@n refer to the average proportion of triples that rank less than n in link prediction. The specific calculation method is as follows:

$$\mathrm{HITS@}n = \frac{1}{|S|}\sum_{i=1}^{|S|}\mathbb{I}\left(\mathrm{rank}_i \le n\right), \tag{11}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

Lower MR, higher MRR, or higher Hits@1,3,10 indicate better performance. Final scores on the test set are reported for the model obtaining the highest Hits@1,3,10 on the validation set.

4.4. Training Protocol. We first train TransD for 1000 and 3000 epochs on FB15K-237 and WN18RR, respectively. Then, we get a 200-dimensional entity and relationship vectors and use the vectors to initialize entity and relation embeddings in GAT. In the GAT, we use the following hyperparameters for training to select the optimal result. We use Adam learning rate $lr \in \{1e^{-4}, 5e^{-4}, 5e^{-3}, 1e^{-3}\}$, l_1-norm or l_2-norm, margin $\in\in\{1, 3, 5, 7\}$, and dropout $\in\in\{0.1, 0.2, 0.3\}$. The highest Hits@10 and Hits@3 scores and lower MR on the validation set are obtained when we use learning rate at $1e^{-3}$, l_2-norm, margin = 1, and dropout = 0.3 for FB15K-237 and learning rate at $1e^{-3}$, l_2-norm, margin = 5, and dropout = 0.3 for WN18RR.

After GAT, we train capsule network for 200 epochs. We set batch size to 128. We use the Adam optimizer with the initial learning rate $\in\in\{5e^{-6}, 1e^{-5}, 5e^{-5}, 1e^{-4}\}$. We monitor the MRR score after each training epoch and obtain the highest MRR score on the validation set when using the initial learning rate at $5e^{-5}$.

4.5. Results and Analysis. Table 2 compares the experimental results of our model with the common classic methods.

Table 3 shows the comparison between our model and some traditional methods on FB15K-237 and WN18RR. These methods only consider the structure vector of the triples and do not consider the semantic information around the triples.

Table 4 compares the experimental results of our model with previous published results of only using neural network methods.

Compared to the baseline method KBAT, our model outperforms it on FB15K-237 across all the metrics and on three metrics for WN18RR. Figures 5 and 6 show that KRDGC gains significant improvement of 0.527 − 0.518 = 0.009 in MRR (which is 1.7% relative improvement) and 0.662 − 0.626 = 0.036 in Hits@10 (which is 3.6% absolute improvement) on FB15K-237. We confirm previous findings that KBAT in fact is a strong baseline model, e g., KBAT obtains better MRR and Hits@1 than KRDGC on WN18RR.

In Figure 7, KRDGC achieves better performance of 210 − 158 = 52 (which is about 25% relative improvement) and 1940 − 1850 = 90 (which is about 4.6% relative improvement) on MR for FB15K-237 and WN18RR, respectively.

In summary, combining structural information with neighborhood information can capture more semantic information and improve the effect of knowledge representation learning. The capsule network can extract more semantic information in the same dimension of the feature maps.

Table 1: Details of the datasets used.

Category	FB15K-237	WN18RR
#Entities	14,541	40,943
#Relations	237	11
#Train	271,115	86,835
#Valid	17,535	3034
#Test	20,466	3134

Table 2: Experimental result on FB15K-237 and WN18RR.

	FB15K-237					WN18RR				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE	323	0.279	0.441	0.376	0.198	2300	0.243	0.532	0.441	0.043
RotatE	177	0.338	0.533	0.375	0.241	3340	0.476	0.571	0.492	0.428
TuckER	–	0.358	0.544	0.394	0.266	–	0.470	0.526	0.482	0.443
KBAT	210	0.518	0.626	0.54	0.46	1940	0.44	0.581	0.483	0.361
KRDGC	158	0.527	0.662	0.563	0.46	1850	0.424	0.584	0.489	0.34

Table 3: Comparison result with some traditional methods on FB15K-237 and WN18RR.

	FB15K-237					WN18RR				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
DistMult	254	0.241	0.419	0.263	0.155	5110	0.430	0.490	0.440	0.390
ComplEx	339	0.247	0.428	–	0.158	5261	0.440	0.510	–	0.410
HAKE	–	0.346	0.542	0.381	0.250	-	0.497	0.582	0.516	0.452
InteractE	172	0.354	0.535	–	0.263	5202	0.463	0.528	–	0.430
KRDGC	158	0.527	0.662	0.563	0.46	1850	0.424	0.584	0.489	0.34

Table 4: Comparison result with neural network methods on FB15K-237 and WN18RR.

	FB15K–237					WN18RR				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
CapsE	303	0.523	0.593	–	–	719	0.415	0.56	–	–
ConvKB	216	0.289	0.471	0.327	0.198	1295	0.265	0.558	0.445	0.058
ConvE	245	0.312	0.497	0.341	0.225	4464	0.456	0.531	0.47	0.419
SCAN	–	0.35	0.54	0.39	0.26	–	0.47	0.54	0.48	0.43
R-GCN	–	0.249	0.417	0.264	0.151	–	–	–	–	–
KRDGC	158	0.527	0.662	0.563	0.46	1850	0.424	0.584	0.489	0.34

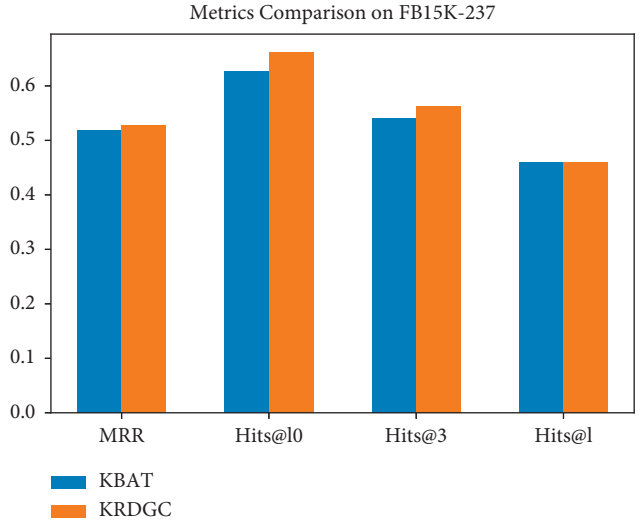


Figure 5: Comparison results of models KBAT and KRDGC on FB15K-237.

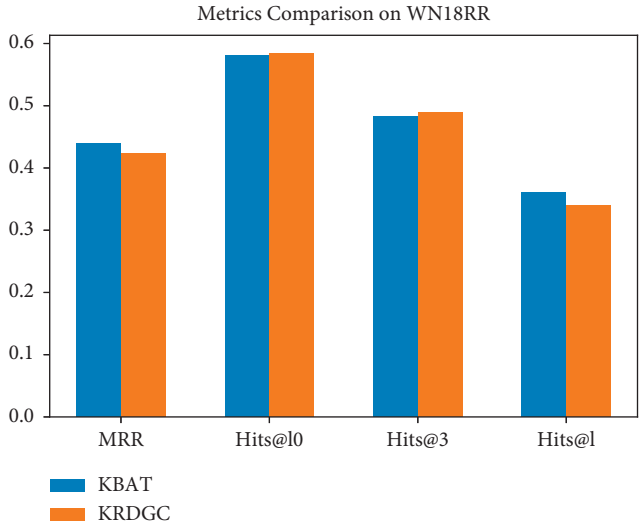


Figure 6: Comparison results of models KBAT and KRDGC on WN18RR.

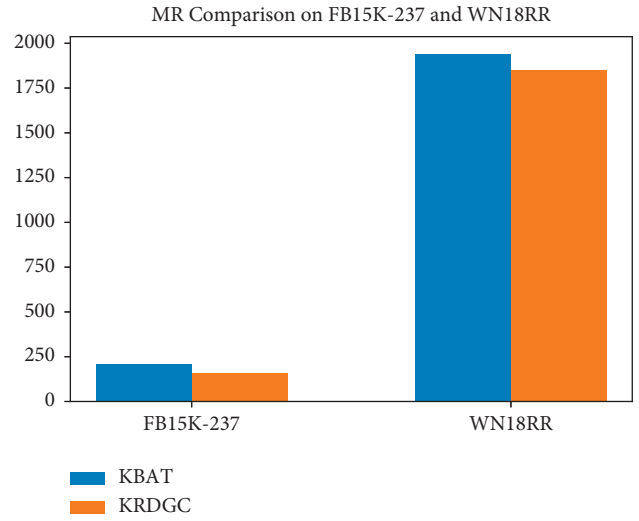


Figure 7: MR comparison on FB15K-237 and WN18RR.

5. Conclusion and Future Work

In this paper, we proposed a KG embedding model KRDGC which is able to take advantage of the structure and neighborhood information of the triple. Our method uses TransD to model structural information and GAT to model neighborhood information and finally extracts deep features through a capsule network. We evaluate our model on link prediction, and the experimental results show significant improvements over the major baselines. In the future, we would like to further represent the relation vector in GAT, incorporate our model into the KBQA system, and verify its response time in mobile edge computing.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This study was supported by the National Natural Science Foundation of China (no. 61602044). The authors are grateful to the research of http://export.arxiv.org/pdf/1808.04122 for providing new ideas.

References

[1] Z. Tong, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[2] R. Bi, Q. Liu, J. Ren, and G. Tan, "Utility aware offloading for mobile-edge computing," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 239–250, 2021.

[3] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, "Dynamic admission control and resource allocation for mobile edge computing enabled small cell network," *IEEE Transactions on Vehicular Technology*, p. 1, 2021.

[4] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "Toffee: task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1634–1644, 2021.

[5] D. Kim, J. Son, D. Seo, Y. Kim, H. Kim, and J. T. Seo, "A novel transparent and auditable fog-assisted cloud storage with compensation mechanism," *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 28–43, 2020.

[6] W. Zhang, X. Chen, and J. Jiang, "A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems," *Tsinghua Science and Technology*, vol. 26, no. 1, pp. 95–111, 2021.

[7] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[8] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[9] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge ai: intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019.

[10] Y. Liu, D. Li, S. Wan et al., "A long short-term memory-based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 135–151, 2021.

[11] Y. N. Malek, M. Najib, M. Bakhouya, and M. Essaaidi, "Multivariate deep learning approach for electric vehicle speed forecasting," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 56–64, 2021.

[12] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: representation, acquisition, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, 2022.

[13] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in Neural Information Processing Systems*, vol. 26, pp. 2787–2795, 2013.

[14] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, pp. 1112–1119, 2014.

[15] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pp. 2181–2187, Austin, TX, USA, January 2015.

[16] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web, ESWC 2018, Lecture Notes in Computer Science*, vol. 10843, pp. 593–607, Springer, 2018.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the International Conference On Learning Representations (ICLR)*, Toulon, France, April 2017.

[18] T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proceedings of the, Conference Of the North American Chapter Of the Association For Computational Linguistics: Human Language Technologies*, vol. 2, pp. 327–333, New Orleans, LA, USA, June 2018.

[19] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence. AAAI*, pp. 1811–1818, Toronto, Canada, February 2018.

[20] J. Mabrouki, M. Azrour, G. Fattah, D. Dhiba, and S. E. Hajjaji, "Intelligent monitoring system for biogas detection based on the internet of things: mohammedia, Morocco city landfill case," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 10–17, 2021.

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

[22] Y. Chen, F. Zhao, Y. Lu, and X. Chen, "Dynamic task offloading for mobile edge computing with hybrid energy supply," *Tsinghua Science and Technology*, 2021.

[23] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1145–1153, 2021.

[24] J. Huang, Y. Lan, and M. Xu, "A simulation-based approach of qos-aware service selection in mobile edge computing," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–10, 2018.

[25] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix,"vol. 1, pp. 687–696, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 1, Long Papers, Beijing, China, July 2015.

[26] Z. Zhang, J. Cai, Y. Zhang, and J. Wang, "Learning hierarchy-aware knowledge graph embeddings for link prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3065–3072, New York, NY, USA, Februray 2020.

[27] D. Q. Nguyen, T. Vu, T. D. Nguyen, and D. Q. Nguyen, "A capsule network-based embedding model for knowledge graph completion and search personalization,"vol. 1, pp. 2180–2189, in *Proceedings of the Conference Of the North American Chapter Of the Association For Computational Linguistics: Human Language Technologies*, vol. 1, Long and Short Papers, Minneapolis, MA, USA, June 2019.

[28] S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, and P. Talukdar, "Interacte: improving convolution-based knowledge graph embeddings by increasing feature interactions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3009–3016, Beijing, China, April 2020.

[29] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, pp. 2659–2665, 2016.

[30] R. Xie, Z. Liu, and M. Sun, "Representation learning of knowledge graphs with hierarchical types," in *Proceedings of the IJCAI*, pp. 2965–2971, NY, USA, July 2016.

[31] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Jointly embedding knowledge graphs and logical rules," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 192–202, Austin, TX, USA, November 2016.

[32] R. Xie, Z. Liu, H. Luan, and M. Sun, "Image-embodied knowledge representation learning," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia, August 2017.

[33] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases," in *Proceedings of the 2015 Conference On Empirical Methods In Natural Language Processing*, pp. 705–714, Lisbon, Portugal, September 2015.

[34] J. Feng, M. Huang, Y. Yang, and X. Zhu, "Gake graph aware knowledge embedding," in *Proceedings of the COLING the 26th International Conference on Computational Linguistics Technical Papers*, pp. 641–651, Osaka, Japan, December 2016.

[35] J. Shi, H. Gao, G. Qi, and Z. Zhou, "Knowledge graph embedding with triple context," in *Proceedings of the 2017 ACM on Conference On Information And Knowledge Management*, pp. 2299–2302, Singapore, September 2017.

[36] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," in *Proceedings of the 57th Annual Meeting Of the Association For Computational Linguistics*, pp. 4710–4723, Florence, Italy, August 2019.

[37] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1499–1509, Lisbon, Portugal, September 2015.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 5019584, 18 pages
https://doi.org/10.1155/2022/5019584

Research Article

Towards Optimal Resources Allocation in Cloud Manufacturing: New Task Decomposition Strategy and Service Composition Model

Zhou Fang, Qilin Wu, and Dashuai Guan

College of Information Engineering, Chaohu University, Chaohu/238024, China

Correspondence should be addressed to Qilin Wu; qlw@chu.edu.cn

Received 2 November 2021; Accepted 4 January 2022; Published 15 February 2022

Academic Editor: Xiaolong Xu

Service composition optimization is one of the core issues in cloud manufacturing research. However, all current studies of service composition in cloud manufacturing assume that tasks have been decomposed into subtasks, so they can be directly mapped to existing services. However, due to the complexity, diversity, and multilevel of services in cloud manufacturing, services have different granularity. Therefore, the matching between tasks and services does not always occur at the lowest level. For solving the problem of discontinuity between task decomposition and service composition, this paper considers the characteristics of existing services in the cloud pool and proposes a task decomposition strategy based on task/service matching on the basis of refining the description model of tasks and services. Then, for the decomposed subtask set, the E-CARGO model is used to model the optimal composition process of services, and CPLEX is used to solve the model. Practical cases show that the proposed task decomposition strategy can solve the problem of discontinuity between task decomposition and service composition without relying on more expert systems. In addition, the proposed service composition model is more flexible, can easily model more variable factors, and CPLEX can solve the model more quickly and stably.

1. Introduction

Recently, with the promotion of emerging technologies such as cloud computing, Internet of things (IOT), network physical system (CPS), big data analysis, and artificial intelligence, a new industrial manufacturing mode with the core characteristics of globalization, personalization, digitization, cloud computing, collaboration and integration, namely cloud manufacturing, has been proposed [1–7]. As an emerging manufacturing model, cloud manufacturing can effectively solve the problems of shortage and idleness of manufacturing resources, deficiency, and excess of manufacturing capacity in China's manufacturing industry, and realize a manufacturing-oriented service model characterized by sharing, collaboration and on-demand use, which can provide a new impetus for the transformation and upgrading of manufacturing industry [8, 9].

Up to now, cloud manufacturing has attracted a large number of scholars around the world, and many problems that need to be solved in the future are proposed and discussed, such as the architecture of cloud manufacturing service platform, business model, description of manufacturing resources, optimal allocation of resources, etc. [9, 10]. Among these, optimal allocation of manufacturing resources is the core function of cloud manufacturing, mainly including two core processes: task decomposition and service composition.

Task decomposition is the basis and premise of service composition. Its goal is to obtain a highly cohesive task sequence to ensure that different service providers can cooperate to fulfill the customers' requirements. However, in the current research, task decomposition and service composition are usually carried out as two independent and irrelevant procedures. On the one hand, task decomposition heavily relies on industry expert systems and lacks consideration of the existing services' status in the cloud pool, which makes it difficult to keep up with the changing market demand; on the other hand, the existing service composition research lacks the modeling of collaboration between services, which has a certain impact on the overall execution

efficiency of manufacturing tasks. Therefore, how to integrate task decomposition and service composition together is the core issue of this paper.

As an intermediate procedure between task decomposition and service composition, service matching is mainly used to connect manufacturing tasks and services to maximize the satisfaction of both supply and demand. How to effectively use service matching to solve the problem of discontinuity between task decomposition and service composition is one core problem to be solved in this paper. In addition, the E-CARGO model, which was proposed by Professor H. B. Zhu in 2006, is used to describe a role-based collaboration system, and how to use it to model the collaboration between services to improve the practicability of the service composition model is another problem to be solved in this paper.

The main contributions of this paper are as follows:

(1) According to the characteristics of candidate service sets obtained after task service matching, specific computable formulas are given for the internal competition within candidate service sets and the collaboration and dependence between candidate service sets.

(2) Considering the service state, a new task decomposition algorithm based on task/service matching is proposed, which can combine the two processes of task decomposition and task/service matching and reduce the dependence on the expert system.

(3) A new cloud manufacturing service composition model based on role collaboration is constructed, where the mapping relationship between the service composition problem and the E-CARGO model is established. This can not only solve the problem that the heuristic algorithm is easy to fall into local optimization but also facilitate the introduction of multiple variable factors to expand and optimize the model.

The remainder of this paper is organized as follows. Related work is described in Section 2. Section 3 describes the core problems related to the optimal allocation of manufacturing resources. Section 4 gives the description model of tasks and services, introduces the task/service matching based task decomposition algorithm, and describes the subtask reorganization algorithm combined with service characteristics. Section 5 formalizes the service composition approach by utilizing the E-CARGO model and the corresponding calculation methods. Section 6 analyzes and verifies the proposed methods through application case and performance analysis. Finally, the conclusions appear in Section 7.

2. Related Work

As the core part of implementing cloud manufacturing, manufacturing resource optimized allocation aims to provide a set of capabilities/services for satisfying personalized manufacturing demands through a process of resource composition and optimal selection. Efficient shared manufacturing resource allocation can not only achieve rapid response to diverse manufacturing demands but also facilitate full-scale sharing of enterprises' resources.

Task decomposition is one of the most important preprocessing stages of manufacturing resource optimized allocation while is a challenging task, not only because they are characterized by cross-industry heterogeneity, but also because the decomposition process need considering the state of service (such as service granularity and relevance).

In the existing research on task decomposition, some methods used a design structure matrix (DSM) to express the interaction information and correlation degree between tasks [11–13]. Kherbachi et al. used DSM to cluster the tasks in product development and matched the corresponding development tasks to the appropriate research group and development group [14]. Liu and Zhou proposed a method of task decomposition and reorganization based on DSM combined with adjustable task granularity [15]. However, DSM has shortcomings in both the quantitative analysis ability of uncertain information and the decomposition ability of complex tasks. In [16], Shriyam et al. proposed an approach based on a dynamic grid for decomposing exploration tasks among multiple Unmanned Surface Vehicles (USVs) in port regions. In other ways, the task is decomposed into subtasks with appropriate granularity according to the task hierarchy and task correlation. In [17], Zhang et al. constructed a global manufacturing business process network (GMBPN) according to the input and output relationships of manufacturing business activities. Based on the GMBPN, they further presented a two-phase decomposition algorithm. Hu et al. divided the complex manufacturing tasks into multiple stages according to the attributes and characteristics of the production process and proposed a novel hybrid method combining depth first search, fast modular, and artificial bee colony to optimize multistage production processes [18]. To solve complex parts machining problems in CMfg, Guo et al. presented a machining task decomposition strategy that uses features of the complex part as task granularity [19]. Liu et al. proposed an ordered task decomposition method (task decomposition method based on hierarchical task network) considering task granularity, cohesion, and correlation [20].

The above literature studies the task decomposition strategy from different levels and perspectives, but they only use the characteristics of the task itself or the correlation between tasks to decompose the task and take no consideration of the service state, which results in the problem that the process of task decomposition is divorced from that of matching and assignment of task and service. To solve the above problems, Yi et al. [21] decomposed the manufacturing task into atomic tasks according to the predefined decomposition rules and then reorganized these atomic tasks using clustering algorithms by considering task correlation, matching degree of tasks and services, and competition between services. Although the algorithm considers the state of the service, it inverts the dependency between the task and the service. On the one hand, the decomposition process of atomic tasks will rely heavily on

expert systems, and then it is difficult to realize a comprehensive expert system for massive heterogeneous tasks in different industries and categories. On the other hand, even if atomic subtasks can be successfully decomposed, there is a high probability that atomic subtasks cannot match the appropriate candidate service set.

Service composition is another core function in the process of resource optimized allocation. Its main purpose is to select a group of optimal service combinations from the candidate services of each subtask to complete the demander's manufacturing task. In essence, this process is the process of combining multiple services (atomic services or composite services) into value-added services to complete one or a group of tasks. As a typical multiobjective and multiconstraint NP-hard problem, a large number of metaheuristic algorithms, such as genetic algorithm, particle swarm optimization algorithm, and ant colony algorithm, have been proposed to find the optimal or nearly optimal combination scheme in a reasonable time [22–31]. The typical processes of these algorithms are (1) propose heuristic algorithms for fixed models and (2) repeatedly test and adjust the heuristic algorithms to obtain the required performance. If some aspects of the service (such as service availability or service quality) change, this process must usually be repeated. It can be seen that these algorithms have a complex design process and are usually for specific problems. Therefore, they are not adaptive to the dynamic environment, which means that when the environment changes, they may need to be redesigned.

In addition to devising these algorithms, another important problem is how to establish an appropriate service composition model. As an important index to determine the quality of service composition and an important factor to be considered in the process of service composition, Quality of Service (QoS) is widely used in the modeling of service composition. Therefore, Que et al. [30] proposed the method of using the user model (M2U) to solve service composition and optimization selection and established the corresponding mathematical evaluation model by comprehensively considering the four QoS evaluation indexes (time, cost, reliability, and capability). Li et al. proposed an extended Gale-Shapley (GS) algorithm for service composition that allows the generation of multiple service composition solutions effectively, where the requirements with different constraints have been considered [31]. Considering the one-to-one mapping between basic services and subtasks, Liu and Zhang [32] freely combined multiple basic services with equivalent functions into a cooperative service group (SESG) to complete each subtask together. At the same time, the optimized structure of SESG was introduced into the QoS evaluation model, and the corresponding QoS evaluation formula was given. In [33], Jin et al. proposed a correlation-based service description model to describe the QoS dependence of a single service on other related services and then introduced a service correlation mapping model to automatically obtain the value of QoS correlation between services. Afterward, Laili et al. [34] studied the multistage integrated scheduling problem of hybrid tasks in a cloud manufacturing environment to maximize production efficiency while balancing different production task orders. The experimental results show that this method reduces the production cost and shortens the production time. In [23], Yuan et al. proposed six basic QoS indexes including time, composability, quality, availability, reliability, and cost, and determined the weight of each index value in the QoS model by using the improved fuzzy comprehensive evaluation method.

Because services and tasks in cloud manufacturing have different granularity, service composition is essentially a dynamic matching process between multigranularity tasks and services. However, most of the current research on the service combination focuses on QoS modeling and rarely considers how to add more practical constraints in practical applications to easily expand the existing models, such as cooperation and competition between related services.

3. Problem Description

Cloud manufacturing software service platform needs to solve the problems of low sharing rate of manufacturing resources, poor collaboration among enterprises, and low customization level of manufacturing solutions in the manufacturing process. Figure 1 shows its operation principle, and it mainly includes three types of user roles: resource providers, resource demanders, and platform operators. Resource providers describe and publish the manufacturing resources (which will be encapsulated as services) in the manufacturing process in a unified model; resource demanders submit their manufacturing requirements (which can also be called tasks) and access various manufacturing resources on demand with the support of the platform; platform operators mainly audit and manage the resources and requirements in the platform, release or update various templates of resources and requirements in time, and monitor the transactions between the suppliers and the demanders.

Optimal allocation of manufacturing resources is an important procedure in cloud manufacturing, and its basic workflow is shown in Figure 2:

(1) Decomposing the total tasks submitted by resource demanders into subtasks for collaborative completion.

(2) According to the task-service matching method, the cloud manufacturing software service platform searches and matches the candidate service sets that can complete each subtask of the initial total tasks.

(3) Considering the influence factors of QoS, such as time, cost, quality, reliability, and so on, the cloud manufacturing software service platform searches for the best services in the candidate service set of each subtask to form an optimal execution plan for the initial total manufacturing task.

(4) Executing and supervising the completion process of tasks according to the optimal execution plan.

Task decomposition is the most basic initialization step of optimal allocation of manufacturing resources, and its

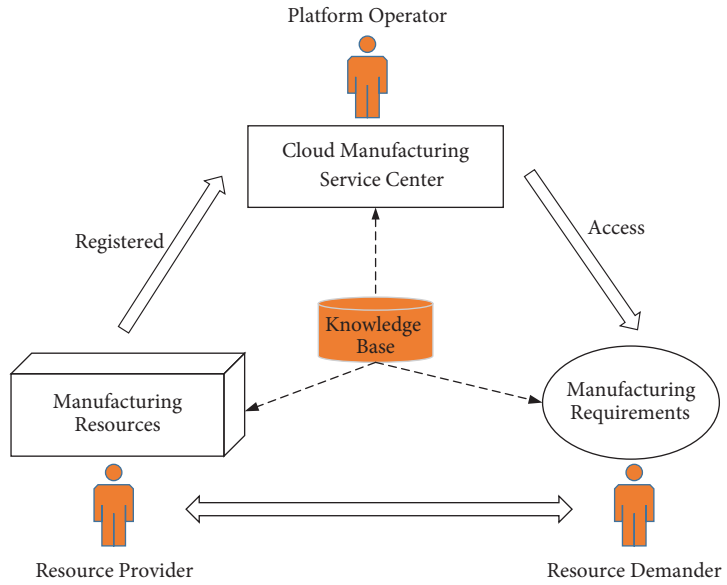


Figure 1: Operation principle of cloud manufacturing software service platform.

purpose is to decompose the overall manufacturing task into subtasks that should be executable and have low relevance between each other so that the cloud manufacturing software service platform can match the appropriate service to complete the user's demand. It is clear that the task decomposition results are closely related to the processing quality of the subsequent procedure of resource optimization.

In the process of task decomposition, the scale of subtasks is characterized by task granularity, which will directly affect the quality and progress of task collaboration. Specifically, the larger the granularity, the higher the task integrity, but the implementation is complex, which is not conducive to multienterprise cooperation. On the other hand, the finer the granularity, the more interaction between tasks, but the coordination is difficult, and the logistics cost and information interaction cost are prominent, which affect the quality, progress, and cost of task completion. Therefore, how to combine the characteristics of the services in the cloud pool and complete the decomposition of tasks at an appropriate granularity is of great significance in resource optimization.

After matching the manufacturing service set for each manufacturing subtask under the functional constraints, some suitable services need to be selected from each candidate set and assembled into composite services in a certain order to collaboratively complete the user's manufacturing requirements. How to build a more flexible combination model, which can easily model more variable factors to adapt to the open and dynamic manufacturing environment, is one of the urgent problems to be solved.

4. Task Decomposition Strategy

In this paper, task decomposition is divided into two stages: preliminary decomposition and reorganization. In the preliminary task decomposition stage, the total task is decomposed into executable atomic subtasks based on task/service matching. In the task reorganization stage, subtasks with the small granularity are merged into subtasks with appropriate granularity by considering the internal competition of candidate service setting, cooperation, and dependence between candidate service sets.

4.1. Description Model of Tasks and Services. Cloud manufacturing users come from different enterprise and engineering application fields, and their needs are more diversified and personalized. In addition, manufacturing resources are widely distributed in various forms and types. The description of requirements and resources by manufacturing enterprises is often unclear, incomplete, and inconsistent, which is difficult to realize the dynamic cooperation between users and resources in the cloud manufacturing environment. Therefore, a unified formal description is inevitable. Here, the requirements are modeled as manufacturing tasks, and the combination of several resources is modeled as manufacturing services to complete the specified tasks. Based on the formal description of the existing manufacturing services and manufacturing tasks (cloudservice = {ID, TypeInfo, BaseInfo, ResourceInfo, FuncInfo, AssessInfo, StatuInfo}) [35], we further extend the description of the function information FunInfo = {FunProfile, InputParam, OutputParams}, where we have the following:

(1) Funprofile: function summary, which briefly describes the functions provided by the service.

(2) Inputparams: the input information of the function, indicating that the demand side needs to provide necessary information or materials during the implementation of the service. For example, for a service of manufacturing a special wrench, the demand side may need to provide corresponding design drawings.

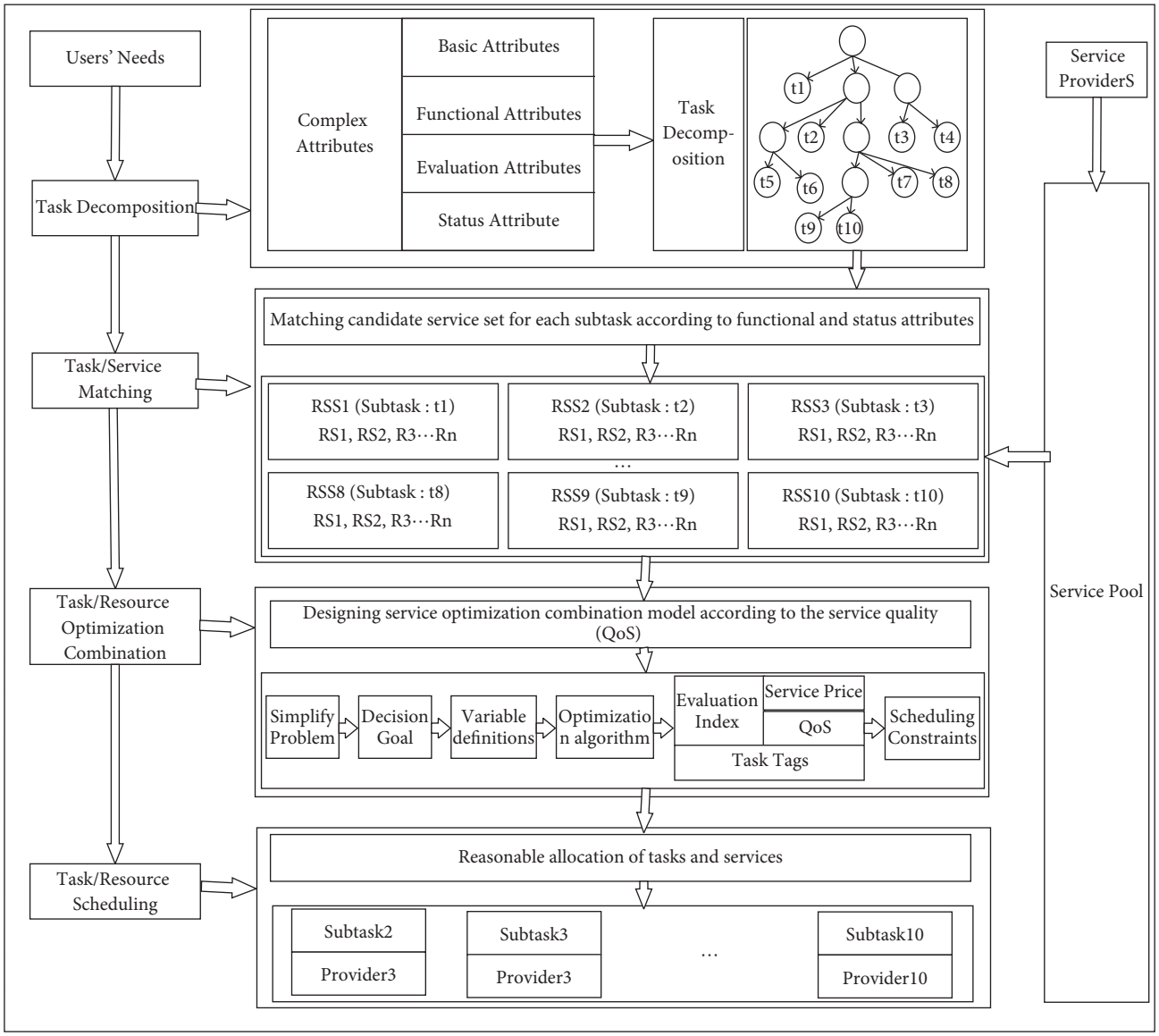


Figure 2: Basic process of resource optimization in cloud manufacturing software service platform.

(3) Outputparams: the output content of the function, which indicates the service results to be provided to the demander after the service is executed. As for a software development service, the output content may be executable source code or a one-year technical support service.

4.2. Preliminary Decomposition Method. To avoid relying too much on the industry knowledge and prevent the problem that the decomposed subtasks cannot match any appropriate services, we integrate the task decomposition and task-service matching as a whole, where we use the existing services in the cloud pool to adaptively complete the preliminary task decomposition. The basic flow of the method is shown in Algorithm 1.

Step 1. Initialize the variable T_k^i, where k represents the k-th subtask waiting to be decomposed, and i is the i-th subtask of the k-th subtask. In particular, when $k=0$, T_k^i means the original task T, and when $i=0$, T_k^i means the k-th original subtask which has not been decomposed. RSS_k^i is the corresponding candidate service set of T_k^i.

Step 2. Match the service set that meets the output requirements of T_k^i in the cloud pool. If the service set can be matched, proceed to Step 3, otherwise, proceed to Step 5.

The matching methods are as follows:

(1) Use TF-IDF algorithm to find out the keywords of task T_k^i and service j $(0 \le j \le \mathrm{N})$ respectively, where N represents the number of all services in the cloud pool that have not yet participated in the matching.

(2) Select several keywords from task T_k^i and service j respectively, and merge them into set D.

(3) Calculate the word frequency of task T_k^i and service j relative to all words in set D in turn.

Input: task waiting to be decomposed T
Output: sub-tasks T_k^i the corresponding candidate service set RSS_k^i
(1) initialize k to 0, i to 0, n to 0
(2) label T as T_k^i
(3) match the service set RSS_k^i in the cloud pool that meets the output requirements of T_k^i
(4) **if** RSS_k^i is not empty
(5) save T_k^i and RSS_k^i
(6) calculate the count n of the intersection of inputs of RSS_k^i, and label it as InSect
(7) set the input of the T_k^i to InSect
(8) **if** n is not equal to 0
(9) set $k+1$ to k
(10) for epoch = 1, 2, . . ., n do
(11) set $i+1$ to i
(12) create a new subtask T_k^i whose output of Fun-Info is the i-th value of InSect
(13) **goto 3**
(14) **end for**
(15) set i to 0
(16) **else:**
(17) **goto 11**
(18) **end if-else**
(19) **else**:
(20) **if** k equal to 0 and i equal to 0
(21) end algorithm
(22) **else**:
(23) **goto 11**
(24) **end if-else**
(25) **end if-else**

Algorithm 1: Preliminary decomposition algorithm.

(4) Calculate and save the cosine similarity $S_{i,j}$ between task T_k^i and service j.
(5) Sort all $S_{i,j}$ and calculate the average value of $S_{i,j}$ ranking in top M. If the average value is greater than the threshold C, the matching is regarded as successful, and the top m services are the corresponding service candidate set; otherwise, the matching is regarded as failed, where C and M are constants.

Step 3. Save task T_k^i and the corresponding service set RSS_k^i and calculate the intersection of inputs of services in RSS_k^i, which can be labeled as InSect, and then set the InSect to be the input of T_k^i.

Step 4. Create a new subtask using the content of the InSect to be the output, and then repeat Steps 2 to 4 to obtain subtask T_{k+1}^i and the corresponding service candidate set RSS_{k+1}^i, where $i\varepsilon[1,n]$ and n is the number of InSect.

Step 5. Judge T_k^i is the original task T; if it is, the algorithm is terminated; otherwise, proceed to Step 4.

4.3. Subtask Reorganization Algorithm. In the preliminary task decomposition stage, only the function matching of the service is considered, and the original task is decomposed into executable subtasks with smaller granularity, without considering the competition within the candidate set, the dependence between candidate sets and the difficulty of collaboration, which will be detrimental to the collaborative work between the final services.

Definition 1. Internal competitiveness of candidate service sets. It refers to the relative number N of services available in the service candidate set corresponding to each subtask after preliminary decomposition. To preserve the competitiveness between services, formula (1) can be used for calculation in the actual calculation process:

$$N = \frac{\sum_s \mathrm{similarity}_i}{S}, \tag{1}$$

where S is the number of services in the candidate service set (that is, in the preliminary task decomposition stage, the service candidate set with the matching degree in the top s), $\mathrm{similarity}_i$ indicates the matching degree between the ith service in the candidate set and the corresponding subtask. The higher the N, the more competitive the candidate service set.

It should be noted that in the cold start stage of the system, the number of services in the cloud pool is less, so the value of S can be set smaller. As the number increases, the value of S can be gradually increased, but the increased value should also be weighed against the efficiency of the algorithm.

The competitiveness of a candidate service set is to ensure that when a resource is selected, we can quickly find a substitute when it is unable to provide services due to unexpected circumstances. From a long-term perspective,

reserving competition space will make the pricing given by manufacturing resource owners more reasonable and urge them to actively improve service quality so as to promote the healthy development of the cloud manufacturing platform [21].

Definition 2. The degree of dependency between services R, which can be expressed by the correlation between services. Considering that the correlation is mainly determined by the logistics correlation and information exchange correlation between them, to reduce the complexity of calculation, the number, and type of inputs of task dependencies can be used as parameters to calculate the dependence between service candidate sets, as shown in

$$R = \sum_{i}^{M} w_i \times Q_i, \tag{2}$$

where M is the number of categories of inputs, Q_i is the number of categories i, w_i $(0 \le w_i \le 1)$ is the correlation coefficient of category i. The actual value of w_i is determined by the expert evaluation method and will be modified according to the operation results during the daily operation of the platform. The smaller the granularity of task decomposition, the greater the dependency.

Definition 3. Coordination difficulty between services. It refers to the difference between the longest and shortest service time (i.e., the execution waiting time of dependent tasks), which can be calculated by

$$T_i = \sum_{N} \left(t_{\max} - t_{\min}\right), \tag{3}$$

where T_i is the coordination difficulty of the i-th task, N is the number of layers of all subtasks of the i-th task, $t_{\max}$ and $t_{\min}$ represent the maximum execution time and minimum execution time of subtask in the specified level, respectively. Obviously, the larger the granularity of task decomposition, the more difficult it is to cooperate.

Definition 4. The granularity G of the candidate service set, which indicates the suitability of the granularity of the candidate service set to complete the specified task. According to the comprehensive analysis formulas (2) and (3), the higher the interdependence of the candidate service sets, the more unfavorable it is to complete the tasks in collaboration. That is, with the increase of R value, the G value should be appropriately increased (i.e., subtasks should be merged). However, the larger the decomposition granularity, the more waiting time for other parallel tasks, which is more unfavorable for the system to complete the task, that is, with the increase of T value, the g value should be appropriately reduced (that is, the task should be decomposed), so it can be expressed by

$$G = N \times R - (1 - N) \times T. \tag{4}$$

It should be noted that when we calculate G using formula (4), the values of N, R, and T need to be regularized.

Definition 5. The state tree of the candidate service set, which is used to simplify the description of the task reorganization process. The node in the tree is the meta-task obtained after the preliminary decomposition of the task. The value of the node represents the internal competitiveness of the service candidate set corresponding to the node, and the weight of edges represents the granularity of the candidate set relative to the parent node.

With the purpose of increasing the internal competitiveness of the candidate service set, reducing the degree of dependency between candidate service sets, and reducing the waiting time of the parallel services in candidate service sets, we design the pruning algorithm as shown below to realize task reorganization (Algorithm 2).

Step 6. Calculate the internal competition of each candidate service set, the dependencies between each candidate service set and others, and the coordination difficulty of each candidate service set, so as to construct the state tree of the candidate resource set using formula (4).

Step 7. Traverse all nodes in the m-th layer of the candidate service set state tree to determine whether the subnodes under the node need to be merged.

(1) Set $k = k + 1$ and repeat steps (1)–(3) if the k-th node in the m-th layer is a leaf node; otherwise, go to step (2).

(2) Get the value nodeVal of the k-th node in the m-th layer; if the value is less than zero, set k to $k + 1$ and return back to step (1); otherwise, go to step (3).

(3) Crop all child nodes under the k-th node, recalculate the node values in the new state tree, set $k = k + 1$, and repeat steps (1)–(3).

Step 8. Set $m = m + 1$ and go back to Step 7.

5. Service Composition Model

5.1. Problem Description. For the original manufacturing task T submitted by the customer, after task decomposition, the subtask list is obtained as shown in Table 1, where the "number of acceptable services" represents the number of service providers that can be accepted by the demander, In the specific implementation process, considering that some subtasks may be complex, multiple services are required to complete them.

Considering that there may be a large number of manufacturing tasks of the same category with the same or similar input and output in the cloud manufacturing software service platform, it is necessary to consider the composition quality of all tasks of the same category while building the optimal composition model of services. Let subtask T_k belong to category TC_k. Therefore, the service candidate set RSS_k matching task T_k also can match all other tasks under category TC_k. Afterward, according to the given QoS evaluation model, we can calculate the competency of each service in the candidate set RSS_k for each task under

Input: atomic subtasks and the corresponding candidate service sets
Output: subtasks with suitable granularity and the corresponding candidate service sets
(1) initialize k to 0, m to 0, nodeVal to −1
(2) construct the state tree of candidate resource set
(3) set m to the count of the original subtasks
(4) **for** epoch = 1, 2, . . ., m do
(5) set k to the count of the sub-tasks of the m-th original subtasks
(6) **for** epoch = 1, 2, . . ., k do
(7) **if** sub-task T_m^k is not a leaf node
(8) get the value of nodeVal
(9) **if** nodeVal is greater than 0
(10) crop and recalculate the state tree
(11) **end if**
(12) **end if**
(13) **end for**
(14) **end for**

Algorithm 2: Subtask reorganization algorithm.

Table 1: List of subtasks of task T.

Subtasks	T_1	T_2	...	T_k	...
Number of acceptable services	3	2	...	1	...

category TC_k, as shown in Table 2. The combination target is to find a service combination with the highest competency, which means the sum of competencies of each subtask of the manufacturing task T is the largest, and ensure all other tasks under each category TC_k have the highest competency at the same time.

It should be noted that the competency value in Table 2 needs to be calculated under a given QoS model according to the evaluation data of the specific evaluation system in the cloud manufacturing service platform. As each attribute of QoS has different measurement methods and dissimilar units, the aggregated QoS values for each attribute should be normalized before evaluating the global QoS of the cloud manufacturing service. Each attribute is either a positive or a negative factor (for a negative factor, the smaller the value of the index, the better for the service requesters and vice versa). This can be obtained using the following equations:

$$F_n = \begin{cases} \dfrac{f_n - \min f_n}{\max f_n - \min f}, & \min f_n \neq \max f_n, \\ 1, & \min f_n = \max f_n, \end{cases} \tag{5}$$

$$F_n = \begin{cases} \dfrac{\max f_n - f_n}{\max f_n - \min f}, & \min f_n \neq \max f_n, \\ 1, & \min f_n = \max f_n. \end{cases} \tag{6}$$

Formulas (5) and (6) are associated with the normalization of positive QoS indices (such as availability and reliability) and negative QoS attributes (such as time and cost), respectively. In this paper, for the convenience of expression, the aggregated QoS values are generated randomly between 0 and 1.

Collaboration is a typical feature of service composition,while as a typical model using to describe role-based collaborative system, E-CARGO [36–41], which is proposed by Professor Zhu in 2006, is applied to the service composition in cloud manufacturing for the clearly description of the collaboration of service composition in this paper. In E-CARGO, a role-based system is described as nine-tuples. $\Sigma := (C, O, A, M, R, E, G, S0, H)$, where C is a set of classes, O is a set of objects, A is a set of agents who are representatives of human users, M is a set of messages, R is a set of roles, E is a set of environments, G is a set of groups, $s0$ is the initial state of a collaborative system, and H is a set of users.

5.2. The Proposed Service Composition Model. To use the E-CARGO model for service composition in cloud manufacture, we map the related concepts involved in service composition to the corresponding tuples, and the specific process is shown in Figure 3.

Here, we introduce some necessary parameters to simplify the description of the model.

N is a nonnegative integer

$m(=|A|)$ is the number of agents, which is mapped to the number of services in one service candidate set

$n(=|R|)$ is the number of roles, which is mapped to the number of all tasks whose category is the same as the specified sub-task in the current total task

$q(=|Q|)$ is the number of all sub-tasks of the current original task after decomposition

Next, we give the following definitions combined with the above parameters.

Table 2: Competency of each service in RSS_k for each task under category TC_k.

	T_K^1	T_K^2	T_K^3	...	T_K^i	...
RSS_K^1	0.43	0.73	0.74	0.59	0.69	0.73
RSS_K^2	0.54	0.84	0.78	0.63	0.72	0.94
RSS_K^3	0.65	0.95	0.83	0.75	0.89	0.85
...				...		
RSS_K^j	0.95	0.35	0.87	0.83	0.92	0.65
...				...		
RSS_K^m	0.78	0.48	0.28	0.89	0.02	0.48

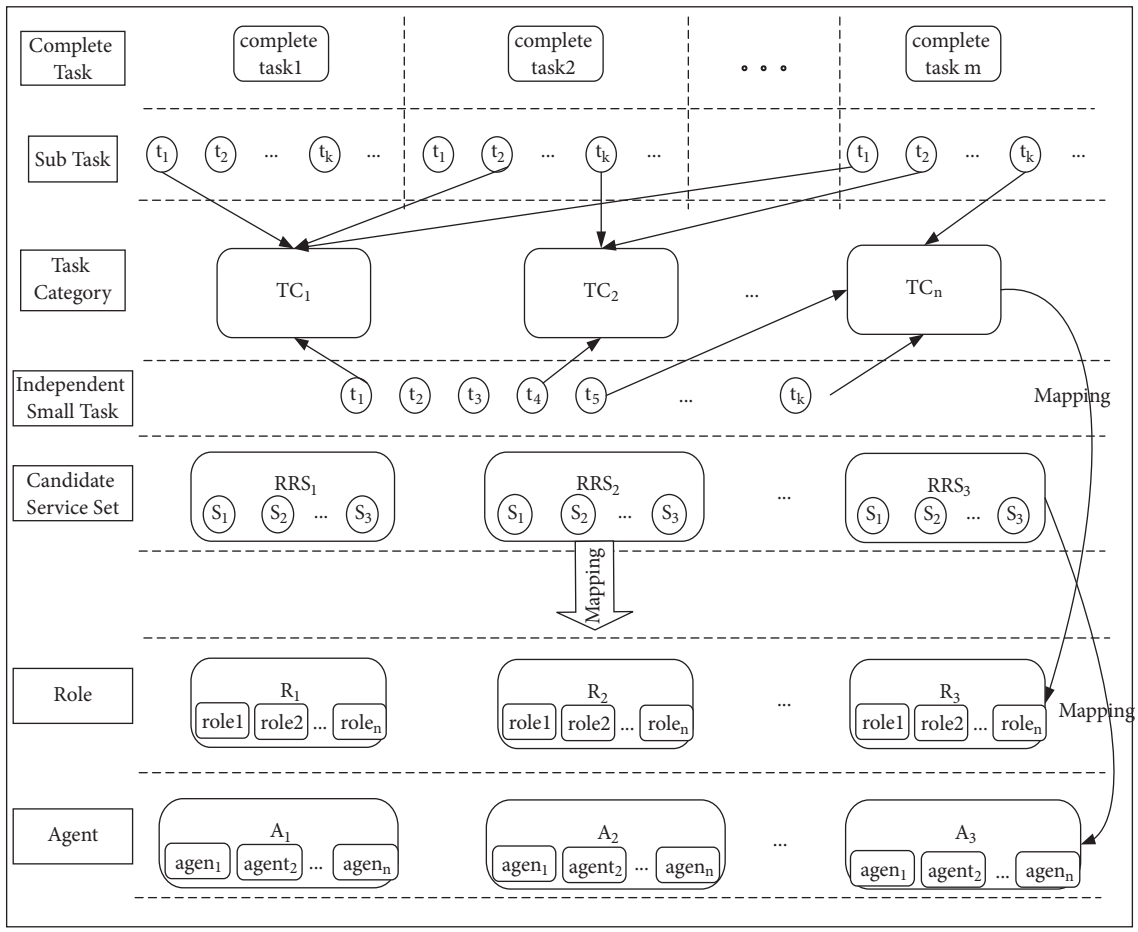


Figure 3: Mapping relationship between service composition and E-CARGO model.

Definition 1: in group G, $<i, j>$ is used to indicate that role j is assigned to agent i. It means that the manufacturing task j is assigned to service i in the cloud manufacturing environment.

Definition 2: in group G, $L(j) \in \in n$ $(1 \le j \le n)$ expresses that the role j needs to be assigned at least L $[j]$ agents. As for the cloud manufacturing environment, $L(c, j) \in n$ $(1 \le c \le q, 1 \le j \le n)$ expresses that subtask j of category c needs at least $L(c, j)$ services to complete cooperatively.

Definition 3: in group G, $L^a(i) \in m$ $(1 \le i \le m)$ means that agent i can only be assigned to $L^a(i)$ roles at most. As for the cloud manufacturing environment, $L^a(c, i) \in m$ $(1 \le c \le q, 1 \le i \le m)$ means that service i in the c-th candidate service set can only serve $L^a(i)$ tasks at the same time.

Definition 4: in group G, the matrix $Q[i, j] \in [0, 1]$ $(1 \le i \le m, 1 \le j \le n)$ represents the competence degree of agent I for role j, where 0 is the lowest competence degree, and 1 is the highest. In the cloud manufacturing environment, we introduce the variable c representing the category to expand the matrix Q. The expanded matrix Q $[c, i, j] \in [0, 1]$ (where $1 \le c \le q$, $1 \le i \le = m$, $1 \le j \le n$) describes the ability of service i in the c-th candidate service set to complete the j-th subtask of

category c. The values of competence degree are usually calculated according to the corresponding QoS model, which generally assigns different weights to different QoS criteria (e.g., time, cost, etc.).

Definition 5: in group G, the assignment matrix $T[i, j] = \{0, 1\}$ (where $1 \le i \le m$, $1 \le j \le n$) indicates whether role j is assigned to agent i. $T[i, j] = 1$ means role j is assigned to agent i, and $T[i, j] = 0$ means not assigned. In the cloud manufacturing environment, $T[c, i, j] \in [0, 1]$ (where $1 \le c \le q$, $1 \le i \le m$, $1 \le j \le n$) is used to indicate whether the j-th task of category c is assigned to the i-th candidate service set.

Definition 6: in group G, the assignment efficiency σ represents the total competency degree of all agents assigned roles, and it can be calculated by

$$\sigma = \sum_{1}^{m} \sum_{1}^{n} T(i, j) * Q(i, j). \tag{7}$$

In the cloud manufacturing environment, σ can be calculated by

$$\sigma = \sum_{c=1}^{q} \sum_{i=1}^{m} \sum_{j=1}^{n} T(c, i, j) * Q(c, i, j). \tag{8}$$

Definition 7: in group G, role j is workable when there are enough agents that can compete for it, and there is

$$\sum_{i=1}^{m} T[i, j] = L(j), \quad (1 \le j \le n). \tag{9}$$

In the cloud manufacturing environment, a manufacturing task j can be effectively assigned when all of its subtasks are effectively assigned, and formula (10) should be satisfied:

$$\sum_{i=1}^{m} T[c, i, j] = L(j), \quad (1 \le c \le q, 1 \le j \le n). \tag{10}$$

Definition 8: in group G, agent i is workable when the number of roles assigned to agent i does not exceed its workload, and there is

$$\sum_{i=1}^{n} T[i, j] = L^a(i) \quad (1 \le i \le m). \tag{11}$$

In the cloud manufacturing environment, the assignment result for service i is effective if it satisfies the constraint condition in

$$\sum_{c=1}^{q} \sum_{j=1}^{n} T[c, i, j] = L^a(i) \quad (1 \le c \le q, 1 \le j \le n). \tag{12}$$

Definition 9: in group G, the assignment matrix T is workable if each role and each agent is workable. If T is workable, then group G is workable. In the cloud manufacturing environment, if all subtasks decomposed from an original task can be assigned to enough services that meet the workload requirements, we can say there is effective service composition.

Finally, the process of role collaboration-based service composition is to find the optimal assignment scheme T, where $A(|A| = m)$, $R(|R| = n)$, $Q(|Q| = q)$, L and L^a are given. Namely, we need to solve the maximum value of the target object σ shown as formula (13) under the specified constraints.

$$\max \sum_{c=1}^{q} \sum_{i=1}^{m} \sum_{j=1}^{n} T(c, i, j) * Q(c, i, j). \tag{13}$$

subject to

$$\begin{aligned} &\sum_{i=1}^{m} T[c, i, j] = L(c, j), \quad (1 \le c \le q, 1 \le j \le n), \\ &\sum_{c=1}^{q} \sum_{j=1}^{n} T[c, i, j] = L^a(i), \quad (1 \le c \le q, 1 \le j \le n), \\ &T[c, i, j] \in \{0, 1\}, \quad (1 \le c \le q, 1 \le i \le m, 1 \le j \le n). \end{aligned} \tag{14}$$

5.3. Cplex-Based Solving Method. For obtaining higher execution efficiency, this paper bypasses the compilation process of the IBM ILOG CPLEX development environment and uses the method of directly referencing the ILOG development package in Java project to solve the above model. The specific steps are as follows:

(1) Find the mapping relationship: It is necessary to map the relevant elements involved in the service combination model to the four basic elements (objective function, function variable, variable coefficient, constraint condition) of the linear programming problem in ILOG, where the objective function is σ, the variables of the objective function correspond to the assignment matrix T, the variable coefficients correspond to the quality of service (QoS), and the constraint conditions are related to L and La.

(2) Add objective function: When using ILOG to solve linear programming problems, we need to transform the matrices Q and T into one-dimensional vectors and form the final objective function. Then the optimization target is added in the Java code by calling the following method of ILOG:

IloIntVar[] X = cplex.intVarArray($q * m * n$, 0, 1);
cplex.addMaximize(cplex.scalProd(X, V));
where, $X[c * m * n + i * n + j] = T[c, i, j]$, $V[c * m * n + i * n + j] = Q[c, i, j]$ ($1 \le c \le q$, $1 \le i < m$, $1 \le j \le n$).

(3) Add constraints:

First, declare the expression object of the constraints:

6. Experimental Analysis and Verification

Considering the types of open-source toolkits in different development languages and the characteristics of task decomposition algorithm and service composition model in this paper, Python language, and its mainstream scientific computing library are used to implement the task decomposition algorithm, and Java language and IBM ILOG CPLEX library are used to implement the service composition algorithm. The specific hardware and software experimental environment are shown in Table 3.

6.1. Case Analysis. The design and production process of military electric vehicles is extremely complex, and it is difficult to rely on a single service provider to complete the manufacturing task. Therefore, after receiving the task, the platform needs to decompose the task into executable and appropriate fine-grained subtasks and assign them to different service providers to complete the task cooperatively. After an in-depth understanding of the design and production process of military electric vehicles, we combine them in detail to form a more specific process as shown in Table 4, and a logical relationship between the processes is shown in Figure 4, where Serial Number 0 represents the completed vehicle [39].

The implementation of the task decomposition method proposed in this paper relies on the existing services. Therefore, the first step is to collect sufficient service data sets. We crawl some related service sets from the network and expand them through reproduce, mirroring, local adjustment and other methods according to the characteristics of the crawled network data in a reasonable range, where the service times of each service are randomly generated in a specified range according to the complexity of the service, and meantime, the dependency degree of the specified input type is set to a fixed value initially, for simplifying the difficulty of the solution, we only consider two input types: logistics and communication.

For verifying the effectiveness of the task preliminary decompose algorithm, we take the design and production of the drive motor system as input and obtain the result shown in Figure 5, where the values of nodes is the corresponding serial number in Table 5. Next, we reorganize the result using the reorganization algorithm proposed in this paper, but the result has not changed.

Then, the design and production of the electric drive system is taken as input for the proposed algorithm. The result of the preliminary decomposition shown in Figure 6(a) and that of the reorganization shown in 6(b) are obtained, respectively.

Without any expert system in the industry, this proposed decomposition method can decompose complex manufacturing tasks into subtasks with enough candidate service sets according to the characteristics of the existing service in the platform, which provides the necessary premise for the realization of collaborative manufacturing.

After the subtasks and the corresponding candidate service sets are obtained by task decomposition, an optimal assignment scheme is needed to lay the foundation for the subsequent task scheduling. To simplify the description process, this paper takes the task described as the design and production of the electric drive system as an example to verify the effectiveness of the proposed service composition model.

Known from Figure 6, the task named design and production of electric drive system can be decomposed into two first-level subtasks, design and production of drive motor system and design and production of power system, which can come from task decomposition or direct release by other users. Assuming that the number of them is n1 and n2, respectively, and that of the corresponding candidate service sets are m1 and m2, respectively. At the same time, the related data involved in this experiment are initialized as the following: n1, n2, m1, and m2 are random numbers between 3 and 5; the number of services can be accepted by a single task L and the workload of single service L^a are all random numbers between 1 and 3; the service quality of each service in the candidate set (in real application scenarios, it is calculated based on its historical evaluation data) is a random number between 0 and 1. Finally, the initial situation is shown in Table 6.

According to the initialization data of the above model shown in Table 6, we use the Cplex-based method to solve the model and get the assignment scheme shown in Table 5, which takes 204.02 ms and the sum of the QoS in this scheme is 7.13. It should be noted that the above randomly generated data may not be able to find the assignment scheme, which is a common scenario in practical applications, and the specific processing measures are not within the scope of this paper.

6.2. Performance Analysis. On the one hand, the current research on task decomposition is relatively few, and there is no unified evaluation standard. On the other hand, considering that the main advantage of the proposed decomposition strategy in this paper is to reduce the dependence on industry expert system and solve the problem of disconnection between task decomposition and service composition process, which has been reflected in the specific implementation process. Therefore, we take no in-depth comparative analysis of that.

The proposed service composition model can smoothly transition from the task decomposition procedure to make full use of the existing service state in the cloud pool, and it can be easily introduced with a variety of variable factors to expand and optimize itself. For example, all the same, or similar independent tasks want to be assigned the best services. Therefore, the demander's acceptance matrix M can be introduced to expand the model (13) to obtain the optimization model (14); all interdependent services need a solid foundation for cooperation with each other. Therefore, the provider's acceptance matrix N can be introduced to expand the model (14) to (15); in addition, with the rapid development of social networks, the influence of peer effect in the evaluation system of the platform becomes increasingly important, which can directly or indirectly affect the choice of users themselves or other users. Therefore, the

Table 3: Hardware and software experimental environment.

System environment	
CPU	Intel(R) Core(TM) i7-6500U CPU @2.50 GHz 2.50 GHz
Memory	8 G (7.87 G可用)
Operating system	Windows 7 ultimate
Java development environment	
Development tool	Eclipse version: Luna Release (4.4.0)
JDK	jdk8u241
Third-party library	Cplex
Python development environment	
Development tool	PyCharm Community Edition 2021.1.1+ Anaconda3
Python	Python 3.7.3
Third-party library	Jieba 0.42.1+ gensim 3.8.3+ numpy 1.18.3

Table 4: Specific design activity units chart.

Serial number	Design and production name
1	Electric drive system
2	Power system
3	Power battery
4	Battery management system
5	Car charger
6	Auxiliary power source
7	Drive motor system
8	Electronic controller
9	Power converter
10	Drive motor
11	Mechanical transmission
12	Clutch
13	Transmission
14	Transmission shaft and other universal transmissions
15	Axle (main reducer, differential axle housing, etc.)
16	Wheels
17	Vehicle controller
18	Motor controller
19	Current sensor
20	Voltage sensor
21	Temperature sensor
22	Body
23	Body-in-white
24	Body safety guard
25	Auxiliary system
26	Automotive instrumentation, lighting and accessories
27	Power steering
28	Steering mechanism
29	Steering gear
30	Steering transmission mechanism
31	Front and rear suspension
32	Braking system

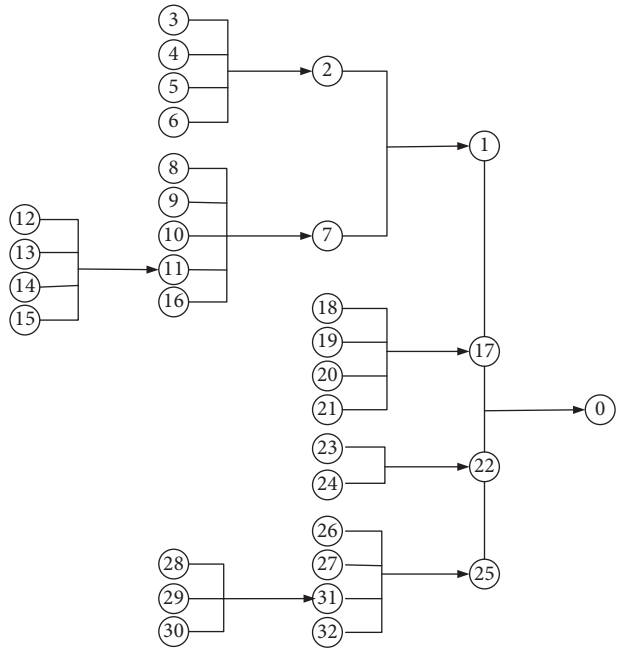


Figure 4: Logical relationship between the processes.

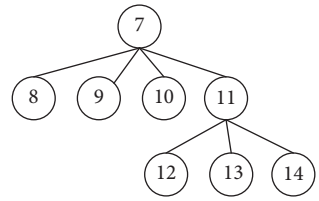


Figure 5: Decomposition result of "design and production of drive motor system."

Table 5: Current tasks waiting to be assigned, corresponding candidate service sets, and their constraints.

Design and production of power system				
Services(m1)	Tasks(n1)			
	Task1/L [1] = 1	Task2/L [2] = 1	Task3/L [3] = 2	Task4/L [4] = 2
Service1/L^a [1] = 2	0.88	0.70	0.04	0.52
Service2/L^a [2] = 2	0.94	0.47	0.87	0.74
Service3/L^a [3] = 2	0.72	0.46	0.94	0.24
Design and production of drive motor system				
Services(m2)	Tasks(n2)			
	Task1/L [1] = 1	Task2/L [2] = 1	Task3/L [3] = 1	Task4/L [4] = 2
Service1/L^a [1] = 2	0.27	0.50	0.71	0.09
Service2/L^a [2] = 1	0.22	0.06	0.89	0.01
Service3/L^a [3] = 1	0.58	0.16	0.00	0.49
Service4/L^a [4] = 2	0.65	0.52	0.53	0.01

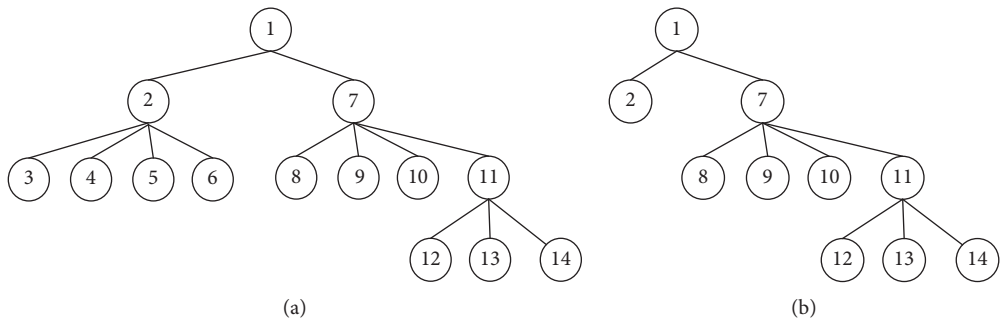


Figure 6: Decomposition result of "design and production of the electric drive system." (a) Result of decomposition. (b) Result of reorganization.

Table 6: Signment results based on the proposed model.

Design and production of power system				
Services(m1)	Tasks(n1)			
	Task1/L [1] = 1	Task2/L [2] = 1	Task3/L [3] = 2	Task4/L [4] = 2
Service1/L^a [1] = 2	0	1	0	1
Service2/L^a [2] = 2	0	0	1	1
Service3/L^a [3] = 2	1	0	1	0
Design and production of drive motor system				
Services(m2)	Tasks(n2)			
	Task1/L [1] = 1	Task2/L [2] = 1	Task3/L [3] = 1	Task4/L [4] = 2
Service1/L^a [1] = 2	0	0	0	1
Service2/L^a [2] = 1	0	0	1	0
Service3/L^a [3] = 1	0	0	0	1
Service4/L^a [4] = 2	1	1	0	0

user's social relationship S can be introduced to optimize the model and obtain the optimization model (17). However, considering the limited space of this paper, the modeling of M, N, and S is not described in detail. In the next experimental verification, we only take the proposed service composition model as the basic model and verify its effectiveness and adaptability by comparing it with the improved Hungarian algorithm [37, 42] (we call it KMB in the following description):

$$\max \sum_{l=1}^{q} \sum_{i=1}^{m} \sum_{j=1}^{n} T(l,i,j) * Q(l,i,j) * M(l,i,j), \tag{15}$$

$$\max \sum_{l=1}^{q} \sum_{i=1}^{m} \sum_{j=1}^{n} T(l,i,j) * Q(l,i,j) * M(l,i,j) * N(l,i,j), \tag{16}$$

$$\max \sum_{l=1}^{q} \sum_{i=1}^{m} \sum_{j=1}^{n} T(l,i,j) * Q(l,i,j) * M(l,i,j) * N(l,i,j)\delta * S(l,i,j). \tag{17}$$

Firstly, the validity is verified, and the experimental parameters are set as follows: q, n, m, $L[j]$, and L^a[i] are all random integers, where, for improving the success rate of assignment, the values of m are usually set to an integer multiple of values of n (the number of services is generally required to be more than the number of tasks), $L^a[i]$ is set between 1 and 3 and $L[j]$ is set between 1 and $2m/n$. The experimental results are shown in Table 7. It can be seen that the proposed method and the KBM algorithm can get comparable results at most times; in addition, KMB algorithm is more efficient when the number of tasks n is less than 20. However, when the number of tasks n is greater than 20, the KMB algorithm's efficiency is significantly lower than that of the proposed method. Considering that in the real business environment, the number of subtasks and their corresponding service candidate sets is usually much more than 20, so the proposed composition model and the corresponding solution algorithm can better meet the engineering requirements.

Table 7: Comparison of validity of the proposed composition model, the corresponding solution algorithm and km.

Numbers	q	m	n	Cplex execution time (s)			km execution time (s)			Sum of QoS (times) (Cplex ? km)		
				Avg	Max	Min	Avg	Max	Min	>	=	<
100	[1–5]	$2 * n$	[5–10]	0.0233	0.188	0.005	0.0010	0.004	0.0	51	0	49
100	[1–5]	$5 * n$	[5–10]	0.0357	0.206	0.007	0.0026	0.009	0.0	54	0	46
100	[5–10]	$2 * n$	[20–50]	0.4283	0.774	0.116	0.7094	1.52	0.078	57	0	43
100	[5–10]	$5 * n$	[20–50]	0.8664	1.838	0.236	4.0076	8.924	0.414	49	0	51
100	[1–5]	$2 * n$	[100–150]	1.5421	3.644	0.287	10.7309	26.428	1.257	57	0	43
100	[5–10]	$5 * n$	[100–150]	4.0185	9.091	0.745	66.5779	161.536	8.317	55	0	44

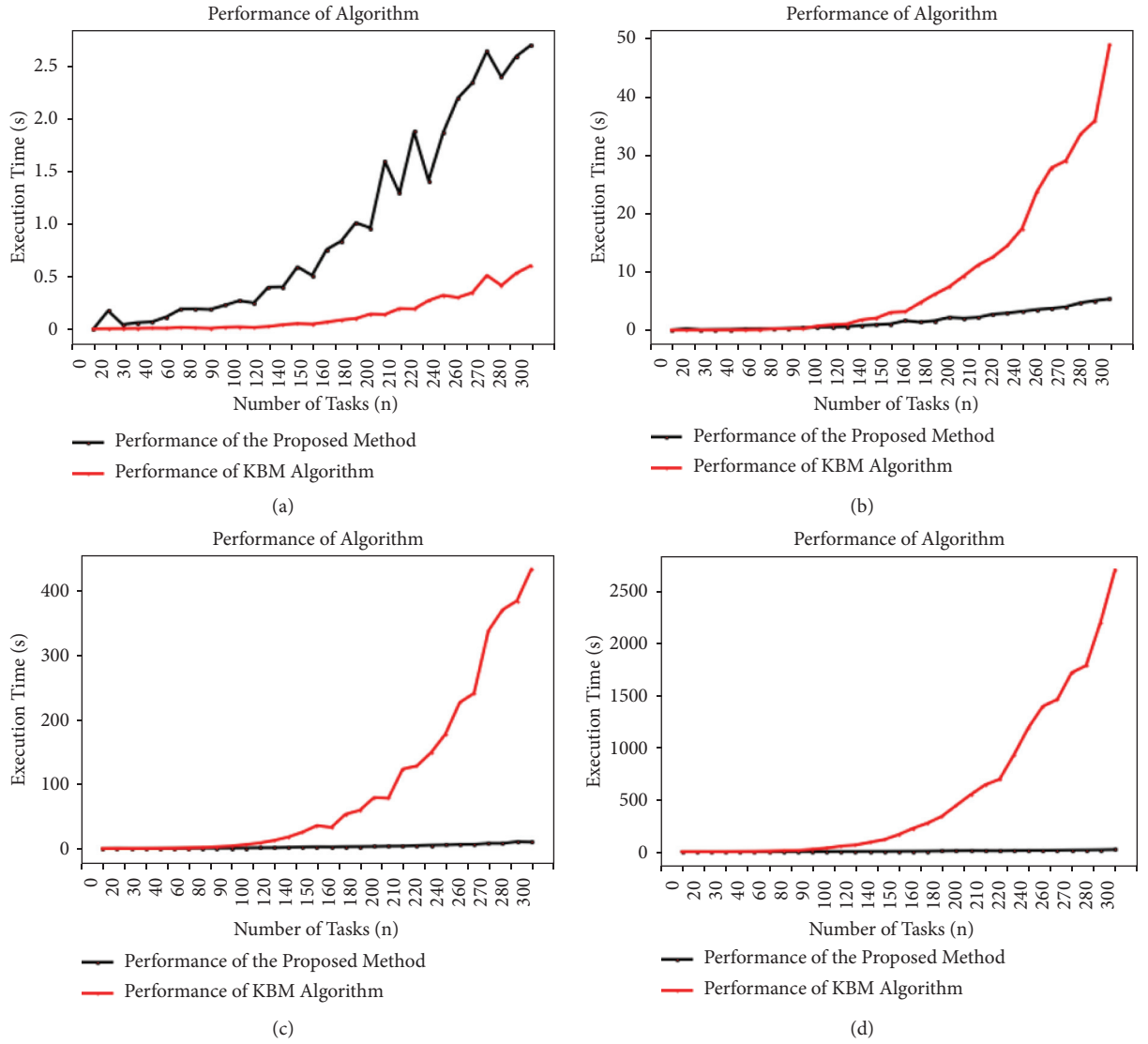


Figure 7: Performance comparison of our method and KBM algorithm. (a) $m = n$, $L^a[i] \in [1, 2]$. (b) $m = 2 * n$, $L^a[i] \in [1, 3]$. (c) $m = 3 * n$, $L^a[i] \in [1, 4]$. (d) $m = 4 * n$, $L^a[i] \in [1, 5]$.

Secondly, to verify the overall performance of the proposed composition model and the corresponding solution algorithm, we use different scales of random integers for experimental analysis, where q is a fixed value of 2, n increases from 10 to 300 with the pace of 10 each time, m is set to $1 * n$, $2 * n$, $3 * n$, $4 * n$ in turn, $L[j]$ and $L^a[i]$ are random numbers from 1 to m/n. At the same time, for avoiding the randomness of the experimental results, each data pair(q, m, n) is randomly tested for 100 times, and the average value is collected as the final experimental data, and the final performance trend chart as shown in Figure 7 is formed.

It can be seen that the KMB algorithm's execution efficiency decreases sharply with the increase of $L^a[i]$, and its complexity is $O\sum L^a[i]^3$. Compared with the KMB algorithm, the execution efficiency of the proposed method is more stable, and when $L^a[i]$ is greater than or equal to 2, it is far better than the KMB algorithm; at the same time, we notice that both KMB algorithm and our solution method have some fluctuations in the execution process. When $m = n$, more exceptions exist in the proposed composition model and the corresponding solution method than that of the KBM algorithm, which is caused by the unreachable assignment scheme under the existing conditions (the number of services and tasks are the same, but some tasks need multiple services). And when $m > n$, the exception of our method does not exist. However, the KMB algorithm still has some anomalies in some cases. In most cases, the above two solutions can meet the practical needs, but compared with the KMB algorithm, our composition model and solution method is obviously better than KMB algorithm in both efficiency and stability.

7. Conclusions and Future Work

Optimal manufacturing resource allocation is a core problem in the cloud manufacturing mode. In the specific implementation process, we should solve the problems of resource virtualization, task decomposition, task service matching, service composition, and scheduling in turn. This paper mainly proposes the corresponding solutions for task decomposition and service composition model. As for task decomposition, we refine the description model of task and service and propose the preliminary decomposition scheme based on task service matching and the reorganization strategy based on the characteristics of the service candidate set. As a result, the problem of discontinuity between task decomposition and service composition is solved. For solving the problem of task and service assignment, we design the service composition model using E-CARGO and solve the model using Cplex, which not only greatly reduces the problem of falling into local optimum of heuristic algorithms but also provides the necessary foundation for the introduction of more variable factors(such as cooperation, conflict and other constraints). Finally, the practicability of decomposition strategy and service composition model is proved by the experimental analysis.

The future work may follow several aspects:

(1) Task scheduling of manufacturing resource optimized allocation. Compared with the management of independent resources of enterprises, the management of shared resources is more dynamic (e.g., devices can join or withdraw from sharing services at any time). Hence, the difficulty in task scheduling will be greatly increased.

(2) Personalized recommendation applied in manufacturing resource optimized allocation. To improve the user-friendliness and convenience of online platforms, the personalized service recommendation for different customer requirements is an effective means. However, since manufacturing services usually appear in the form of composite services, existing Web service-based personalized recommendation technologies are difficult to be applied effectively.

(3) Real-time response of large-scale cases in a dynamic environment should be investigated to prove the superiority of Blockchain technology-based method over the centralized optimization methods. And more comparisons with other methods (e.g., PSO, game theory) should be made [43].

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This work was supported by the Key Research and Development Program of Anhui Province of China (T04a05020094, 904a05020091, 04a05020091, 04a05020092, and 04a05020093), the Scientific Research Project of Chaohu University (XLZ-202107, XLZ-201807), the Teaching and Research Project of Chaohu University (ch18jxyj18), and the Applied Curriculum Project of Chaohu University (ch18yygc13).

References

[1] B. H. Li, L. Zhang, S. L. Wang, F. Tao, and X. D. Chai, ""Cloud manufacturing: a new service-oriented networked manufacturing model," [In Chinese]," *Computer Integrated Manufacturing Systems*, vol. 16, no. 1, pp. 1–7, 2010.

[2] B. H. Li, L. Zhang, L. Ren, X. D. Chai, F. Tao, and Y. L. Luo, ""Further discussion on cloud manufacturing," [in Chinese]," *Computer Integrated Manufacturing Systems*, vol. 17, no. 3, pp. 449–457, 2011.

[3] Y. Liu and X. Xu, "Industry 4.0 and cloud manufacturing: a comparative analysis," *Journal of Manufacturing Science and Engineering*, vol. 139, no. 3, 2017.

[4] Y. Zhang, K. Wang, Q. He et al., "Covering-based Web service quality prediction via neighborhood-aware matrix factorization," *IEEE Transactions on Services Computing*, vol. 14, no. 5, pp. 1333–1344, 2021.

[5] Y. Zhang, G. Cui, S. Deng, F. Chen, Y. Wang, and Q. He, "Efficient query of quality correlation for service composition," *IEEE Transactions on Services Computing*, vol. 14, no. 3, pp. 695–709, 2021.

[6] Y. Zhang, J. Pan, L. Qi, and Q. He, "Privacy-preserving quality prediction for edge-based IoT services," *Future Generation Computer Systems*, vol. 114, no. 2021, pp. 336–348, 2021.

[7] Y. Zhang, C. Yin, Q. Wu, Q. He, and H. Zhu, "Location-Aware deep collaborative filtering for service recommendation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3796–3807, 2021.

[8] F. Tao, L. Zhang, H. Guo, Y. L. Luo, and L. Ren, ""Typical characteristics of cloud manufacturing and several key issues of cloud service composition," [In Chinese]," *Computer Integrated Manufacturing Systems*, vol. 17, no. 3, pp. 477–486, 2011.

[9] Y. Liu, L. Wang, X. V. Wang, X. Xu, and P. Xu, "Cloud manufacturing: key issues and future perspectives," *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 9, pp. 858–874, 2019.

[10] D. Wu, M. J. Greer, D. W. Rosen, and D. Schaefer, "Cloud manufacturing: strategic vision and state-of-the-art," *Journal of Manufacturing Systems*, vol. 32, no. 4, pp. 564–579, 2013.

[11] S. Son, J. Kim, J. Lee, and J. Ahn, "Improving supply chain management process using design structure matrix based cross-functional analysis," *Systems Engineering*, vol. 22, no. 4, pp. 313–329, 2019.

[12] H. Son, Y. Kwon, S. C. Park, and S. Lee, "Using a design structure matrix to support technology roadmapping for product-service systems," *Technology Analysis & Strategic Management*, vol. 30, no. 3, pp. 337–350, 2018.

[13] G. E. da Cunha Barbosa and G. F. M. de Souza, "A risk-based framework with Design Structure Matrix to select alternatives of product modernisation," *Journal of Engineering Design*, vol. 28, no. 1, pp. 23–46, 2017.

[14] S. Kherbachi, Q. Yang, and L. Yang, "Multi-domain integration of team-product-function and organization clustering in product development project," *Systems Engineering*, vol. 38, no. 6, pp. 1557–1565, 2017.

[15] D. T. Liu and D. J. Zhou, ""Task decomposition and recombination design structure matrix based on interval number," [In Chinese]," *Machine Design and Reasearch*, vol. 25, no. 6, pp. 7–9, 2009.

[16] S. Shriyam, B. C. Shah, and S. K. Gupta, "Decomposition of collaborative surveillance tasks for execution in marine environments by a team of unmanned Surface vehicles," *Journal of Mechanisms and Robotics*, vol. 10, no. 2, 2018.

[17] Z. Zhang, Y. Zhang, J. Lu, F. Gao, and G. Xiao, "A novel complex manufacturing business process decomposition approach in cloud manufacturing," *Computers & Industrial Engineering*, vol. 144, Article ID 106442, 2020.

[18] Y. Hu, Z. Zhang, J. Wang, Z. Wang, and H. Liu, "Task decomposition based on cloud manufacturing platform," *Symmetry*, vol. 13, no. 8, p. 1311, 2021.

[19] L. Guo, Y. Xu, W. He, and Y. Cheng, "Optimization of complex part-machining services based on feature decomposition in cloud manufacturing," *International Journal of Computer Integrated Manufacturing*, vol. 33, no. 12, pp. 1227–1244, 2020.

[20] M. Z. Liu, Q. Wang, and L. Lin, "Cloud manufacturing task decomposition method based on HTN," *China Mechanical Engineering*, vol. 28, no. 8, pp. 924–930, 2017.

[21] S. P. Yi, M. Z. Tan, Z. L. Guo, W. P. Han, and J. Zhou, "Manufacturing task decomposition optimization in cloud manufacturing service platform," *[In Chinese] Computer Integrated Manufacturing Systems*, vol. 21, no. 8, pp. 2201–2212, 2015.

[22] Y. Liu, L. Wang, X. V. Wang, X. Xu, and L. Zhang, "Scheduling in cloud manufacturing: state-of-the-art and research Challenges," *International Journal of Production Research*, vol. 57, no. 15-16, pp. 4854–4879, 2019.

[23] M. Yuan, Z. Zhou, X. Cai, C. Sun, and W. Gu, "Service composition model and method in cloud manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 61, Article ID 101840, 2020.

[24] E. Aghamohammadzadeh and O. Fatahi Valilai, "A novel cloud manufacturing service composition platform enabled by Blockchain technology," *International Journal of Production Research*, vol. 58, no. 17, pp. 5280–5298, 2020.

[25] J. Zhou and X. Yao, "Multi-objective hybrid artificial bee colony algorithm enhanced with Lévy flight and self-adaption for cloud manufacturing service composition," *Applied Intelligence*, vol. 47, no. 3, pp. 721–742, 2017.

[26] L. Zhu, P. Li, G. Shen, and Z. Liu, "A novel service composition algorithm for cloud-based manufacturing environment," *IEEE Access*, vol. 8, pp. 39148–39164, 2020.

[27] J. Zhou, X. Yao, Y. Lin, F. T. S. Chan, and Y. Li, "An adaptive multi-population differential artificial bee colony algorithm for many-objective service composition in cloud manufacturing," *Information Sciences*, vol. 456, no. 5, pp. 50–82, 2018.

[28] C. Li, J. Guan, T. Liu, N. Ma, and J. Zhang, "An autonomy-oriented method for service composition and optimal selection in cloud manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 96, no. 5-8, pp. 2583–2604, 2018.

[29] B. Xu, J. Qi, X. Hu, K.-S. Leung, Y. Sun, and Y. Xue, "Self-adaptive bat algorithm for large scale cloud manufacturing service composition," *Peer-to-Peer Networking and Applications*, vol. 11, no. 5, pp. 1115–1128, 2018.

[30] Y. Que, W. Zhong, H. Chen, X. Chen, and X. Ji, "Improved adaptive immune genetic algorithm for optimal QoS-aware service composition selection in cloud manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 96, no. 9-12, pp. 4455–4465, 2018.

[31] F. Li, L. Zhang, Y. Liu, and Y. Laili, "QoS-Aware service composition in cloud manufacturing: a gale-shapley algorithm-based approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 7, pp. 2386–2397, 2020.

[32] B. Liu and Z. Zhang, "QoS-aware service composition for cloud manufacturing based on the optimal construction of synergistic elementary service groups," *International Journal of Advanced Manufacturing Technology*, vol. 88, no. 9-12, pp. 2757–2771, 2017.

[33] H. Jin, X. Yao, and Y. Chen, "Correlation-aware QoS modeling and manufacturing cloud service composition," *Journal of Intelligent Manufacturing*, vol. 28, no. 8, pp. 1947–1960, 2017.

[34] Y. Laili, S. Lin, and D. Tang, "Multi-phase integrated scheduling of hybrid tasks in cloud manufacturing environment," *Robotics and Computer-Integrated Manufacturing*, vol. 61, pp. 101850.1–101850.18, 2020.

[35] H. Huang, Z. Wang, Y. J. Ji, and Y. Yan, ""Analysis on distributed networked cloud manufacturing mode," [In Chinese]," *Modern Manufacturing Engineering*, vol. 11, pp. 42–48+65, 2017.

[36] H. B. MengChu Zhou and M. C. Zhou, "Role-based collaboration and its kernel mechanisms," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 36, no. 4, pp. 578–589, 2006.

[37] H. Zhu, M. Zhou, and R. Alkins, "Group role assignment via a kuhn-munkres algorithm-based solution," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 3, pp. 739–750, 2012.

[38] Y. Sheng, H. Zhu, X. Zhou, and W. Hu, "Effective approaches to adaptive collaboration via dynamic role assignment," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 1, pp. 76–92, 2016.

[39] H. Zhu, "Avoiding conflicts by group role assignment," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 4, pp. 535–547, 2016.

[40] H. Zhu, Y. Sheng, X. Zhou, and Y. Zhu, "Group role assignment with cooperation and conflict factors," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 6, pp. 851–863, 2018.

[41] D. Liu, Q. Jiang, H. Zhu, and B. Huang, "Distributing UAVs as wireless repeaters in disaster relief via group role assignment," *International Journal of Cooperative Information Systems*, vol. 29, no. 01n02, Article ID 2040002, 2020.

[42] H. Zhu, D. Liu, S. Zhang, Y. Zhu, L. Teng, and S. Teng, "Solving the Many to Many assignment problem by improving the Kuhn-Munkres algorithm with backtracking," *Theoretical Computer Science*, vol. 618, pp. 30–41, 2016.

[43] X. Xiangqian, Y. Kewei, D. Yajie, Z. Zhou, and T. Yuejin, "High-end equipment development task decomposition and scheme selection method," *Journal of Systems Engineering and Electronics*, vol. 32, no. 1, pp. 118–135, 2021.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 4827956, 16 pages
https://doi.org/10.1155/2022/4827956

Research Article

Game-Based Channel Selection for UAV Services in Mobile Edge Computing

Y. Chen,[1] **H. Xing**,[1] **S. Chen**,[1] **N. Zhang**,[2] **X. Chen**,[1] **and J. Huang**[3]

[1]*School of Computer Science, Beijing Information Science and Technology University, Beijing 100101, China*
[2]*University of Windsor, Windsor, Canada*
[3]*Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum, Beijing 102249, China*

Correspondence should be addressed to Y. Chen; chenying@bistu.edu.cn

Received 22 October 2021; Revised 8 December 2021; Accepted 21 December 2021; Published 3 February 2022

Academic Editor: Xiaolong Xu

Computation offloading is a hot research topic in mobile edge computing (MEC). Computation offloading among multiedge nodes in heterogeneous networks can help reduce offloading cost. In addition, the unmanned aerial vehicles (UAVs) play a key role in MEC, where UAVs in the air communicate with ground base stations to improve the network performance. However, limited channel resources can lead to the increase of transmission delay and the decline of communication quality. Effective channel selection mechanisms can help address those issues by improving transmission rate and ensuring communication quality. In this paper, we study channel selection during communication between multiple UAVs and base stations in an MEC system with heterogeneous networks. To maximize the transmission rate of each UAV user, we formulate a channel selection problem and model it as a noncooperative game. Then, we prove the existence of Nash equilibrium (NE). In addition, we design a multiple UAV-enabled transmission channel selection (UTCS) algorithm to obtain the equilibrium strategy profile of all the UAV users. Experimental results validate that UTCS algorithm can converge after a finite number of iterations and it outperforms random transmission algorithm (RTA) and sequential transmission algorithm (STA).

1. Introduction

With the development of mobile computing, the number of mobile users has soared. However, the network resources are limited. To reduce the communication and computing delay, MEC is proposed as a promising paradigm [1, 2]. The computing and storage capacities are provided at edge nodes, providing services for terminal tasks and effectively reducing data processing delay and energy consumption of terminal devices. 5G-enabled MEC [3] is composed of heterogeneous base stations, where small cell base stations (SBSs) [4] reuse the channel resources with macro cell base station (MBS). Users of MBS and SBSs use the same set of communication resources for data transmission, which can effectively improve the utilization of communication resources.

With the development of unmanned aerial vehicles (UAVs) technologies, smaller and cheaper UAVs are available. As such, UAVs are no longer only used in specific fields (such as military domain) but are also introduced into civil networks [5]. Due to the high altitude flight characteristics, UAVs have been used as a key component of MEC [6], providing aerial network services [7, 8] in the event of a disaster [9]. In the face of some major natural disasters (such as earthquakes, floods, and typhoons), deployed ground base stations are likely to be damaged and local network disruptions can occur. UAVs play a vitally important role in damaged network rescue. Due to the flexible operating altitude and wide coverage, UAVs can communicate with user terminals in network damaged areas to obtain computationally intensive tasks and then communicate with base stations in network normal areas for processing. In addition, with the help of UAVs' unique LoS link, UAVs can ensure high-quality communication in extreme environments [10, 11] (e.g., rain, snow, and broken trees).

As the number of computing tasks brought by UAVs increases, the demand for communication resources keeps growing. However, channel resources for data transmission are still heavily limited. When the number of users is too large, the channel is multiplexed by deploying micro base stations. In this case, channel multiplexing would bring the issue of interference, resulting in a decrease in the transmission rate. In order to improve the data transmission rate, it is necessary to design a reasonable channel selection strategy to reduce the interference of data transmission. An effective channel selection strategy can improve the utilization of channel resources and ensure the high transmission rate of each computing task.

During network rescue, UAVs can communicate with user terminals and base stations. Many existing studies merely focused on the communication between UAVs and users, and the communication between multiple UAVs and ground base stations had not been explored widely. Thus, we study the communication between UAVs and ground base stations. UAVs transmit computing data to the aerial and ground integrated wireless networks and process the task with the help of the computing resources [12–14] of the ground base stations. As the number of tasks increases, the transmission rate [15] of each task decreases. That is mainly because of the interference caused by channel multiplexing. Considering the characteristics of tasks, each computing task [16] selects the appropriate channel for data transmission, which can effectively reduce channel multiplexing. We aim to propose a multi-UAV transmission channel selection method to achieve distributed and high-quality data transmission.

In this paper, aiming at the offloading problem in edge computing, we study channel selection strategies for communication between multi-UAV users and base stations in MEC with heterogeneous networks. The following are our main contributions:

(i) We consider the MEC with heterogeneous networks consisting of an MBS and multiple SBSs, in which each SBS is assigned an MEC server to provide computing resources for relevant users. In addition, the UAV users of SBSs use channel resources of the MBS for data transmission. We formulate a noncooperative game to model and analyze the multi-UAV channel selection problem in such a heterogeneous network. In particular, our model considers the interference generated by channel multiplexing to ensure high-quality data transmission for each UAV user.

(ii) The channel selection strategies of users are coupled to each other, so it is difficult to optimize each user's data transmission rate simultaneously. To show the existence of Nash equilibrium (NE) in the formulated noncooperative game, we prove its equivalence to an exact potential game which has at least one NE.

(iii) To reach the NE of the formulated noncooperative game, we propose a multiple UAV-enabled transmission channel selection (UTCS) algorithm. The proposed algorithm operates in a completely distributed manner; that is, each user does not know the selection strategies of other users and independently adjusts the channel selection strategy according to their prior experience.

(iv) We perform extensive numerical simulations to validate UTCS algorithm to compute the NE solution (i.e., the equilibrium channel selection and the equilibrium data transmission rate of each user). Experimental results show that UTCS algorithm can converge quickly through a finite number of iterations. Compared with the random transmission algorithm (RTA) and sequential transmission algorithm (STA), UTCS algorithm can obtain NE channel selection strategy and each user can achieve a higher data transmission rate.

The remainder of this paper is organized as follows. In Section 2, we briefly summarize the related researches. We model the communication between UAV users and base stations based on heterogeneous network scenarios and establish a noncooperative game model in Section 3. In Section 4, we analyze the existence of NE and design UTCS algorithm to obtain the equilibrium strategy. In addition, we illustrate the performance of UTCS algorithm through parameter analysis, convergence analysis, and comparison experiments in Section 5 and conclude our research work in Section 6.

2. Related Work

Some researches have been done on UAV network [17–22]. Zhang et al. [19] considered two communication modes, UAV to base station and UAV to UAV, and divided the problem into three subproblems to optimize channel allocation and UAV speed, respectively. Gu et al. [20] considered the network scenario of multi-UAV collaborative work in the context of environment awareness and studied resource allocation and task scheduling of multi-UAV collaborative work based on reinforcement learning. Berate et al. [21] regarded the UAV as an aerial base station and proposed a user cache framework involving multiple UAVs. Then, QoE was used as an indicator to study the user cache for multiple UAV base station deployment and the optimal caching strategy was obtained. Zhao et al. [22] considered a network scenario in which UAV and base station (BS) cooperate to serve ground users and investigated the problem of transmission rate optimization through joint optimization of UAV trajectory and NOMA precoding. The above studies introduced UAVs into different network scenarios to optimize the data transmission process from different perspectives of network communication. We introduce UAVs into 5G heterogeneous networks, which can greatly improve the power efficiency and spectrum efficiency.

Channel selection is a hot issue in mobile network communication. Gour et al. [23] selected underlay D2D

network and proposed D2D channel allocation and power allocation schemes with and without quality of service constraints and transformed the problem into a nonconvex mixed-integer nonlinear programming problem. Shattal et al. [24] focused on a new vehicle-mounted ad hoc network (VANETs) architecture in which nodes continuously and autonomously selected one of three channel selection strategies and applied evolutionary games to solve the channel selection problem. Ko et al. [25] studied the joint optimization problem of LTE channel selection and frame scheduling to maximize LTE throughput and proposed heuristic algorithms to solve the problem. However, these works do not consider the cochannel transmission interference factor, which leads to the result that channel selection is not accurate enough. Thus, considering the transmission interference in the same channel, we study the channel selection decision problem for multi-UAV users.

Game theory is a good tool to solve the problem of multiplayer competitive decision-making. Cui et al. [26] considered the dynamics and uncertainties in the environment for modeling, constructed the long-term resource allocation problem as a stochastic game to maximize the expected return, and applied the reinforcement learning theory to design algorithm to solve the problem. In order to solve the optimization problem of relay selection in UAV network, Liu et al. [27] proposed a matching game classification method based on the competitive relationship between players and constructed a basic preliminary model of UAV relay model. However, these works do not highlight the competitive relationship between users. In this paper, we consider the communication resource competition in the process of data transmission by multi-UAV users, and a noncooperative game model is established to describe the channel resource competition in multi-UAV communication.

This paper considers a heterogeneous network of multiple types of multiple base stations communicating with multiple UAV users. Since different base station users select the same channel for transmission and cause interference, we apply the noncooperative game method to construct the communication model between users and base stations. In order to maximize the transmission rate of each user, a scheme of user equilibrium channel selection strategy is presented. Then, we prove the existence of equilibrium strategy. Combining the strategy selection probability, we design UTCS algorithm to ensure that multi-UAV users can obtain their own equilibrium strategy in the 5G heterogeneous networks. Finally, the convergence and effectiveness of UTCS algorithm are verified by experiments.

3. The UAV-Enabled Heterogeneous Network Model for MEC and the Game Problem Formulation

3.1. UAV-Enabled Heterogeneous Network for MEC. We consider a UAV-enabled heterogeneous network as shown in Figure 1, where multiple SBSs are distributed in the macro cell. Each SBS owns an MEC server to provide computing resources for corresponding users. In this network, the UAV

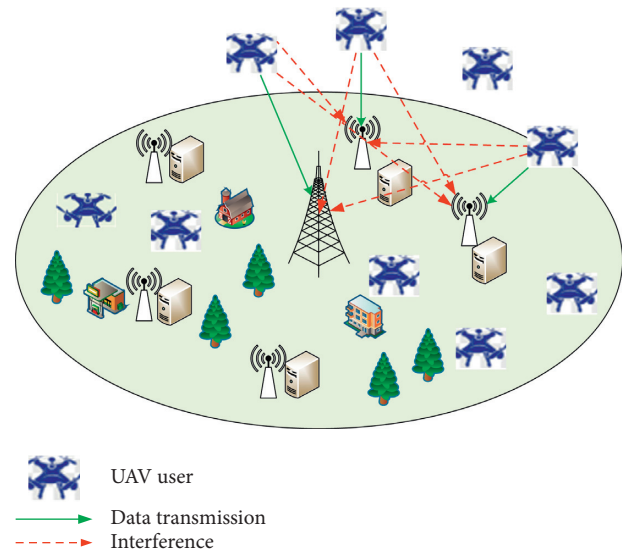


Figure 1: The UAV-enabled heterogeneous networks in MEC.

user transmits the data to the base station for task computation. The set of UAV users is defined as $\mathcal{N} = \{1, 2, \ldots, N\}$, which can be further divided into macro cell base station users (MUs) and small cell base station users (SUs). Specifically, the MU is the user outside of the service area of the SBS and is served by MBS, and the set of MUs is defined as $\mathcal{N}_0 = \{1, 2, \ldots, N_0\}$. In addition, the set of SUs is represented by $\mathcal{N}_M = \{1, 2, \ldots, N_M\}$. The set of all the base stations is represented as $\mathcal{M} = \mathcal{M}_0 \cup \mathcal{M}_m = \{0, 1, \ldots, M\}$, which contains an MBS and M SBSs. The MBS provides services to MUs and SBSs provide services to SUs in their own service areas. $\mathcal{M}_0 = \{0\}$ represents the MBS, which provides computing services to all the MUs. The set of SBSs is $\mathcal{M}_m = \{1, 2, \ldots, M\}$, which is deployed in the service area of the MBS. The SUs' set of the SBS m service area is $\mathcal{N}_m = \{1, 2, \ldots, N_m\}$, and $\sum_{m \in \mathcal{M}_m} |\mathcal{N}_m| = |\mathcal{N}_M|$. The main symbols are given in Table 1.

We define the channel set of the MBS as $\mathcal{S} = \{1, 2, \ldots, S\}$. SBSs are deployed in the service area macro cell and the channels of the MBS are reused for data transmission. In addition, with dense deployment of SBS, it is considered that the number of users in each base station is no more than the number of channels, and users within the same base station do not need to reuse channels.

Due to the multiplexing of channels, there is interference when the user transmits the data, including the interference from MUs to SUs and the interference from SUs to SUs. Interference affects the data transmission rate, and users obtain efficient data transmission strategies by the analysis of the interference. d_i denotes the channel selection strategy of user i, and $d_i \in \mathcal{S}$. The channel selection strategy profile of all the users is represented as $D = \{d_1, d_2, \ldots, d_N\}$. In addition, the users in the same base station do not reuse the channels; that is, $d_i \neq d_j, \forall i, j \in \mathcal{N}_0$, and $d_i \neq d_j, \forall i, j \in \mathcal{N}_m, m \in \mathcal{M}_m$.

When data is transmitted from UAV user to base station, the UAV is in a hover state and $U_i = [x_i, y_i, H_i]^T$ represents the location information of user i. The data transmission rate is affected by the channel gain which is related to the distance

Table 1: Notations.

Symbol	Description
$\mathcal{N}$	The set of UAV users
$\mathcal{N}_0$	The set of MUs
$\mathcal{N}_m$	The set of SUs
$\mathcal{M}$	The set of base stations
$\mathcal{M}_0$	The set of MBSs
$\mathcal{M}_m$	The set of SBSs
$\mathcal{S}$	The set of channels
d_i	The channel selection strategy of user i
D	The channel selection strategy profile of all the users
U_i	The location information of user i
Q_0	The location information of the MBS
Q_m	The location information of SBS m
β_0	The channel power of MBS
α_{i0}	The decay coefficient of MBS
β_m	The channel power of SBS m
α_{im}	The fading coefficient of SBS m
p_i	The transmission power of user i
σ^2	The background noise
d_{-i}	The channel selection strategy profile other than user i
B	The bandwidth of the base station

between the user and the base station. The position coordinate of MBS is defined as $Q_0 = [x_0, y_0, H_0]^T$, and $Q_m = [x_m, y_m, H_m]^T$ represents the position of SBS m.

Different from ordinary users on the ground, the UAV and the base station may transmit data through the LoS link [28]. The path loss for LoS link transmission α_{in}^{LoS} and the path loss for NLoS link transmission α_{in}^{NLoS} between UAV user i and base station n are expressed, respectively, as follows:

$$\begin{aligned}\alpha_{in}^{LoS} &= 20\log\left(\frac{4\pi f_c\left\|Q_n - U_i\right\|}{c}\right) + \varepsilon^{LoS},\\ \alpha_{in}^{\mathrm{NLoS}} &= 20\log\left(15\frac{4\pi f_c\left\|Q_n - U_i\right\|}{c}\right) + \varepsilon^{\mathrm{NLoS}},\end{aligned} \tag{1}$$

where $\varepsilon^{\mathrm{LoS}}$ represents the average value of excessive path loss in LoS link and ε^{NLoS} is the average value of excessive path loss in NLoS link. f_c is the carrier frequency, c is the speed of light, and $\|Q_n - U_i\|$ is the distance between UAV user i and base station n. It is worth noting that base station n is either an MSB or an SBS. The probability of data transmission between UAV user i and base station n through the LoS link is given as

$$\rho_{in} = \frac{1}{1 + be^{-\varsigma\left(\arcsin\left(H_n - H_i/\left\|Q_n - U_i\right\|\right) - b\right)}}, \tag{2}$$

where b and ς are environmental parameters determined by the deployment environment of UAVs. $\arcsin(H_n - H_i/\|Q_n - U_i\|)$ is the horizontal angle between UAV user i and base station n.

The average path loss between UAV user i and base station n is denoted as

$$\overline{\alpha}_{\mathrm{in}} = \rho_{\mathrm{in}}\alpha_{\mathrm{in}}^{\mathrm{LoS}} + \left(1 - \rho_{\mathrm{in}}\right)\alpha_{\mathrm{in}}^{\mathrm{NLoS}}, \tag{3}$$

and the channel gain h_i^0 of MU i can be calculated as follows:

$$h_i^0 = \frac{\beta_0}{\left\|Q_0 - U_i\right\|^{\overline{\alpha}_{i0}}}, \tag{4}$$

where $\|Q_0 - U_i\|$ represents the distance between MU i and MBS, and β_0 is the channel power of MBS.

The channel gain h_i^m between SU i and SBS m is defined as

$$h_i^m = \frac{\beta_m}{\left\|Q_m - U_i\right\|^{\overline{\alpha}_{im}}}, \tag{5}$$

where $\|Q_m - U_i\|$ is the transmission distance between SU i and SBS m. β_m represents the channel power of SBS m.

Based on the channel gain and multiplexing, users can attain the channel selection strategy. We define a_i^s to indicate whether user i selects channel s for data transmission. Specifically, when user i selects channel s for data transmission, $a_i^s = 1$; otherwise, $a_i^s = 0$. The SINR of MU i is given as follows:

$$r_i^0 = \frac{a_i^s h_i^o p_i}{\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2}, \tag{6}$$

where p_i and p_j are the transmission powers of MU i and SU j, respectively. σ^2 represents the noise power. $\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s$ represents the interference to MU i by SU j that selects the same channel for transmission as MU i.

For SUs, we calculate the interference between the MBS and the SBS and the interference between SBSs. The SINR of SU i is as follows:

$$r_i^m = \frac{a_i^s h_i^m p_i}{\sum_{j\in\mathcal{N}_o} h_j^0 p_j a_j^s + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_{m/j\neq i}} h_j^m p_j a_j^s + \sigma^2}, \tag{7}$$

where $\sum_{j\in\mathcal{N}_o} h_j^0 p_j a_j^s$ denotes the interference of MU j to SU i, and $\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s$ represents the interference of other SBS users to SU i.

Each user considers the impact of strategies for other users on themselves when obtaining channel selection strategy. We define the channel selection strategy profile of all the users other than user i as follows:

$$d_{-i} = \{d_1, d_2, \ldots, d_{i-1}, d_{i+1}, \ldots, d_N\}. \tag{8}$$

When user i selects channel s for data transmission, the transmission rate of user i can be given by

$$R_i\left(d_i, d_{-i}\right) = \begin{cases} \dfrac{1}{N_0} B\log_2\left(1 + r_i^0\right), & i \in \mathcal{N}_o,\\ \dfrac{1}{N_m} B\log_2\left(1 + r_i^m\right), & i \in \mathcal{N}_m, \end{cases} \tag{9}$$

where B is the bandwidth of the base station. The bandwidth of each base station is the same, and the base station allocates the bandwidth resource evenly to each user in its coverage. In addition, not only is the data transmission rate affected by

the user's own channel selection strategy, but also it depends on other users' strategies.

3.2. The Game Problem Formulation. In this model, we regard UAV users as rational users. Each user is a selfish decision-maker, and they only care about their own benefits. Specifically, UAV users select the channel with the highest data transmission rate, considering their locations and channel reuse. Therefore, the objective function of user i is given as follows:

$$\max_{d_i \in \mathcal{S}} R_i(d_i, d_{-i}), \quad \forall i \in \mathcal{N}. \tag{10}$$

Because of channel multiplexing, the channel selection strategies of users affect each other. If there are too many users selecting the same channel, interference will be more severe among these users, and the transmission rate of these users will decrease. Therefore, each user prefers to select the channel selected by a small number of users in pursuit of the maximum transmission rate. However, in real multiuser data transmission, users do not know the channel selection strategies of other users when making the transmission decision. Therefore, users are independent decision-makers and can only follow their prior experience to make decisions.

In the process of pursuing their own benefits maximization, users form a competitive relationship with each other. We can describe this process of user channel selection as a noncooperative game model. Then, the game can be formulated as follows:

$$\Gamma = \left(\mathcal{N}, \{\mathcal{S}\}_{i \in \mathcal{N}}, \{R_i\}_{i \in \mathcal{N}}\right), \tag{11}$$

where $\mathcal{N}$ is the set of game players, which include both MUs and SUs. $\mathcal{S}$ is the feasible strategy profile of the participant and also is the channel set. Note that the set of feasible strategies is the same for each user. In addition, R_i is the benefit of participant i's strategy, which is the transmission rate. The transmission rate of each user varies as the channel selection strategy changes.

In Γ, each user adjusts its channel selection strategy to gain more benefits. After a certain number of iterations, all the users reach a state in which they can no longer improve their benefits by changing their strategies. Therefore, all the users would keep the strategy unchanged. In this state, there is an equilibrium between all the users.

4. Analysis of Nash Equilibrium and Decision Algorithm Design

4.1. Proof of the Existence of NE. We consider an equilibrium state. Since users can no longer improve their own benefits by adjusting their strategies in this equilibrium state, the channel selection strategies of all users are no longer changed. The equilibrium state makes a tradeoff among all the users; that is, considering the satisfaction of all the users, the benefits of each user reach the relative best state. In other words, the equilibrium state satisfies the goal of maximizing the transmission rate of each user in Γ, which is called a Nash equilibrium (NE), and we give the definition of the NE of Γ in the following.

Definition 1. For a noncooperative game Γ, if there is a channel selection strategy profile $D^* = \{d_1^*, d_2^*, \ldots, d_N^*\}$, and no user is willing to unilaterally change its channel selection strategy to improve its transmission rate in this environment, then

$$R_i(d_i^*, d_{-i}^*) \geq R_i(d_i, d_{-i}^*), \quad \forall d_i \in \mathcal{S},\ i \in \mathcal{N}, \tag{12}$$

and D^* is the NE strategy profile of Γ.

When all the users select the equilibrium strategy for data transmission, they have no intention to further change the channel selection strategy because each user reaches the maximum benefit in the current environment. Therefore, NE is a stable state and a solution that satisfies the goal of Γ. In the following, we discuss the existence of NE.

The exact potential game has the finite improvement property (FIP), which indicates that the exact potential game can reach NE through a finite number of iterations. Therefore, we further prove the existence of NE in Γ by proving that Γ is an exact potential game. Then, we give the definition of the exact potential game.

Definition 2. For Γ, the channel selection strategy d_i of user i, the channel selection strategy profile d_{-i} of all the users other than user i, and the objective function $R_i(d_i, d_{-i})$ of Γ are given. If and only if there is a potential function $\Phi(d_i, d_{-i})$, the relationship between $\Phi(d_i, d_{-i})$ and $R_i(d_i, d_{-i})$ is as follows:

$$R_i(d_i, d_{-i}) - R_i(d_i', d_{-i}) = \Phi(d_i, d_{-i}) - \Phi(d_i', d_{-i}), \tag{13}$$

where Γ is an exact potential game, and there is at least a pure strategy NE.

In Γ, we consider two types of users and they are independent of each other. Therefore, we prove the existence of NE for two kinds of users, respectively. In the following, we first analyze the case for MU.

Lemma 1. *For Γ, if user i is the MU, there is a function $\Phi^0(D)$ as follows:*

$$\Phi^0(D) = \frac{1}{N_0} B \log_2\left(\sum_{m \in \mathcal{M}_m} \sum_{j \in \mathcal{N}_m} h_j^m p_j a_j^s + h_i^0 p_i a_i^s + \sigma^2\right), \tag{14}$$

so that the relationship between $\Phi^0(D)$ and $R_i(d_i, d_{-i})$ satisfies the following condition:

$$R_i^0(d_i, d_{-i}) - R_i^0(d_i', d_{-i}) = \Phi^0(d_i, d_{-i}) - \Phi_0(d_i', d_{-i}). \tag{15}$$

Proof. When user i is an MU, the utility function of MU i, that is, the transmission rate, can be given as

$$
\begin{aligned}
R_i^0(d_i, d_{-i}) &= \frac{1}{N_0} B \log_2\left(1 + \frac{a_i^s h_i^o p_i}{\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2}\right) \\
&= \frac{1}{N_0} B \log_2\left(\frac{\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2 + a_i^s h_i^o p_i}{\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2}\right) \\
&= \frac{1}{N_0} B \log_2\left(\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2 + a_i^s h_i^o p_i\right) \\
&\quad - \frac{1}{N_0} B \log_2\left(\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2\right).
\end{aligned} \tag{16}
$$

We analyze the change of the utility function when the channel selection strategy of user i changes. Specifically, when the transmission channel of user i changes from channel s to channel $s\prime$, that is, from a_i^s to $a_i^{s'}$, the utility function changes as follows:

$$
\begin{aligned}
&R_i^0(d_i, d_{-i}) - R_i^0(d_i', d_{-i}) \\
&= \frac{1}{N_0} B \log_2\left(\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2 + a_i^s h_i^o p_i\right) \\
&\quad - \frac{1}{N_0} B \log_2\left(\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2\right) \\
&\quad - \frac{1}{N_0} B \log_2\left(\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2 + a_i^{s'} h_i^o p_i\right) \\
&\quad + \frac{1}{N_0} B \log_2\left(\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2\right) \\
&= \frac{1}{N_0} B \log_2\left(\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2 + a_i^s h_i^o p_i\right) \\
&\quad - \frac{1}{N_0} B \log_2\left(\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2 + a_i^{s'} h_i^o p_i\right).
\end{aligned} \tag{17}
$$

Then, we show that when the channel selection strategy of user i changes, the potential function changes as follows:

$$
\begin{aligned}
&\Phi^0(d_i, d_{-i}) - \Phi^0(d_i', d_{-i}) \\
&= \frac{1}{N_0} B \log_2\left(\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2 + a_i^s h_i^o p_i\right) \\
&\quad - \frac{1}{N_0} B \log_2\left(\sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2 + a_i^{s'} h_i^o p_i\right) \\
&= R_i^0(d_i, d_{-i}) - R_i^0(d_i', d_{-i}).
\end{aligned} \tag{18}
$$

Through the above proof, we can obtain Lemma 1. □

Lemma 2. *In Γ, when user i is a small cell base station user, there is a potential function* $\Phi^m(D)$*:*

$$
\Phi^m(D) = \frac{1}{N_m} B \log_2\left(\sum_{j\in\mathcal{N}_0} h_j^0 p_j a_j^s + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2\right). \tag{19}
$$

When the channel selection strategy of user i changes, the following condition is satisfied between $\Phi^m(D)$ *and* $R_i^m(d_i, d_{-i})$*:*

$$
R_i^m(d_i, d_{-i}) - R_i^m(d_i', d_{-i}) = \Phi_m(d_i, d_{-i}) - \Phi_m(d_i', d_{-i}). \tag{20}
$$

Proof. If user i is an SU, according to (9), the utility function of SU i is equivalent to

$$
\begin{aligned}
R_i^m(d_i, d_{-i}) &= \frac{1}{N_m} B\log_2\left(1 + \frac{a_i^s h_i^m p_i}{\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_{m/j\neq i}} h_j^m p_j a_j^s + \sigma^2}\right) \\
&= \frac{1}{N_m} B\log_2\left(\frac{\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2}{\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_{m/j\neq i}} h_j^m p_j a_j^s + \sigma^2}\right) \\
&= \frac{1}{N_m} B\log_2\left(\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2\right) \\
&\quad - \frac{1}{N_m} B\log_2\left(\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_{m/j\neq i}} h_j^m p_j a_j^s + \sigma^2\right),
\end{aligned}
\tag{21}
$$

where $\xi^0 = \sum_{j\in\mathcal{N}_0} h_j^0 p_j a_j^s$ represents the interference of MUs to user i.

When the channel selection strategy of user i changes, the utility function changes as follows:

$$
\begin{aligned}
R_i^m(d_i, d_{-i}) - R_i^m(d_i', d_{-i}) &= \frac{1}{N_m} B\log_2\left(\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2\right) \\
&\quad - \frac{1}{N_m} B\log_2\left(\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_{m/j\neq i}} h_j^m p_j a_j^s + \sigma^2\right) \\
&\quad - \frac{1}{N_m} B\log_2\left(\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_{m/j\neq i}} h_j^m p_j a_j^s + h_i^m p_i a_i^{s'} + \sigma^2\right) \\
&\quad + \frac{1}{N_m} B\log_2\left(\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_{m/j\neq i}} h_j^m p_j a_j^s + \sigma^2\right) \\
&= \frac{1}{N_m} B\log_2\left(\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2\right) - \\
&\quad \frac{1}{N_m} B\log_2\left(\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_{m/j\neq i}} h_j^m p_j a_j^s + h_i^m p_i a_i^{s'} + \sigma^2\right).
\end{aligned}
\tag{22}
$$

The relationship between the potential function and the utility function when the channel selection strategy changes is as follows:

$$
\begin{aligned}
&\Phi^m(d_i, d_{-i}) - \Phi^m(d_i', d_{-i}) \\
&= \frac{1}{N_m} B\log_2\left(\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_m} h_j^m p_j a_j^s + \sigma^2\right) \\
&\quad - \frac{1}{N_m} B\log_2\left(\xi^0 + \sum_{m\in\mathcal{M}_m}\sum_{j\in\mathcal{N}_{m/j\neq i}} h_j^m p_j a_j^s + h_i^m p_i a_i^{s'} + \sigma^2\right) \\
&= R_i^m(d_i, d_{-i}) - R_i^m(d_i', d_{-i}).
\end{aligned}
\tag{23}
$$

We can obtain that when user i is an SU, Γ is an exact potential game. Thus, Lemma 2 is proved.

As shown above, we respectively prove the relationship between the potential function and the utility function of two types of users. Now we give the potential function of the whole Γ. □

Theorem 1. *Γ is an exact potential game, and the potential function is as follows:*

$$\Phi(D) = \begin{cases} \Phi^0(D), & i \in \mathcal{N}_o, \\ \Phi^m(D), & i \in \mathcal{N}_m. \end{cases} \tag{24}$$

Proof. In Γ, we consider two types of users, MUs and SUs. In Lemmas 1 and 2, we theoretically prove whether the game is an exact potential game in two user cases. In addition, the two types of users are independent of each other. Specifically, a user will not be both an MU and an MU. Thus, the feasible strategy profiles of users do not affect each other. From Lemmas 1 and 2, we derive the following:

$$R_i(d_i, d_{-i}) - R_i(d_i', d_{-i}) = \Phi(d_i, d_{-i}) - \Phi(d_i', d_{-i}), \forall i \in \mathcal{N}. \tag{25}$$

According to the above proof, we conclude that, for each user $i \in \mathcal{N}$, when the channel selection strategy changes, the utility function and potential function satisfy the equality relation given in (25). Thus, Γ is an exact potential game, and Theorem 1 is proved. □

4.2. Multiple UAV-Enabled Transmission Channel Decision Algorithm (UTCD). Because the exact potential game has FIP property, users in Γ can reach NE after a finite number of iterations. We design the algorithm to reach the NE. Specifically, we consider how the user selects the transmission channel in each iteration. Designing a proper channel selection mechanism can make the game reach equilibrium quickly.

How to design effective methods to solve network game problems has been widely studied, for example, adopting the idea of best response. In the idea of best response, each user, in accordance with the principle of maximizing their own interests, continuously adjusts their strategies in each iteration to achieve equilibrium. Each user obtains the channel selection strategy by traversing its feasible strategy profile based on the current game environment and calculates the benefits respectively for comparison. It is a strategy iterative process. In each iteration, the user calculates the benefits of all the feasible strategies to obtain the most profitable strategy.

In the above solution process, the default strategies of different users can be the same. However, in the heterogeneous network, users of the same base station service cannot reuse the same channel for data transmission. Therefore, we introduce the channel selection probability and combine the channel selection probability with the best response [29] idea to design our channel selection algorithm. Considering the particularity of the scenario, we propose a channel strategy selection algorithm applicable to the UAV-enabled game model in Algorithm 1, that is, the multiple UAV-enabled transmission channel decision (UTCD) algorithm. Specifically, each user's feasible strategy profile is accompanied by a selection probability vector. For a user, the probability of selecting the feasible channel to transmit is different, and the change of probability is related to the corresponding benefits of the strategy. If the corresponding benefits of the selected strategy are higher, the selection probability of the strategy will be higher. However, the total probability for each user's feasible strategy profile is constant. The specific update method of the strategy selection probability is shown in Algorithm 2.

The probability of all users' strategy selection constitutes the strategy selection matrix, which is used to update the game environment. The selection of probability affects the benefits. When we update the game environment, we select the strategy with the maximal probability and at the same time ensure that users of the same base station service select different strategies. The main steps of UTCS algorithm are given as follows:

Strategy selection probability initialization. Each user's strategy profile corresponds to a selection probability vector. Therefore, the number of elements in the vector is equal to the number of elements in the strategy profile. The probability is initially evenly allocated to each strategy. The probability vector of initial strategy selection for user i is given as follows:

$$P_i(t = 0) = \left(\frac{1}{S}, \frac{1}{S}, \ldots, \frac{1}{S}\right). \tag{26}$$

Game environment initialization. For users of a base station service, we assign the initial channel strategy to users in order of channel set $\mathcal{S}$. The initial channel selection for users of base station m is $E_m(t = 0) = \{1, 2, \ldots, N_m\}, E_m \subseteq \mathcal{S}$. The initial channel selection of all the users constitutes the initial game environment.

Benefit calculation. In each iteration, the user adjusts the strategy selection probability by calculating the benefits. The benefits are calculated by (9), and the current game environment is considered in the calculation process. In addition, in order to satisfy the fairness and simultaneity of the user's decision, the user, after adjusting the probability vector, keeps the current strategy unchanged until the game environment updates.

Selection probability update. The selection probability is updated according to the calculated benefits of the selected strategy. The probability after the update is related to the current benefits and the probability before the update, and the specific calculation method is as follows:

$$P_i(t + 1) = P_i(t) + \varepsilon r_i^t \left(e_{d_i}^t - P_i(t)\right), \tag{27}$$

where ε is the update step size applied to control the overall change rate of the probability and $e_{d_i}^t$ is the unit

(1) Initialize: $\mathcal{N}$, $\mathcal{S}$, U_i, Q_m, Q_0, β_0, β_m, α_0, α_m, p_i, p_j, σ^2, a_i^s, B;
(2) According to channel set $\mathcal{S}$, UAV users of the same base station service are allocated the initial transmission channel in order;
(3) The initial game environment is constituted as follows, $E(t=0) = (d_1, d_2, \ldots, d_N)$;
(4) The update strategy probability matrix of users is obtained from Algorithm 2 as follows, $\mathrm{E}(t) = \begin{pmatrix} p_1^1 & \cdots & p_1^S \\ \vdots & & \vdots \\ p_N^1 & \cdots & p_N^S \end{pmatrix}$;
(5) According to $\mathrm{E}(t)$, the strategy with the highest probability of each UAV user i is selected to form a new game environment.
(6) The selection method is, $d_i(t) = \arg\max P(d_i, d_{-i}), \forall d_i \in \mathcal{S}$;
(7) While satisfying the maximization principle, the selection of the channel selection strategy should ensure that there is no channel multiplexing between users of the same base station service;
(8) When all the users' selection strategies no longer change, the update is stopped;
Output: The equilibrium strategy profile, $D^* = \{d_1^*, d_2^*, \ldots, d_N^*\}$.

Algorithm 1: The multiple UAV-enabled transmission channel selection (UTCS) algorithm.

(1) The strategy selection probability of each MU i is set according to the fairness principle, $P_i(t=0) = ((1/S), (1/S), \ldots, (1/S))$;
(2) **for** each MU $i \in \mathcal{N}$ **do**
(3) The transmission rate are calculated according to 10;
(4) Each UAV user's interference calculation is based on the current game environment;
(5) The following is applied to update the stratgy selection probability of user i; $P_i(t+1) = P_i(t) + \varepsilon r_i^t (e_{d_i}^t - P_i(t))$;
(6) **end for**
(7) The probability of strategy selection for all UAV users constitutes the strategy selection matrix;
Output: The update strategy probability matrix of users is, $\mathrm{E}(t) = \begin{pmatrix} p_1^1 & \cdots & p_1^S \\ \vdots & & \vdots \\ p_N^1 & \cdots & p_N^S \end{pmatrix}$.

Algorithm 2: The transmission strategy update algorithm.

vector whose d_i-th element is 1. In addition, r_i^t is the utility, which is obtained by $r_i^t = \eta_i R_i^t$, and $\eta_i \le 1/\max R_i$. After the selection probability of all the users has been updated, the next iteration is entered.

Game environment update. The user follows the principle of selecting the strategy of maximum probability; that is,

$$d_i(t) = \arg\max P(d_i, d_{-i}), \quad \forall i \in \mathcal{N}. \tag{28}$$

However, in the base station m, in order to avoid channel multiplexing, the users' strategy selections are different. Specifically, N_m elements are selected with different rows and columns in the following probability matrix:

$$\mathrm{E}_m = \begin{pmatrix} p_1^1 & \cdots & p_1^S \\ \vdots & & \vdots \\ p_{N_m}^1 & \cdots & p_{N_m}^S \end{pmatrix}. \tag{29}$$

The probability of the strategy is generated according to the amount of the strategy benefits, and the greater the benefits are, the greater the probability increases. Therefore, when the maximum selection probability strategies of different users are the same, the user strategy with the highest probability value is selected.

Termination. Each user selects a transmission strategy for each iteration. When the selections of all the users are no longer changed, the iteration ends, and the strategy profile is the desired equilibrium strategy profile as follows:

$$D^* = \{d_1^*, d_2^*, \ldots, d_N^*\}. \tag{30}$$

As shown above, all the users select strategies in the same game environment, which can ensure the synchronization of all the user decisions, and more in line with the reality.

5. Performance Evaluation

In this section, we first analyze the influence of the values of two parameters on the model and the convergence of the algorithm. Then, the performance of UTCS algorithm is analyzed by comparison experiment.

We consider that there are 50 UAV users in a UAV-enabled heterogeneous network with one MBS and 5 SBSs deployed. The user's location information is randomly generated within the service scope of the base station to which it belongs. For ease of calculation, we assume that all the UAVs hover at a fixed height of 20 m. For the MBS, the altitude is 10 m, and the altitude of SBSs is 5 m. The power gain of the MBS is −50 db, and the power gain of the SBSs is generated randomly from [−30, −40] dB. In addition, the transmission power for each UAV user is randomly generated from [100, 500] mW. The bandwidth is 5 MHz, and the bandwidth is evenly distributed among users. The background noise is −100 dbm. The initial values for the main parameters are set in Table 2.

Table 2: Parameters used in the evaluation.

Parameters	Value
The number of UAV users	50
The number of MBSs	1
The number of SBSs	5
The altitude of UAV users	20 m
The altitude of the MBS	10 m
The altitude of SBSs	5 m
The power gain of the MBS	−50 db
The power gain of SBSs	[−30, −40] dB
The transmission power for each UAV user	[100,500] mW
The bandwidth	5 MHz
The background noise	−100 dbm

5.1. Parameter Analysis. α is the path loss exponent in data transmission between the UAV user and the base station. We set different path loss exponents (i.e., 2.5, 2.6, and 2.7) to analyze the effect of path loss on the transmission rate. Figure 2 shows the impact of path loss exponent on channel gain at different transmission distances. We can see that, with the increase of distance, the overall transmission rate tends to decline, and the greater the distance is, the slower the transmission rate decreases. When the transmission distance is the same, the smaller α is, the higher the channel gain will be. In addition, Figure 3 shows the impact of path loss exponent on the channel gain with base stations of different channel power gains. Obviously, when the channel power gain increases, the channel gain tends to increase. The smaller the loss coefficient is, the higher the channel gain is. According to Figures 2 and 3, the higher the path loss is, the slower the channel gain is. The path loss index is determined by the actual environment in which the data is transmitted. Specifically, when the transmission environment is complex, the transmission loss will be high. Conversely, in a simple environment, the transmission loss is low; and the channel gain affects the transmission rate, so that the UAV hover location would be selected in a simple transmission environment to increase the transmission rate.

Figure 4 shows the impact of the update step size on the change rate of the selection probability of a strategy. ε is the update step size of the selection probability for the strategy. As the transmission rate increases, the selection probability increases from 0.1. Moreover, the larger the update step size is, the faster the probability grows. In Figure 5, we show the effect of update step size on the convergence rate of transmission rate. As can be seen in the figure, with the increase of the number of iterations, the transmission rate of the UAV user gradually reaches a convergence state. The reason is that all users obtain the optimal transmission rate. Meanwhile, no user has the will to change strategy. The transmission rate's convergence is the fastest and the user's transmission rate is the largest when all situations have converged after 11 iterations, which is 0.6. Therefore, the selection of the appropriate update step size can affect the convergence rate and the user's equilibrium rate simultaneously.

5.2. Convergence Analysis. Figure 6 shows the convergence of user transmission strategies. Three users (i.e., user 9, user 33, and user 39) are randomly selected to observe the trend of their channel selection strategies as the number of iterations increased. Theoretically, we prove the existence of NE in the game in Section 3. Therefore, it can be seen from Figure 6 that the channel selection strategies of these users show a convergence trend after a finite number of iterations. Specifically, user 9 converges after 4 iterations and finally selects channel 5 for data transmission. In the 10th iteration, the channel selection strategy of user 33 no longer changes. After 8 iterations, user 39 selects channel 6 for data transmission. Due to the different transmission power, channel gain, and other factors, different users finally reach the convergence of the strategy after different iterations.

When the user's channel selection strategy no longer changes, the user's transmission rate may not reach the constant state due to the influence of other users' adjustment strategies. Therefore, the convergence state of user's transmission rate can show the convergence rate more accurately. Figure 7 shows the convergence of user transmission rates. We randomly select user 2, user 21, and user 45 from 50 UAV users and analyze the change of user transmission rate as the number of iterations increases. As shown in Figure 7, with the increase of the number of iterations, the user's transmission rate gradually converges and eventually remains unchanged. Specifically, after 8 iterations of the UAV user, the user's transmission rate no longer changes. Due to channel multiplexing, there is interference between users. Therefore, the strategy changes of other users can affect the transmission rate of the user, and the transmission rates of all the users converge at the same time.

5.3. Comparison Analysis. Compared with other algorithms, we analyze the performance of UTCS algorithm. In the random transmission algorithm (RTA), each UAV user randomly selects a channel for data transmission. In the sequential transmission algorithm (STA), each user allocates channel resources in the order of the channel set. In UTCS algorithm, users adjust channel selection strategies with the goal of maximizing their own transmission rate and obtain the final channel selections through multiple iterations. Figure 8 shows the change in the total transmission rate for all the users as the number of iterations increases. Obviously,

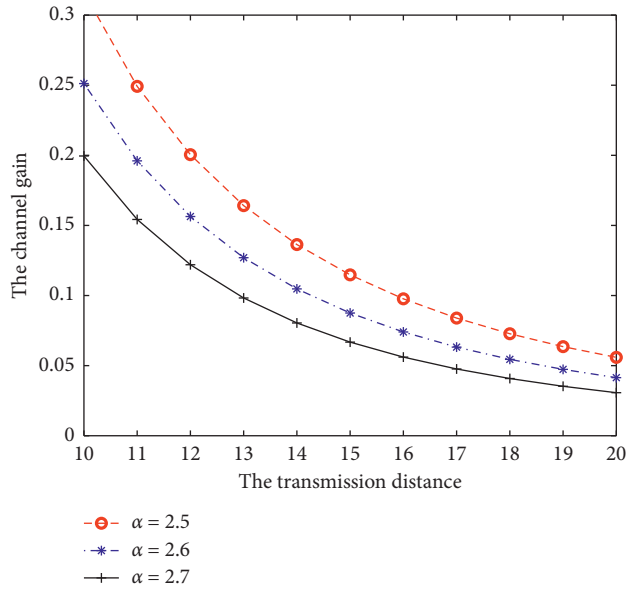


Figure 2: Impact of different values of the path loss exponent α with the change of transmission distance.

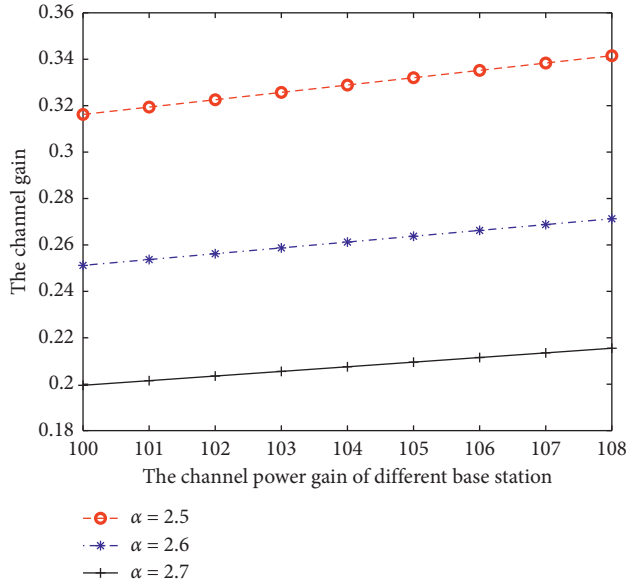


Figure 3: Impact of different values of the path loss exponent α with the change of power gain of different base station.

the total transmission rate of RTA is unstable due to its randomness, and the STA's fixed channel selection strategy keeps the total transmission rate unchanged. Notably, the transmission rate obtained by UTCS algorithm converges after 11 iterations. In the convergent state, we compare the total transmission rates obtained by three algorithms. Specifically, the total transmission rate obtained by UTCS algorithm is 2.15% higher than the maximum transmission rate of RTA and 4.3% higher than STA. It is worth noting that the random channel selection of RTA has a certain probability to make the total transmission rate reach the global optimal state; that is, the total transmission rate obtained by RTA is higher than that of UTCS algorithm. However, the probability of applying RTA to reach the global optimal transmission rate is very low and the global optimal state does not consider the maximum of each user's transmission rate, which does not conform to the goal of the model. Therefore, UTCS algorithm can perform the best.

In heterogeneous network, there is no channel multiplexing among MUs. Therefore, channel interference does not exist among MUs. The data transmission of SUs needs to reuse channel resources, so we analyze the impact of the number of SBSs deployed on the average transmission rate of users. In Figure 9, with the increase of the SBSs, the average transmission rate of users shows a downward trend after rising first. Because the number of MUs is less than the number of channels, channel resources are underutilized. When SUs reuse channel resources for data transmission,

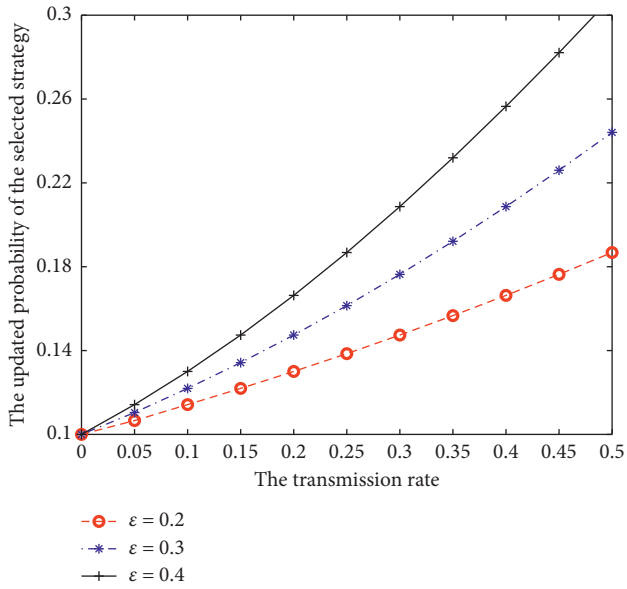


Figure 4: Impact of different values of the update step ε on update rate of the selection probability.

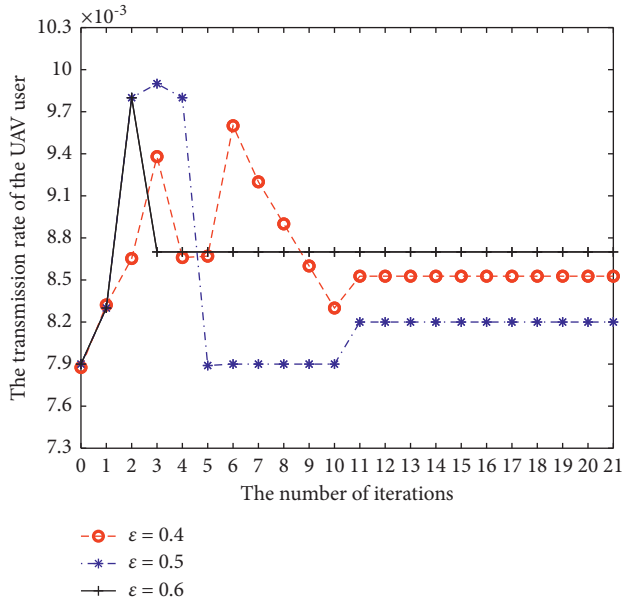


Figure 5: Impact of different values of the update step ε on the convergence rate.

some channel resources are idle. Therefore, the interference is relatively low, and the average rate of users increases. In peak state, channel reaches saturation of resource utilization, and interference increases at a faster rate, so the average transmission rate of users goes down. When the number of SBSs reaches 5, the average transmission rate obtained by UTCS algorithm is 7.36% higher than that of RTA and 4.63% higher than that of STA. Therefore, UTCS algorithm can effectively reduce interference and slow down the rate of decline of the average transmission rate.

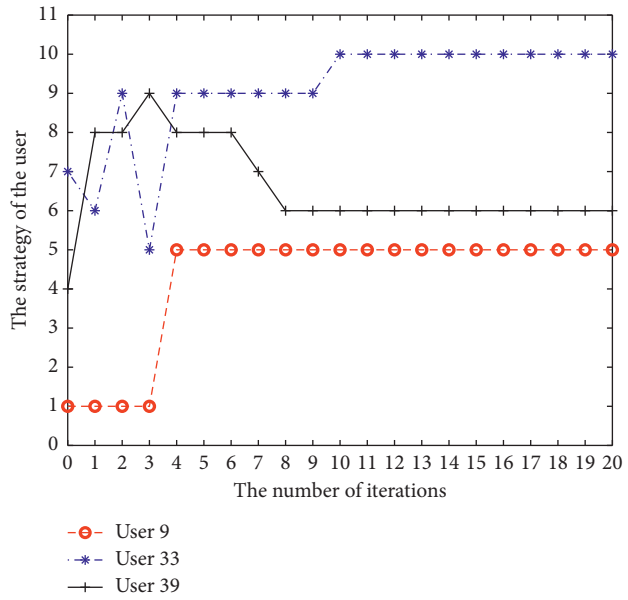


Figure 6: The convergence of channel selection strategy of the UAV users.

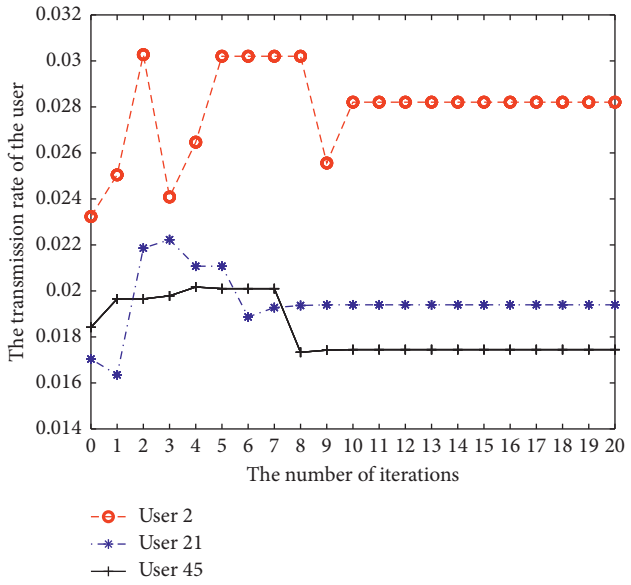


Figure 7: The convergence of transmission rate of the UAV users.

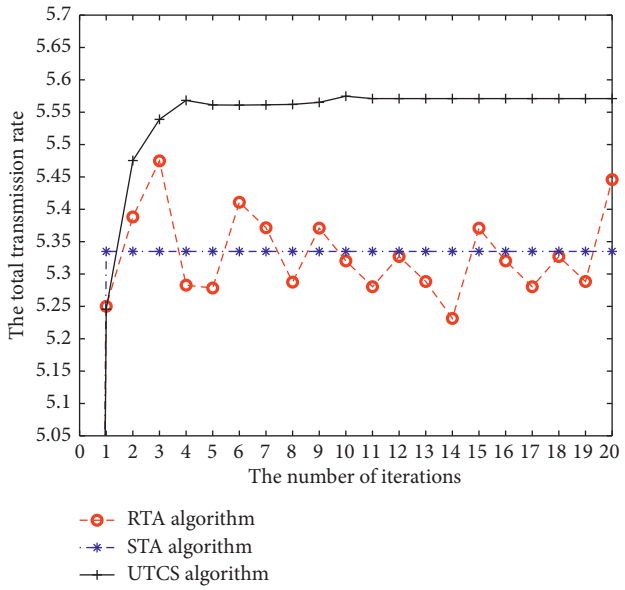


Figure 8: Performance comparison of different algorithms in terms of transmission rate for different iterations.

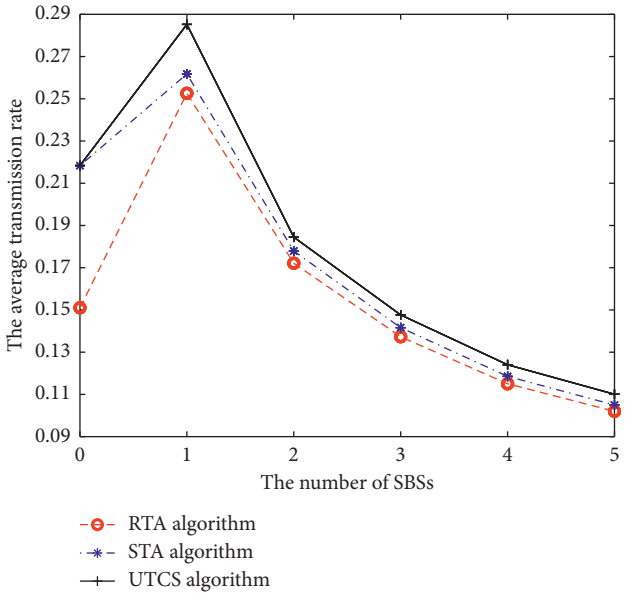


Figure 9: Performance comparison of different algorithms in terms of the total transmission rate for different number of SBSs.

6. Conclusion

In this paper, we study the channel selection for 5G heterogeneous networks to maximize the transmission rate of each user. We apply a noncooperative game method to construct the communication model and prove the existence of NE. A multiple UAV-enabled transmission channel selection (UTCS) algorithm has been proposed to obtain the equilibrium strategy profile of all the UAV users. Experimental results demonstrate that the UTCS algorithm can converge and can perform the best.

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 in part by the National Natural Science Foundation of China under Grants 61902029, 61972414, and 61 973161, the Excellent Talents Projects of Beijing under Grant 9111923401, the Scientific Research Project of Beijing Municipal Education Commission under Grant KM202011232015, the Beijing Nova Program under Grant Z201100006820082, the Beijing Natural Science Foundation under Grant 4202066, and the Fundamental Research Funds for Central Universities under Grant 2462018YJRC040.

References

[1] R. Bi, Q. Liu, J. Ren, and G. Tan, "Utility aware offloading for mobile-edge computing," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 239–250, 2021.

[2] D. Kim, J. Son, D. Seo, Y. Kim, and H. Kim, "A novel transparent and auditable fog-assisted cloud storage with compensation mechanism," *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 28–43, 2020.

[3] Y. Chen, F. Zhao, Y. Lu, and X. Chen, "Dynamic task offloading for mobile edge computing with hybrid energy supply," *Tsinghua Science and Technology*, 2021.

[4] J. Huang, B. Lv, Y. Wu, Y. Chen, and X. Shen, "Dynamic admission control and resource allocation for mobile edge computing enabled small cell network," *IEEE Transactions on Vehicular Technology*, p. 1, 2021.

[5] J. Mabrouki, M. Azrour, G. Fattah, D. Dhiba, and S. E. Hajjaji, "Intelligent monitoring system for biogas detection based on the internet of things: mohammedia, Morocco city landfill case," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 10–17, 2021.

[6] Y. Chen, F. Zhao, X. Chen, and Y. Wu, "Efficient multi-vehicle task offloading for mobile edge computing in 6G networks," *IEEE Transactions on Vehicular Technology*, p. 1, 2021.

[7] Y. Zhang, K. Wang, Q. He et al., "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Transactions on Services Computing*, vol. 14, no. 5, pp. 1333–1344, 2021.

[8] X. Zhang, H. Huang, H. Yin, D. O. Wu, G. Min, and Z. Ma, "Resource provisioning in the edge for IoT applications with multilevel services," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4262–4271, June 2019.

[9] T. Zhao, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[10] Y. Liu, D. Li, S. Wan et al., "A long short term memory based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 135–151, 2021.

[11] Y. N. Malek, M. Najib, B. Mohamed, and M. Essaaidi, "Multivariate deep learning approach for electric vehicle speed forecasting," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 56–64, 2021.

[12] Y. Zhang, J. Pan, L. Qi, and Q. He, "Privacy-preserving quality prediction for edge-based IoT services," *Future Generation Computer Systems*, vol. 114, pp. 336–348, 2020.

[13] X. Zhang, H. Chen, Y. Zhao et al., "Improving cloud gaming experience through mobile edge computing," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 178–183, August 2019.

[14] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1634–1644, 1 Oct.-Dec.

[15] W. Zhang, X. Chen, and J. Jiang, "A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems," *Tsinghua Science and Technology*, vol. 26, no. 1, pp. 95–111, 2021.

[16] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-Aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1145–1153, 2021.

[17] S. Mousavi, F. Afghah, J. D. Ashdown, and K. Turck, "Leader-follower based coalition formation in large-scale UAV network, a quantum evolutionary approach," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Honolulu, HI, USA, April 2018.

[18] J. Li and Y. Han, "A traffic service scheme for delay minimization in multi-layer UAV networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5500–5504, June 2018.

[19] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X communications: design and optimization for multi-UAV networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 1346–1359, Feb. 2019.

[20] J. Gu, T. Su, Q. Wang, X. Du, and M. Guizani, "Multiple moving targets surveillance based on a cooperative network for multi-UAV," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 82–89, April 2018.

[21] A. Bera, S. Misra, and C. Chatterjee, "QoE analysis in cache-enabled multi-UAV networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6680–6687, 2020.

[22] N. Zhao, X. Pang, Z. Li et al., "Joint trajectory and precoding optimization for UAV-assisted NOMA networks," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3723–3735, May 2019.

[23] Y. Dai, M. Sheng, J. Liu, N. Cheng, X. Shen, and Q. Yang, "Joint mode selection and resource allocation for d2d-enabled NOMA cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6721–6733, July 2019.

[24] M. A. Shattal, A. Wisniewska, B. Khan, A. Al-Fuqaha, and K. Dombrowski, "from channel selection to strategy selection: enhancing VANETs using socially-inspired foraging and deference strategies," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8919–8933, Sept. 2018.

[25] H. Ko, J. Lee, and S. Pack, "Joint optimization of channel selection and frame scheduling for coexistence of LTE and WLAN," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6481–6491, July 2018.

[26] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 729–743, 2020, Feb. 2020.

[27] D. Liu, Y. Xu, J. Wang et al., "Self-organizing relay selection in UAV communication networks: a matching game perspective," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 102–110, December 2019.

[28] D. Wu, X. Sun, and N. Ansari, "An FSO-based drone assisted mobile access network for emergency communications," *IEEE*

Transactions on Network Science and Engineering, vol. 7, no. 3, pp. 1597–1606, 2020, July-Sept. 1 2020.

[29] N. Zhang, S. Zhang, J. Zheng, X. Fang, J. W. Mark, and X. Shen, "QoE driven decentralized spectrum sharing in 5G networks: potential game approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 7797–7808, 2017.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 8032726, 10 pages
https://doi.org/10.1155/2022/8032726

Research Article

Invoice Detection and Recognition System Based on Deep Learning

Xunfeng Yao, Hao Sun, Sijun Li, and Weichao Lu

Jinling College, Nanjing University, Nanjing, China

Correspondence should be addressed to Xunfeng Yao; 030504@jlxy.nju.edu.cn

Received 13 August 2021; Accepted 29 September 2021; Published 25 January 2022

Academic Editor: Xuyun Zhang

With the development of economy and information technology, a large amount of invoice information has been produced. As one of the important components of the industrial Internet of Things, the recognition of invoice information is urgent to realize its intelligent recognition. Most invoice issuing units basically adopt traditional manual identification methods for the processing of invoices. As the number of invoices increases, problems such as low efficiency in identifying invoice information, error-prone, and difficulty in ensuring security frequently appear. In response to the above problems, this paper designs and implements an invoice information recognition system based on deep learning. The system first solves the problems of low image contrast and lack of image due to poor lighting or noise effects by image preprocessing methods such as image graying and normalization. Second, a target detection and invoice recognition method based on the combination of YOLOv3 + CRNN two models is proposed, and an end-to-end invoice information recognition model is obtained. Finally, the model is used to develop an invoice detection and recognition system based on deep learning. Experiments have verified that the system has the characteristics of high recognition accuracy and high efficiency, which can accurately identify invoice content information and reduce the loss of manpower and material resources.

1. Introduction

Foreign research on invoice recognition system originated in the 1960s and 1970s. But, most research is only on methods of invoice recognition and digital recognition. The concept of OCR (Optical Character Recognition) technology was first proposed by German scientist Tausheck in 1929. As an important part of pattern recognition, OCR is used to identify the information in the image and extract it into computer readable [1]. Until about the 1960s, Japan began to study the basic recognition theory of OCR. After more than ten years of research, it developed a simple recognition system such as postal code recognition, which realized the automatic recognition of codes on mails [2]. After 1970, China began to study OCR technology and first carried out relevant research on Chinese character recognition. Until 1986, Tsinghua University and other universities developed an invoice recognition system based on OCR technology, and Chinese OCR invoice recognition products came out [3]. Due to the low recognition rate of the early invoice system and insufficient productization, it has not been popularized in life. With the rise of artificial intelligence, more systems for invoice recognition have begun to appear on the market. For example, Baidu's OCR recognition system and Tencent's OCR recognition system both use in-depth learning to detect and recognize invoice information [4].

Once deep learning has emerged, it has been widely used in speech recognition, image recognition, and natural language processing. In 2011, Google applied deep learning to speech recognition and successfully reduced the error rate [5, 6]. In the field of image recognition, researchers have further proposed a large-scale deep convolutional neural network, which reduces the error detection rate to 15.3% [7]. In 2015, He et al. proposed the ResNet architecture to improve the accuracy of the algorithm by increasing the amount of data during training [8]. Deep learning has developed rapidly in image recognition, and target detection technology has been applied to text localization in natural scenes. Girshick et al. proposed that R-CNN successfully

applied deep learning to target detection. First, the selective search algorithm was used to select candidate boxes, and then the candidate boxes were sent to the convolutional neural network for classification, but the extracted candidate boxes overlapped a lot, and feature extraction redundancy exists [9, 10]. Later, the improved FastR-CNN algorithm in the research inputs the entire image into the convolutional neural network and then maps the candidate frame on the feature map, avoiding repeated feature extraction and improving the training speed [11]. The concept of anchor frame is proposed in Faster R-CNN in [12], and the extraction of candidate frames is also realized by convolutional network, which effectively reduces the selection time of candidate frames. Dai et al. proposed to integrate the target location information into the ROI pooling layer to construct a location-sensitive score map, which effectively solves the problem of the destruction of the translation invariance of the convolutional network [13]. In order to adapt features to targets of different sizes, Lin et al. proposed a feature pyramid structure for small target detection [14]. Although the above-mentioned algorithm has high detection accuracy, it cannot achieve a real-time effect. In order to solve the efficiency problem, Redmon et al. proposed to use a single-structure convolutional neural network to directly predict the location and category of the target, but the accuracy is slightly lower [15]. Later, an improved YOLOv2 algorithm [16] was proposed, and a batch normalization layer [17] (Batch Normalization) was added on the basis of YOLOv1 to speed up training, and anchor boxes and higher resolution classifiers were used to improve accuracy. Literature [18] improved the YOLOv2 network by changing the screening rules of the candidate frame and other methods and achieved relatively ideal results in the task of positioning the invoice recognition image. In 2014, Liu et al. proposed the SSD algorithm, which takes into account both speed and accuracy, but the shallow feature expression ability of its prediction layer is not strong [19]. In order to strengthen the expressive ability of shallow features, Fu et al. proposed to use deconvolution to add contextual information to the feature map, and the accuracy of the model was further improved [20]. After YOLOv2, Redmon proposed the third version of the YOLO series, YOLOv3 [21]. This algorithm uses Faster R-CNN to extract features to improve the speed of target detection, which is very suitable for natural scenes with multiple anticounterfeit feature detection in invoices.

In order to meet the requirements of efficiently identifying invoice data in engineering applications, this paper first uses the YOLOv3 algorithm for text target detection training. Second, the deep learning CRNN model is used to identify the content of the invoice. Finally, the two models are combined to obtain an end-to-end invoice recognition model, which is verified by the test set, and the recognition result is compared with the recognition result of the traditional OCR technology.

2. Invoice Recognition System Based on Deep Learning

2.1. Invoice Detection Based on YOLOv3 Algorithm. The YOLO algorithm (You Only Look Once, YOLO) is a neural network model that can identify and detect objects and text. The execution process of the algorithm is mainly divided into two parts: (1) first classify the object; (2) identify the position of the object in the picture. Because YOLO's unique end-to-end design method simplifies object detection into a single regression problem, it avoids the problem of slow running speed and difficult model convergence. The emergence of the YOLO algorithm gives new ideas to the target detection task. The algorithm combines the two tasks of positioning and classification to make the image detection speed meet the requirements of real-time detection. YOLO is composed of four parts: input layer, convolution layer, pooling layer, and fully connected layer. Its network structure model is shown in Figure 1.

YOLO extracts the feature value through the convolutional layer CNN, and the final result of the predicted value is completed through the fully connected layer. As shown in Figure 1, among the 24 convolutional layers, channel reduction is first performed by 1×1 convolution, and then 3×3 convolution processing is used. In the convolution and fully connected layers, Leaky ReLU is used to activate the function: max (x, 0.1x), and YOLO uses a mean square error loss function. The positioning error refers to the error of the bounding box coordinate prediction. The calculation of the error uses a weight value of $\lambda_{\text{co ord}} = 5$. The confidence of the bounding box containing the target and the bounding box not including the target is calculated, and the other weight values are set as 1, using the mean square error of the model as the loss function. For bounding boxes with inconsistent sizes, the actual smaller bounding box is more sensitive. In order to solve the above phenomenon, the model changes the predicted value to $(x, y, \sqrt{w}, \sqrt{h})$. The principle is that each cell can predict multiple bounding boxes, and each bounding box corresponds to the corresponding category. Select the bounding box with the largest IOU with the ground truth for the task of predicting the target. Other bounding boxes will ignore the existence of the target to obtain a specialized cell corresponding to the bounding box. For the bounding box that does not have a corresponding target, its error term is confidence. The loss function formula of YOLO is shown in the following formula:

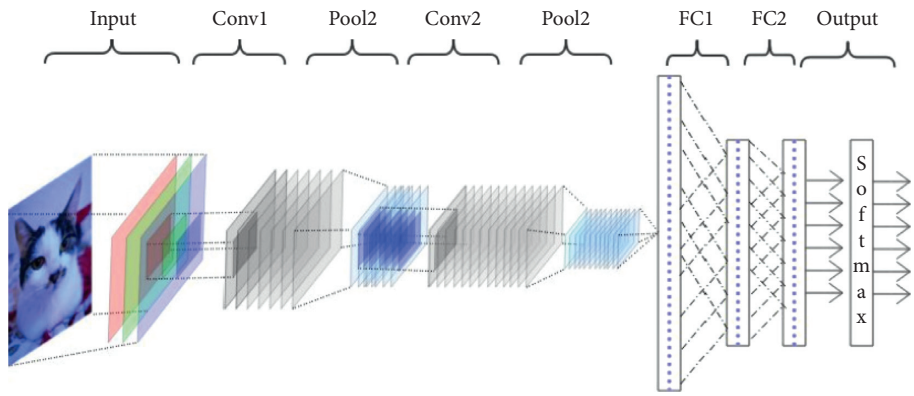


Figure 1: YOLO network structure diagram.

$$\begin{aligned}\text{loss} = {} & \lambda_{\text{co ord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{\text{co ord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 + \lambda_{\text{co ord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 + \sum_{i=0}^{S^2} 1_{ij}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2.\end{aligned} \tag{1}$$

The YOLOv3 algorithm used in this paper combines the advantages of other algorithms with the YOLO algorithm so that the algorithm further improves the accuracy of model detection while maintaining the speed advantage, especially for small target tasks. The improvement of the detection ability is more obvious. YOLOv3 improves model performance mainly by adjusting the network structure. This method uses the new backbone network Darknet-53, and at the same time, it uses multiscale features to detect target tasks by constructing an FPN network. The Darknet-53 network structure has 53 convolutional layers, which combines the advantages of the residual network with shortcut connections between some layers. The specific structure of Darknet-53 is shown in Figure 2.

YOLOv3 mainly uses the remaining 52-layer network structure of darknet-53 except for the fully connected layer. In order to improve the accuracy of the algorithm for detecting small target tasks, YOLOv3 uses a fusion method similar to FPN to perform detection on multiple scale feature maps. The 3 prediction routes of YOLOv3 are for three convolutional structural layers; the number of convolution kernels in the last convolutional layer is 255, which is for the 80 categories of the COCO data set: $3 * (80 + 4+1) = 255$, where 3 represents that a grid cell contains 3 bounding boxes, 4 represents the 4 coordinate information selected by the box, and 1 represents the objectness score. In the Darknet-53 network, $256 * 256 * 3$ is used as input, and the leftmost column of numbers represents repeated residual components. Each residual component has two convolutional layers and a shortcut link. The residual component of the specific direct connection method is shown in Figure 3. Input x to the output process, and the output result is $f(x) + x$. When $f(x) = 0$, $H(x) = x$, at this time, the residual result approaches 0, and the model converges.

YOLOv2 uses the pass-through structure to identify and detect fine-grained features, while the YOLOv3 method uses three different scale feature maps to identify and detect objects based on the YOLOv2 method. Among them, in the first scale, some convolutional layers are added after the traditional basic network for sampling, the sampling multiple is high, and the perception field is large, so this scale is suitable for large object detection; the second scale is from the 79th layer upwards; convolution and sampling are added to the last 16×16 feature map. This scale is suitable for medium-sized object detection; the third-scale tree uses a 32×32 feature map, which is suitable for small object detection. YOLOv3 extends the K-means clustering of YOLOv2, which takes the form of a priori frame size, sets 3 a priori frames for each downsampling scale, and finally clusters a priori frames of 9 sizes. See Table 1 for the specific allocation of a priori boxes of 9 scales.

As shown in Table 1, when the feature map is on a 13×13 feature map with a larger receptive field, a larger prior frame is needed to detect a larger target. When on a 26×26 feature map, a medium a priori box needs to be used to detect medium-sized objects. When on the 52×52 minimum receptive field feature map, a smaller prior frame is needed to detect smaller objects.

Unlike YOLOv1 and YOLOv2, which both use the mean square error as the loss function, YOLOv3 uses the cross-entropy loss function to calculate the coordinate loss. YOLOv3 improves the category prediction function, and the softmax layer will no longer be used. The essence of the softmax layer in the classification network is that a category contains an attribute, such as an image or an object. But when in a complex scene, an object can contain multiple categories. For example, there are two categories of woman and person in the category of people, and there is a woman in an image, which corresponds to the category label in the

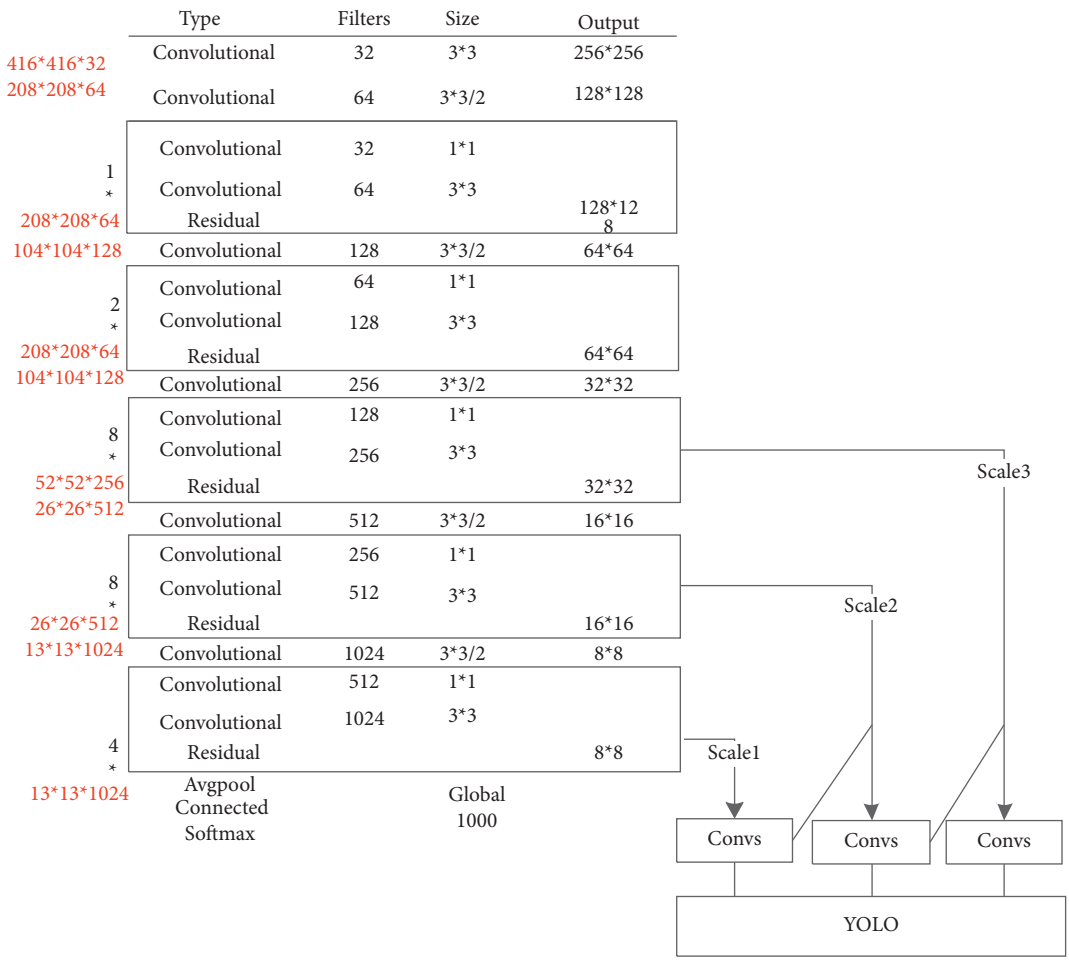


Figure 2: Darknet-53 structure details.

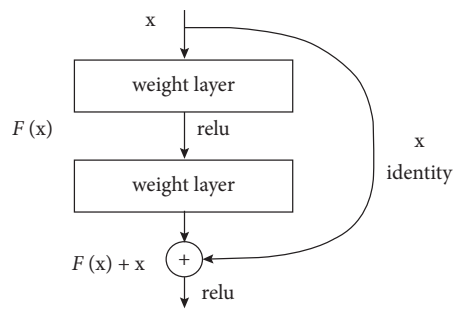


Figure 3: Schematic diagram of the residual group structure.

Table 1: A priori box allocation resources.

Feature map	13 * 13			26 * 26			52 * 52		
Receptive field	Big			Middle			Small		
Anchor	116 * 90	156 * 198	373 * 326	30 * 61	62 * 45	59 * 119	10 * 13	16 * 30	33 * 23

model detection result. There are two classes of woman and person at the same time, which belong to the multilabel classification. For this type of problem, softmax will choose the category with the largest prediction probability, which will eventually result in only one category being detected by woman and person. In order to solve the above problems, YOLOv3 uses a logistic regression layer for classification to obtain different categories. The logistic regression layer uses the sigmoid function. The sigmoid function can control the output between 0 and 1. Therefore, the sigmoid function is

used to control the output of a certain category after the feature is extracted. The value is greater than 0.5. It can be seen that this category belongs to this category; otherwise, it does not belong to this category, so that a box can predict multiple categories in this image, and the cost function here is the cross entropy of sigmoid. The IoU loss function and focal loss are used in the YOLOv3 target detection algorithm, and 1-GIoU is directly used as the bounding box regression loss function to replace the original mean square error and loss function. The focal loss based on the cross-entropy loss is used as the loss function of the confidence of the bounding box object. The target classification loss uses the classical cross entropy as the loss function, using the GIoU loss function and the focal loss function, and the resulting YOLOv3 loss function is shown in the following formula:

$$\begin{aligned}\text{loss} &= b\,\text{boxloss} + \text{confidenceloss} + \text{classloss}\\ &= \sum_{0}^{\text{cell\_number\_}B} I^{\text{object}} \times \left(1 - GIoU_{\text{predict}}^{\text{groud\_truth}}\right)\\ &\quad + \sum_{0}^{\text{cell\_number\_}B} m \times \text{focal\_loss}\left(CE\left(p_0, q_0\right)\right)\\ &\quad + \sum_{0}^{\text{cell\_number\_}B} I^{\text{object}} \times \sum_{0}^{c} CE\left(p(c), q(c)\right).\end{aligned} \tag{2}$$

In the bounding box regression loss function (bboxloss) part, the original mean square error and loss function are replaced by the GIoU loss function. The loss function also adds the focus loss to the boundary box confidence cross entropy loss function, so as to balance the loss proportion of easy samples and difficult samples.

2.2. CRNN-Based Invoice Edge Detection and Recognition. In order to accurately extract the information in the invoice, it is necessary to detect and identify the boundary of the detected invoice. What edge detection can do is to identify the points with obvious brightness changes in the digital image. This process can discard the redundant information in the image, thereby reducing some unnecessary processing. This method improves the information extraction rate. At the same time, important boundary information in the image is retained. This paper uses the CRNN (Convolutional Recurrent Neural Network) to detect the edges of invoices. This network recognizes text sequences of variable length end-to-end, so there is no need to cut individual texts in advance but convert text recognition into a sequence learning problem dependent on timing, that is, image-based sequence recognition. According to the characteristics of Chinese handwriting recognition, this paper improves the CRNN network in order to solve the problem of Chinese handwriting recognition. The network structure of the improved CRNN is composed of three parts, which from bottom to top are the Deep Convolutional Layer, Recurrent Layer, and Transcription Layer. The structure of CRNN is shown in Figure 4.

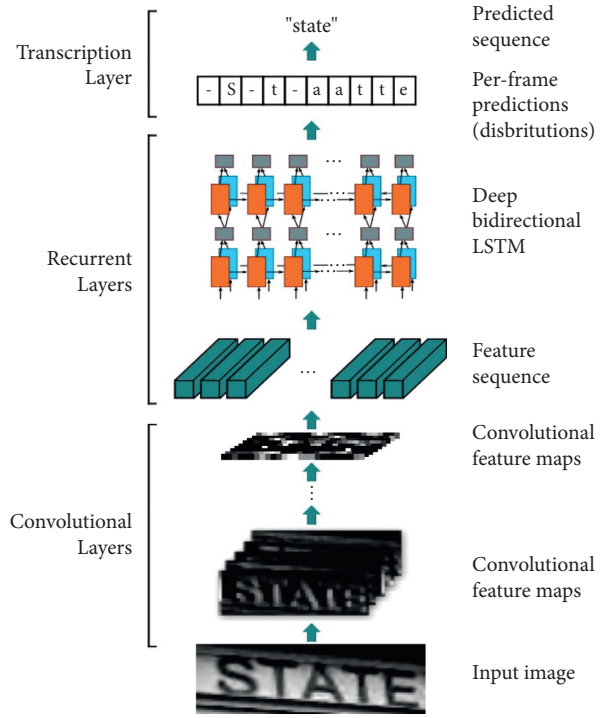


Figure 4: Schematic diagram of CRNN structure.

The specific steps of CRNN model training are as follows:

Step 1. According to the input requirements of the CRNN 7 model, process the data, generate a large batch of invoice sample data, divide the data into the test set and the training set according to the ratio of 9 : 1, and label the image according to the requirements of the CRNN training set.

Step 2. According to the generated data set, set the three major parameters.

Step 3. Design the loss function calculation method in training.

Step 4. Design the entire network training process, loop each epoch, and perform model verification and model storage when the specified number of iterations is reached. The specific process is shown in Figure 5.

3. Experiment

In this paper, the invoice image data is obtained through a scanner device. However, in the process of collecting data, the invoice image will have some noise effects due to environmental changes and improper human operations. Therefore, this paper first preprocesses the invoice data. The process of image preprocessing in this paper includes operations such as normalization, grayscale, and edge detection of the original image, and finally, a binary image whose size meets the model input is obtained through the preprocessing operation. Since the invoice is usually placed randomly by hand during scanning, there will be a certain angle of

Figure 5: Edge detection step process.

inclination. In order to reduce the influence on the subsequent information area positioning and character cutting, this paper uses the Hough line detection method to correct the inclination of the invoice image.

3.1. Experimental Environment. The experiment in this paper is to complete programming and testing on a PC, and the operating system is Windows 7, 64 bits. The programming language is C++, and the system interface is built with MFC. The OpenCV library is needed for image processing, and the LibXL library is needed for data logging to Excel. The experimental environment of this system is shown in Table 2.

OpenCV library is an open-source machine vision development library commonly used at present, which already contains many general algorithms, and the image processing of this system is used for development. MuPDF library is a powerful PDF parser. It is used in this system to convert scanned pdf format images into jpg image format. LibXL library is a package library that implements Excel operations, which is used to automatically save the information recognition results to an Excel table.

3.2. Collection of Data Sets. The collection of invoice images is the initial step of the operation of the entire system. There are generally three methods for collecting invoice images. The first is to capture dynamic video through a camera. This method obtains the invoice data picture by intercepting the invoice information in the video. This process is time-consuming and laborious and the final image obtained is not high-definition; the second one is to collect still images with a high-definition digital camera. This method will generate different edge background information due to different shooting angles or heights; the third is to scan the invoice into a color, grayscale, or binary image through a scanner device. In order to improve the efficiency of invoice recognition and reduce the expenditure of manpower and material resources, this paper chooses a high-definition scanner to obtain invoice mages. The image obtained by the scanner can be saved as a color image, gray image, or binary image. Although the color image is the closest to the real scene, it contains a huge amount of information and a complex color model. It will increase the amount of calculation and time overhead when processing the image, so in general, it is not saved as a color image after scanning. The grayscale image has only one sample color, and the original unclear area in the image can be made clearer through image enhancement technology, and the uninteresting area can also be suppressed. Therefore, the grayscale image is also the input image that people often choose as image processing. All pixels in a binary image have only two values, 0 and 1. Therefore, the data type in the computer generally only occupies 1 binary bit. Comparing the above three images, the color image contains too much information, and the calculation speed is slow; although the calculation of the binary image is simple, the digital information in the invoice image is generally relatively small, and some useful information will be lost after the binarization process. The degree map is a compromise between the two. In order to take into account the recognition rate of numbers and the speed of the system, this paper chooses to save the collected invoice images as grayscale images. This paper uses a D16A3 Jieyu high-speed scanner, which is a high-definition high-speed scanner with a resolution of 4608×3408 dpi and uses the BMP image format for scanning. The pictures collected by this scanner are shown in Figure 6.

3.3. Data Normalization. Since the image of the invoice is obtained by manually operating the scanner when the image is collected, the size of the image obtained by scanning in different environments is different. In order to facilitate the follow-up model to monitor and identify the invoice data information, this paper normalizes the invoice data uniformly.

Image normalization methods mainly include linear and nonlinear processing methods. The advantage of the linear normalization method is that it can retain the linear nature of the original image to a certain extent. The nonlinear normalization will change the quality center of the image and affect the recognition accuracy. Therefore, this paper uses bilinear interpolation to normalize the invoice image to a size of 1245×730.

The bilinear interpolation is shown in Figure 7. The target pixel point $R(i, j)$ is obtained by bilinear interpolation. The four points in the original image are known to be $A11(i1, j1)$, $A11(i1, j2)$, $B21(i2, j1)$, and $B22(i2, j2)$. The principle of using bilinear interpolation to normalize the image is as follows:

$$
\begin{aligned}
f(i, j_1) &= \frac{i_2 - i}{i_2 - i_1} f(i_1, j_1) + \frac{i - i_1}{i_2 - i_1} f(i_2, j_1), \\
f(i, j_2) &= \frac{i_2 - i}{i_2 - i_1} f(i_1, j_2) + \frac{i - i_1}{i_2 - i_1} f(i_2, j_2), \\
f(i, j) &= \frac{j_2 - j}{j_2 - j_1} f(i_1, 1) + \frac{j - j_1}{j_2 - j_1} f(i, j_2).
\end{aligned} \tag{3}
$$

The background in the invoice image is more complicated, and the character spacing is small, which brings greater difficulties to positioning and character cutting. Therefore, this paper designs a fast and accurate positioning and cutting information area positioning algorithm. The specific algorithm flow is shown in Figure 8.

Table 2: Experimental environment table.

Computer configuration	Portable PC, CPU clocked at 2.0 MHz, memory 4G, 64-bit operating system
Operating system	Windows 7
Development environment	VS2013, MFC, and caffe
Open-source library	OpenCV 2.4.10, MuPDF library, and LibXL library

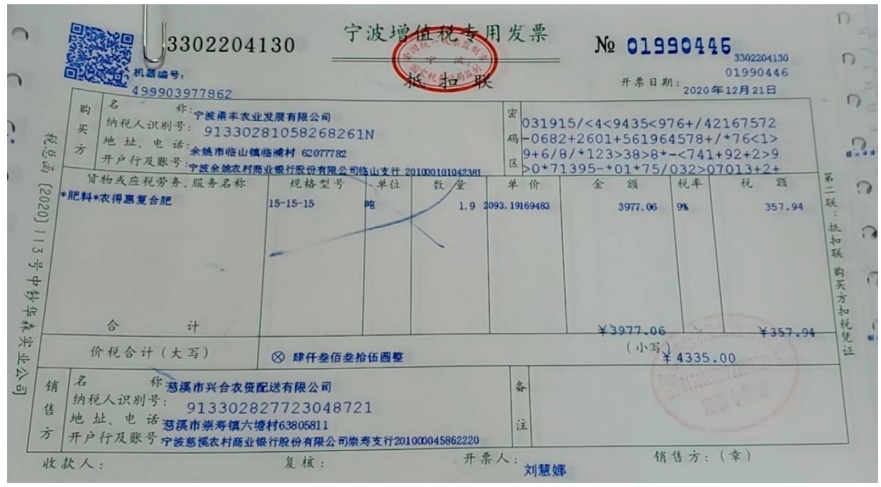


Figure 6: Schematic diagram of invoice data.

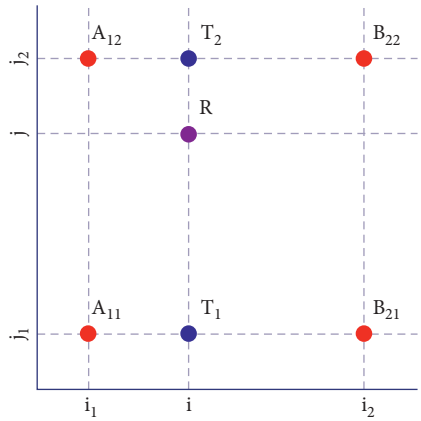


Figure 7: Bilinear interpolation.

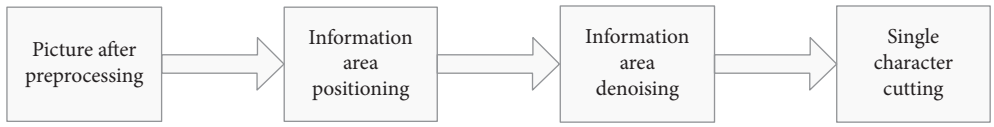


Figure 8: Information area positioning algorithm flow.

Information area positioning is to analyze and locate according to the characteristics of the invoice layout and extract useful information areas such as invoice numbers from the invoice image after data preprocessing. The location of the information area in this paper is mainly based on the characteristics of each functional unit on the invoice page, and a large amount of prior knowledge is obtained through experiments to realize the extraction of the information area. First, use the characteristic of each information area to have a fixed position in the invoice layout, combined with prior knowledge, directly obtain the rough positioning of each information area, then use the symbolic features contained in each information area, and use the method of template matching to compare the roughly extracted information. The area is further accurately positioned, and accurate digital string information is obtained [22].

Information area positioning is a crucial step in the image recognition of invoices. After the previous tilt

Figure 9: Information area positioning.

correction processing, the next step is to extract useful information areas from the entire invoice image to facilitate subsequent character cutting and recognition operations. As value-added tax invoices are unified across the country and have the same and fixed layout structure, the characteristics of the invoice structure can be used to obtain useful information areas. This paper realizes the information extraction of the amount, tax amount, taxpayer identification number, and invoice number, as shown in Figure 9.

The amount and the RMB symbol in the tax information area are matched, the standard template prepared in advance is imported into the database, and the target image uses the matchTemplate() function provided by OpenCV to match the image area that overlaps the template. The result of template matching is shown in Figure 10.

The red rectangular box in Figure 10 is the matching result of the standard squared deviation matching method. By analyzing the digital information of the amount and tax area in the invoice image, it can be seen that the height of the numbers in these two areas is similar to the height of the RMB symbol, and the distance between the RMB symbol and the number string is greater than the distance between the numbers. Therefore, take the upper right corner of the red rectangular box, that is, the upper right corner of the RMB symbol, as the reference point, and assume that the reference point is (x, y). Through experimental analysis, select a suitable point $(x-1, y-3)$ as the starting point, extract the region of interest, and use the height of the RMB symbol template image as the height of the region to be extracted to obtain the precise digital string region as shown in Figure 11.

The amount and tax information area described above are the same, and the positions of the taxpayer identification number and invoice number in the invoice image are also unchanged. Therefore, the recognition process of the regional positioning of the taxpayer identification number and the invoice number is basically the same as the previous positioning principle. This process first directly obtains the value processing of its subregions based on prior knowledge and then uses the standard square deviation matching method to find the precise region of the number string. Different from the traditional method, this paper uses two template matching methods to extract the number string. Although the taxpayer identification number and the invoice number also have specific identifiers in front of the numeric string and they are in a fixed information area, it is inevitable that the printing is unreasonable. At this time, the position of the number string relative to the identifier will be shifted or tilted. In this case, two template matching methods are needed to extract the string. The result of extracting the information area of the taxpayer identification number in

Figure 10: Matching result map.

Figure 11: Accurate result graph.

this paper can accurately find its location, but the corresponding number string has a significant offset, and the offset location is not fixed. It may be a downward offset, or it may be offset. It is an upward shift. If the number string area is directly obtained based on prior knowledge, some data information may be lost. Here, first obtain subregion 2 based on the prior knowledge, given a wider range, so that the number string can be completely contained in the region. Then, analyze the characteristics of the number string. Both the taxpayer identification number and the invoice number are composed of more numbers, and after statistics, it is found that almost all taxpayer identification numbers have the number "1." Therefore, the number "1" is used as the template image, subarea 2 is used as the target image, and the standard square error matching method is used for matching again. The result is shown in the figure. Although subregion 2 contains Chinese characters and other noises, the structure of Chinese characters is more complicated, and the structure of Arabic numerals is quite different. Therefore, the Chinese characters in the picture do not affect the matching effect. The red rectangle is the matching result, which accurately matches "1" in the number string, which is equivalent to finding the position of the number string. Finally, according to the position of the matched coordinate point, deduct the number string in the subarea and get the accurate number string area as shown in Figures 12–16.

The main research of this paper is the recognition of uppercase amounts. The research includes a brief description of convolutional neural networks and residual networks, the preprocessing of uppercase amounts of character data, and the production of data sets, as well as a summary analysis of the test results of the two networks. This chapter combines with the previous information detection, edge detection, information identification extraction, and OCR identification to form an intelligent identification system, which automatically recognizes the reimbursement content after importing invoices. Integrating this system with the financial system of related institutions can realize intelligent financial reimbursement.

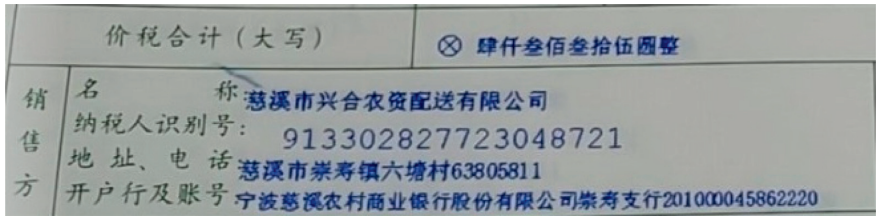


Figure 12: Small area of invoice.

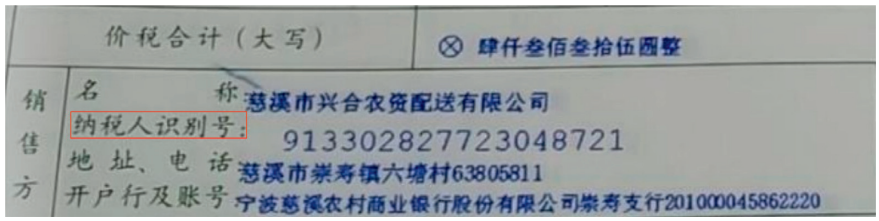


Figure 13: The first matching template.

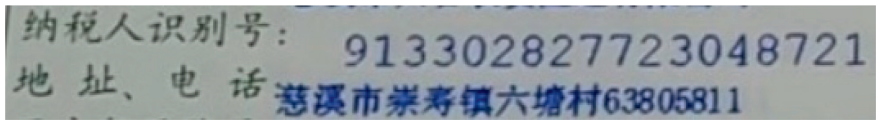


Figure 14: The second match result.

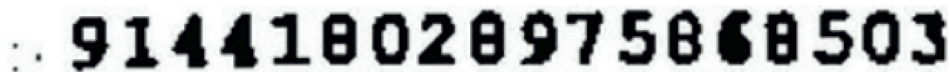


Figure 15: Test results.

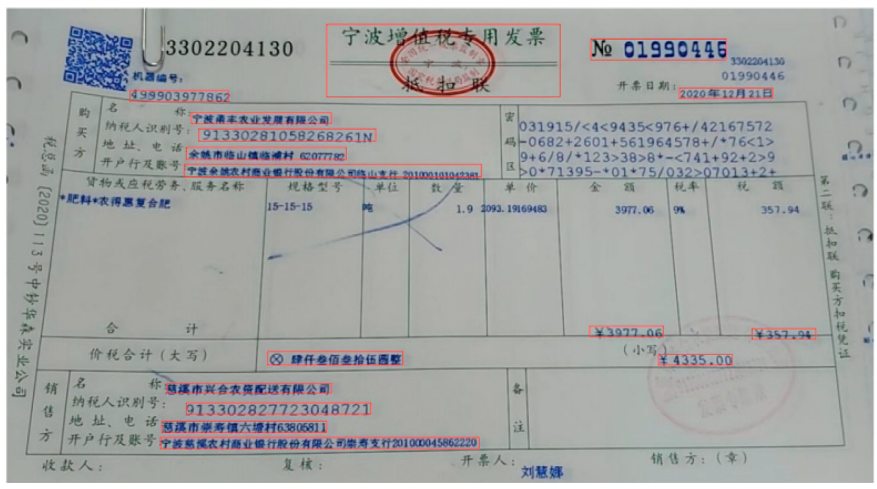


Figure 16: Overall result of invoice recognition.

4. Conclusion

With the vigorous development of artificial intelligence, the automatic invoice recognition system has also received more and more attention. At present, most of the existing recognition methods, such as Monarch Butterfly Optimization (MBO), Earthworm Optimization Algorithm (EWA), Elephant Swarm Optimization (intelligent algorithms such as EHO), and moth search (MS), are often used for image verification and recognition. Although this type of algorithm has a faster recognition rate, it is difficult to recognize problems such as invoices that have a small recognition area. Class methods generally have low recognition accuracy. This paper studies the status quo of invoice recognition and proposes an object detection and invoice recognition method based on the YOLOv3 + CRNN model. It locates the invoice information

area by marking the invoice dataset and realizes the detection and recognition of the VAT invoice information through image processing and deep learning. Finally, the system realized the rapid identification and processing of invoices. In future research, we can further optimize the information collection methods, solve subsequent data storage problems, and realize a more efficient and accurate invoice information detection and recognition system.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. Smith, "An overview of the Tesseract OCR engine," *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, pp. 629–633, IEEE, Curitiba, Brazil, September 2007.

[2] A. Gonzalez, L. M. Bergasa, and J. J. Yebes, "Text detection and recognition on traffic panels from street-level imagery using visual appearance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 228–238, 2013.

[3] E. L.-C. Lai and X. Yu, "Invoicing currency in international trade: an empirical investigation and some implications for the renminbi," *The World Economy*, vol. 38, no. 1, pp. 193–229, 2014.

[4] G. Jiuxiang, W. Zhenhua, K. Jason et al., "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

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

[6] L. Han, C. Tianding, T. Hualiang, and J. Yingtao, "A graph-based reinforcement learning method with converged state exploration and exploitation," *Computer Modeling in Engineering and Sciences: Computer Modeling in Engineering and Sciences*, vol. 118, no. 2, pp. 253–274, 2019.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[9] T. Jijun, C. Xiaolong, Z. Yingjie, and J. Lurong, "Real-time recognition and positioning of moving targets based on deep learning," *Computer Systems Applications*, vol. 27, no. 8, pp. 28–34, 2018.

[10] Z. Chaoping and Y. Yi, "Face detection and recognition in surveillance video based on YOLO2 and ResNet algorithm," *Journal of Chongqing University of Technology (Natural Science)*, vol. 8, pp. 170–175, 2018.

[11] Y. Nana, "Research on face detection algorithm based on deep learning," *Science and Technology Innovation Herald*, vol. 4, no. 26, p. 87, 2018.

[12] C. Shuhong, G. Xu, and C. Shuchun, "Moving vehicle detection based on computer vision," *Acta Metrology*, vol. 38, no. 3, pp. 288–291, 2017.

[13] H. Li, T. Chen, H. Teng, and Y. Jiang, "A graph-based reinforcement learning method with converged state exploration and exploitation," *Computer Modeling in Engineering and Sciences: Computer Modeling in Engineering and Sciences*, vol. 118, no. 2, pp. 253–274, 2019.

[14] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He, and X. Zhou, "TripRes: traffic flow prediction driven resource reservation for multimedia IoV with edge computing," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 2, pp. 1–21, 2021.

[15] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for Internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[16] K. Itakura and F. Hosoi, "Automatic tree detection from three-dimensional images reconstructed from 360° spherical camera using YOLO v2," *Remote Sensing*, vol. 12, no. 6, p. 988, 2020.

[17] F. Ming, S. Tengteng, and S. Zhen, "Fast helmet wearing condition detection based on improved YOLOv2," *Optics and Precision Engineering*, vol. 27, no. 5, pp. 1196–1205, 2018.

[18] N. Ganesh, R. K. Ghadai, A. K. Bhoi, K. Kalita, and X.-Z. Gao, "An intelligent predictive model-based multi-response optimization of EDM process," *CMES-Computer Modeling in Engineering & Sciences*, vol. 124, no. 2, pp. 459–476, 2020.

[19] J. Sheng, H. Min, Z. Qibing, and W. Zhenglai, "Research on pedestrian detection method based on R-FCN," *Computer Engineering and Applications*, vol. 54, no. 18, pp. 180–183, 2018.

[20] C. Rafael, N. E. Vera, J. Lucas, and F. V. Ferran, "An ETD method for American options under the heston model," *CMES-Computer Modeling in Engineering & Sciences*, vol. 124, no. 2, pp. 493–508, 2020.

[21] L. Cen, G. Lijun, Z. Rong, and H. Yetian, "Application of improved YOLOv3 algorithm in container number positioning," *Sensors and Microsystems*, vol. 46, no. 7, 2019.

[22] X. Xu, D. Zhu, X. Yang, S. Wang, L. Qi, and W. Dou, "Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–17, 2021.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 5358873, 14 pages
https://doi.org/10.1155/2022/5358873

Research Article

Research on Intelligent Scheduling Mechanism in Edge Network for Industrial Internet of Things

Zhenzhong Zhang,[1,2] Wei Sun,[1] and Yanliang Yu[3]

[1]*Center of Quantitative Economies, Jilin University, Changchun Jilin 130012, China*
[2]*Zhuhai College of Science and Technology, Zhuhai, Guangdong 519041, China*
[3]*School of Law and Social Work, Dongguan University of Technology, Dongguan, Guangdong 523000, China*

Correspondence should be addressed to Yanliang Yu; yyl3039@email.poe.edu.pl

Received 13 August 2021; Accepted 21 October 2021; Published 5 January 2022

Academic Editor: Xuyun Zhang

With the vigorous development of the Internet of Things, the Internet, cloud computing, and mobile terminals, edge computing has emerged as a new type of Internet of Things technology, which is one of the important components of the Industrial Internet of Things. In the face of large-scale data processing and calculations, traditional cloud computing is facing tremendous pressure, and the demand for new low-latency computing technologies is imminent. As a supplementary expansion of cloud computing technology, mobile edge computing will sink the computing power from the previous cloud to a network edge node. Through the mutual cooperation between computing nodes, the number of nodes that can be calculated is more, the types are more comprehensive, and the computing range is even greater. Broadly, it makes up for the shortcomings of cloud computing technology. Although edge computing technology has many advantages and has certain research and application results, how to allocate a large number of computing tasks and computing resources to computing nodes and how to schedule computing tasks at edge nodes are still challenges for edge computing. In view of the problems encountered by edge computing technology in resource allocation and task scheduling, this paper designs a dynamic task scheduling strategy for edge computing with delay-aware characteristics, which realizes the reasonable utilization of computing resources and is required for edge computing systems. This paper proposes a resource allocation scheme combined with the simulated annealing algorithm, which minimizes the overall performance loss of the system while keeping the system low delay. Finally, it is verified through experiments that the task scheduling and resource allocation methods proposed in this paper can significantly reduce the response delay of the application.

1. Introduction

The rapid development of technologies such as the Internet of Things has brought mankind into a new era of intelligence. The rapid popularization of high-speed Internet has brought about continuous growth in the amount of websites and data. The massive amount of data has promoted the evolution of the entire computing model and also put forward higher requirements on data storage and processing technology [1]. However, due to the disadvantages of traditional cloud computing technology, such as insufficient bandwidth, real-time performance, and high energy consumption, it has been unable to efficiently process massive amounts of data [2]. Therefore, edge computing emerged as a new computing model, which extends data to the edge of the network on the basis of cloud computing to achieve the overall performance of the Internet of Things technology [3, 4].

Edge computing brings convenient services in the face of complex network environments and diverse application services, but it also brings a series of new problems and challenges, such as the configuration of computing resources and task scheduling. Literature [5] puts forward the theory of edge computing on the basis of cloud computing and transmits data to the edge of the Internet to improve the overall availability and scalability of the system; literature [6]

puts forward the theoretical basis of fog computing for the first time, and the technology is also by configuring computing and storage devices at the edge of the Internet to reduce the amount of Internet data transmission, so as to achieve the purpose of reducing latency and saving bandwidth [7]. Compared with fog computing, edge computing technology pays more attention to the collaboration of resources between edge nodes and can handle data upstream of the cloud or downstream of the Internet of Things very well [8]. Resource allocation and task scheduling optimization in edge computing technology is one of the important research issues of this technology, and its implementation plan directly affects the utilization rate of resources and the service experience of users [9]. Literature [10] integrates optimization problems in edge computing scenarios and sorts out a number of optimization indicators according to the optimization scenarios. For the problem of resource optimization and allocation of edge computing, Brogi et al. sorted out the types of optimization algorithms, optimization goals, and constraints [11]. Aiming at the task scheduling problem, literature [12] specifically studied three task scheduling methods. The first method is concurrent, the second is FCFS (first come, first served), and the third method is allocated according to delay priority. In the first method, the acquired tasks are allocated to edge devices for processing, and there is no need to care about the usage of each device. In the second method, the acquired tasks will be processed in sequence according to the entry order. Only when the computing power of the edge node cannot handle the current task will the task be moved to the cloud for processing. In the delayed priority allocation method, the arriving tasks will be scheduled in the order of priority. When the edge computing resources are not enough, the low priority will be processed by the cloud. Research shows that although the number of tasks that can be executed at the same time is the largest in the first method, the equipment utilization is the highest, but because the resources that can be used for computing are limited, each task causes a large delay. In the second method, Since the order of task execution is carried out in order of priority, some tasks with lower priority will be sent to the cloud for execution, so this method cannot cope with some tasks with higher requirements for delay, and it also brings data transmission. Energy consumption is high. For this reason, literature [13] introduced a knapsack algorithm-based symbiosis search scheduling algorithm based on the above research. This method has significantly improved energy consumption, network utilization, and execution cost compared with the traditional knapsack algorithm and the FCFS method.

In order to make full use of the computing power of the edge server and further reduce the response delay of the application, literature [14] proposed a delay-aware application module management method oriented to the edge environment. Te-Yi et al. [15] divided the delay priority of different tasks and used heuristic algorithms to solve the problem of computing resource allocation, thereby improving the efficiency and quality of edge computing. According to the above-mentioned research findings, task scheduling and resource allocation in edge computing technology have attracted the attention of a large number of scholars, but the research on the two aspects of true comprehensive resource allocation and task scheduling needs to be further deepened. This paper studies the problem of resource allocation and task scheduling for edge computing. First, through the realization of collaborative caching between different edge nodes, each edge node caches differentiated data, so as to train and obtain a submodel with greater difference. To achieve a more accurate edge integration model, second, use cache compression records and record sharing to achieve reasonable data distribution scheduling and caching, and finally design and implement the TCP/IP network node cache module in the edge computing framework.

1.1. Principle Analysis of Collaborative Cache for Edge Computing. In order to improve the performance of edge computing, the process of edge computing is first studied. Integrated diversity, that is, the difference between submodels, is the key issue of integrated learning methods. Through research, it is found that if the same submodels are combined, there will be no performance improvement; if there is a performance improvement after the combination, there must be a difference between the submodels. Tumer et al. [16] analyzed the simple soft voting integration method through decision boundary analysis. In order to keep it simple, assuming that all submodels have the same error rate, the θ term is introduced to describe the relationship between the different submodels, and the expected cumulative error after integration is shown in the following formula:

$$\overline{err}(H) = \frac{1+\theta(n-1)}{n} err_i(h_i), i = 1, 2, \ldots, n. \quad (1)$$

In the above formula, $err_i(h_i)$ is the expected error rate of the submodel and n is the size of the integration scale. It can be seen from the formula that if the submodels are independent of each other, namely, $\theta = 0$, the ensemble learning error will be reduced by n times. If each submodel is associated with all other submodels, namely, $\theta = 1$, the performance of the integrated submodel will not be effectively improved. This analysis clearly reveals the importance of different submodels in ensemble learning, and the same conclusion is also applicable to other ensemble methods [17].

However, it is not easy to generate highly diverse submodels [18]. The biggest obstacle is that the submodels are obtained on the same task and the same training set, so there is often a high correlation between the submodels. Many theoretically feasible methods, such as the optimal solution of the weighted average method, are difficult to work in reality, and the situation may even be worse. In fact, the performance of the submodels should not be too bad; otherwise, the combined performance not only will not be improved but also will be reduced, which makes it more challenging to generate diverse submodels. If the performance of the submodel is poor, the cumulative errors after simple soft voting integration will continue to increase, and other integration methods have similar results [19, 20].

If the submodels are independent of each other, the error of ensemble learning will be reduced. If each submodel is related to other submodels, the error of ensemble learning will become larger. This analysis clearly reveals the importance of the different submodels. In this case, it is necessary to implement collaborative caching between different edge nodes, and each edge node caches differentiated data, so as to train a submodel with larger differences and realize a more accurate edge integration model.

1.2. Cache Compression Records and Record Sharing. The compressed record of the cached data plays an important role in the intelligent scheduling of the cache. The efficient compression recording method can record the data information cached by each edge node and realize the exchange and collection of cache information between edge nodes. The cache intelligent scheduling scheme can reasonably schedule the cache according to the data distribution and supports the distributed submodel learning and final integrated learning of each edge node, as shown in Figure 1. In this section, we mainly introduce two parts: cache data record and record sharing.

1.2.1. Cache Data on Edge Computing Nodes and Record Efficiently. The data required for the training of the integrated learning model is collected from the neighboring user terminal equipment and transmitted to the edge node, and the edge node is cached in the LRU mode and is efficiently recorded using the combinable counting bloom filter (CCBF). The specific process is as follows: when the data arrives at a certain edge node, whether the data has been cached by querying the CCBF is judged. If it has been cached, the data is not cached; if it has not been cached, the data is cached using LRU and used CCBF performs high-efficiency compression recording.

The specific operation is as follows: use k hash functions to hash the data received by the edge node into k bit arrays and check whether the corresponding unit of orBarr in CCBF is 1; if it is 1, it means that the data has been cached, so do not proceed. If it is not 1, LRU cache is required. The implementation of LRU adopts the form of a linked list. When caching, it is necessary to determine whether the cache capacity of the edge node is reached. If the cache capacity of the edge node is not reached, the data is cached at the head of the linked list; if the cache capacity has been reached, it is the oldest. The used data is eliminated, and, through the pseudorandom number generator, the bit array corresponding to the unit of each hash function operation in the last insertion operation of the data is cleared to zero, the orBarr array is updated, and the cache record is cleared. Then add the new data to the head of the cache linked list. After adding the data to the cache, use the pseudorandom number generator to correspond to the bit array whose location unit (unit with subscript $Hash_j(d)$) has been set to 1 according to the hash result. Randomly select a bit array $barr_i$ (the i-th bit array of CCBF) in the g bit array, set its subscript as $Hash_j(d)$ to 1, update the OrBarr array, and complete the update of CCBF.

1.2.2. Exchange and Merge Compressed Records of Cached Data with Neighbor Nodes. The compressed representation of cached data (CCBF) is exchanged and merged with neighbors within a certain range in order to obtain a global view of the edge node cached data, and the subsequent cache scheduling process can be guided based on this global view. Once a node receives a compressed record of the neighbor node cache data from the interface, the compressed record will be stored with the name $CCBF_i$, where i is the ID number of the corresponding edge node interface.

Use the ID number of the edge network node to exchange $CCBF_i$, hash the received data into k bit array units through k hash functions, and then query whether the orBarr unit corresponding to the neighbor node is 1; if it is 1, it means that the data has been existing in the cache data of the neighbor node, it is necessary to delete the redundant cache data, set the unit corresponding to the middle bit array of the node to 0, and update $CCBF_i$ the node.

The original compression record of the edge node is merged with the received neighbor nodes. The specific operation is to first determine whether the amount of cached data represented by the merged compression has exceeded the capacity n of CCBF, and then, according to the different bit arrays label, merge each bit array in order, and update the orBarr array to get a global view of the edge network cache data. After merging, a global view of the data compression records cached in the neighbor nodes can be obtained, and this view will be used to guide the neighbor nodes to cache various data subsequently received.

1.3. Differentiated Adaptive Collaborative Caching. The cache intelligent scheduling scheme can reasonably schedule the cache according to the data distribution and supports the distributed submodel learning and final integrated learning of each edge node. In this section, we mainly introduce the differentiated adaptive collaborative caching method for model learning.

1.3.1. Cache Different Data between Neighbor Nodes. When an edge node requests to cache some data, it needs to determine whether the data already exists in the cache of the node and its neighbor nodes according to the global view of

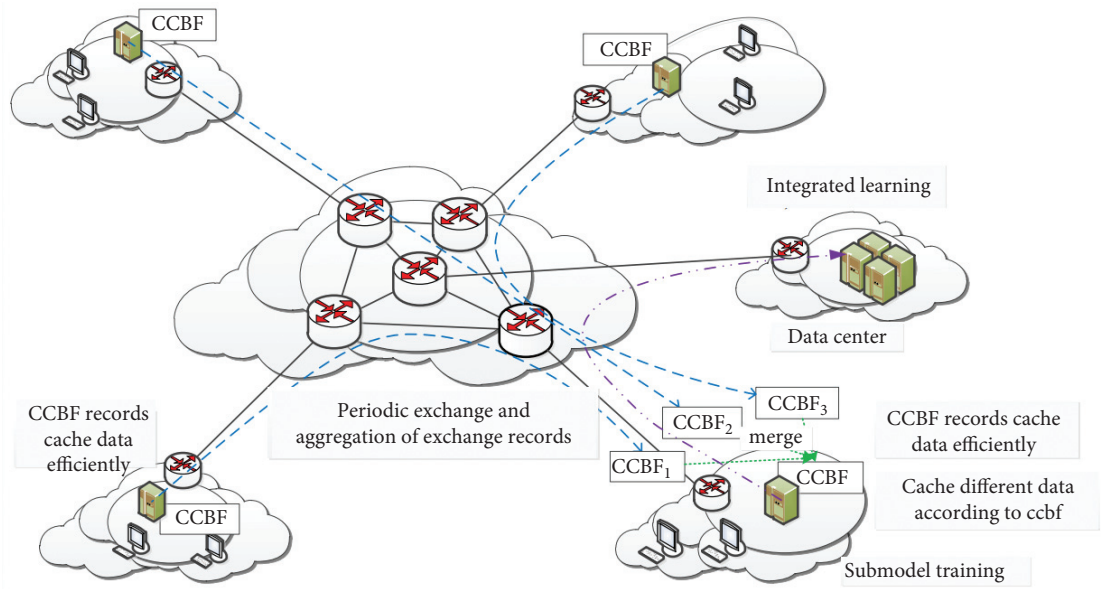


Figure 1: Collaborative caching scheduling method.

the cached data. The specific operation is as follows: query, which represents a global view of data cached in neighbor nodes. Hash the cached data requested by the node k times to obtain k hash results, and check whether the corresponding array orBarr unit is 1; if the corresponding unit in the orBarr array is 1, it means that the data has been cached at this edge. On the network node, the data is ignored, no caching operation is performed, and the following data processing is performed; if the corresponding unit of the array in is 0, it means that there is no compressed record of the data in the cache, indicating the cache of other neighboring nodes. If the data is not included, the data is added to the cache of the edge node, and the compressed record of the data is added to the corresponding node. Through the above operations, it can be ensured that different data can be cached on neighboring nodes for training different submodels, and, at the same time, communication overhead can be reduced through collaborative caching.

1.3.2. Distributed Training of Submodels. The data cached on a node is used to train the local submodel. When the local data is not enough to make the submodel converge, it is necessary to expand the scope of collaboration by requesting differentiated data from other edge nodes. By performing merging of orBarr in $CCBF_i$ of different neighboring nodes' cached data, the obtained cached data records of different neighboring nodes are compared with the cached data records of the local node to obtain the required data compression record $CCBF_i$ and send it to the corresponding edge node. When the corresponding edge node receives the request, it queries the cache of the local node according to $\overline{orBarr}$ and returns the differentiated data to the requesting node. After the requesting node receives the data, it caches the data and updates $CCBF_i$ and $CCBF$ and then inputs the data into the submodel for training. Repeat these processes until the submodel converges.

1.3.3. Integrated Learning. In order to reduce network data transmission traffic and ensure data privacy and security, the training results of the distributed submodels are uploaded to the data center, and the integrated results are obtained by assigning different weights to the output results of each submodel in the data center. The set output result $H(x)$ is shown in the following formula:

$$H(x) = \sum_{i=1}^{n} \omega_i h_i\ (x). \tag{2}$$

In the above formula, ω_i represents the weight of h, usually with the constraints of $\omega_i \geq 0$ and $\sum_{i=1}^{n} \omega_i = 1$.

The weights of these parameters in the submodel are uploaded to the central node, and the central node performs integrated learning. Specifically, for n sub-$h_1, \ldots, h_n$ models, the following methods are used for ensemble learning: $p(x)$ is the distribution of the input, $\epsilon_i(x)$ is the error term, and $C_{ij} = \int (h_i(x) - f(x))(h_j(x) - f(x))p(x)d(x)$.

The optimal weight can be solved by the following formula:

$$\omega = \underset{\omega}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j C_{ij}. \tag{3}$$

By Lagrangian multiplier method, ω_i is obtained as shown in the following formula:

$$\omega_i = \frac{\sum_{j=1}^{n} C_{ij}^{-1}}{\sum_{k=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j C_{kj}^{-1}}. \tag{4}$$

1.4. Design and Implementation of Node Cache Module in TCP/IP Edge Network. The TCP/IP network node cache module is designed and implemented in the edge integrated learning framework. First, the implementation of the LRU cache module of TCP/IP network nodes is introduced, and then the method of deploying the LRU cache module to the NS-3 edge simulation platform is introduced.

LRU is the abbreviation of "Least Recently Used." It is a commonly used cache replacement algorithm. The cache data that has not been used the most recently is selected to be eliminated. The LRU cache in NS-3 mainly implements the function of caching data on each edge network node. In this section, the LRU cache on the TCP/IP edge network node is designed and implemented.

First the implementation of LRU cache is introduced, followed by the way to implement LRU cache on the NS-3 platform. The specific operations are as follows:

(1) The first step is to construct the code of the LRU cache, where the cache size needs to be set, so the function of LRU cache is used, and the capacity of the cache is set as a parameter of this function. The LRU cache is based on the encapsulated data packet as a unit for caching. The LRU cache mechanism uses the form of a linked list, placing the least recently used at the head of the linked list. It mainly includes three functions, namely, the addition, deletion, and search functions of cached data. The specific operations are as follows:

(1) To increase the cache data (Memory), it is necessary to first determine whether the current cache has reached the capacity of the cache. If the cache capacity has been reached, delete the last one of the linked list, and then store it; if the cache capacity is not reached, store it directly at the head of the cache.

(2) In the process of deleting the cached data (Remove), when the cache capacity is not enough, the last data in the linked list represents the most recently unused data, and it is deleted.

(3) In the process of looking up cached data (Lookup), if the data is not found, -1 is returned; if it is found, the value of the data is returned, and then the data is placed at the head of the cache.

(2) Implement LRU cache on the NS-3 platform.

To implement LRU caching on the NS-3 platform, you need to add a custom LRU caching module to the original module of NS-3.

(1) First, the basic structure of NS-3 is introduced here. src is the source code directory of NS-3, and the directory structure basically corresponds to the compiled module. Each file in the src directory basically corresponds to a module, and the structure of all modules in it is basically the same.

(2) Then, according to the design of NS-3, the implementation of LRU cache is added to NS-3 as a custom module.

1.5. Edge Network Node Deployment. This paper deploys the edge network tree topology as shown in Figure 2. As shown in the figure, it includes a remote data center, a gateway node, four edge computing nodes, and eight terminal devices, which are connected through a gigabit link, and the data transmission between them is point-to-point transmission. Peer-to-peer technology (P2P), also known as peer-to-peer network technology, is a new network technology that relies on the computing power and bandwidth of participants in the network, instead of concentrating all the dependencies on a few servers in the edge network topology. The cache size of each edge computing node is 2000 KB. Each edge computing node can cache data, efficiently record cached data, and perform computing tasks for cached data.

The data required for neural network model training are all released by the terminal equipment node. The terminal device generates the learning data of the model and sends the data to the edge computing node. After the edge computing node receives the data, it first performs data caching and efficient recording and then uses the different data in the collaborative cache to train the submodel and finally sends the training results of the submodel to the data center for integrated learning. Background traffic data is released by remote data center nodes. The data center generates background traffic data and sends the data to edge computing nodes. After the edge computing node receives the data, it caches the data and sends the background traffic data to the terminal device.

1.6. Construction of Edge Network Node Learning Module. In this article, the edge network node learning module is designed and implemented. When deploying a neural network model on the NS-3 edge simulation platform, the difficulty encountered is the joint compilation of the NS-3 platform and OpenNN. So, the method of joint compilation of NS-3 platform and OpenNN is introduced first, and then the process of edge integration learning based on OpenNN design pattern is introduced. The operation steps applied on the simulation platform of NS-3 are shown in Figure 3.

Because the NS-3 simulation platform is compiled with /waf, in the process of compiling the OpenNN neural network library, the newly added library needs to be included in the wscript file, and the corresponding library file needs to be included in the script. The specific operations are as follows:

(1) Put the Eigen folder in OpenNN under the ndnSIM/ns-3 folder, which is the preliminary step of the joint compilation of OpenNN and NS-3. Eigen is a C++ template library for linear operations, supporting matrix and vector operations, numerical analysis, and related algorithms. Because OpenNN contains a lot of matrix operations, you need to use the Eigen library.

(2) Add in the corresponding position in the wscript file under the NS-3 folder:

Def build (bld):

Bld.stlib ("opennn")

Module.uselib = 'opennn.'

Module.source = 'opennn/∗∗/∗.cpp.'

Module.full_headers = 'opennn/∗∗/∗.cpp.'

Such an operation is to add the OpenNN neural network library to the wscript file and include the corresponding header files and source files.

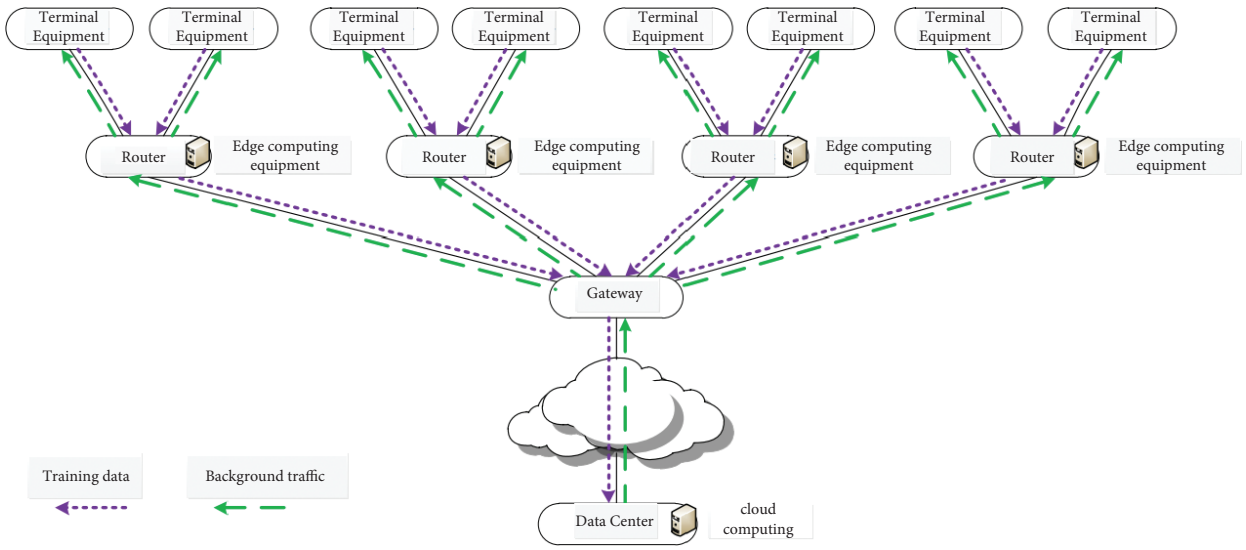


Figure 2: Edge network topology.

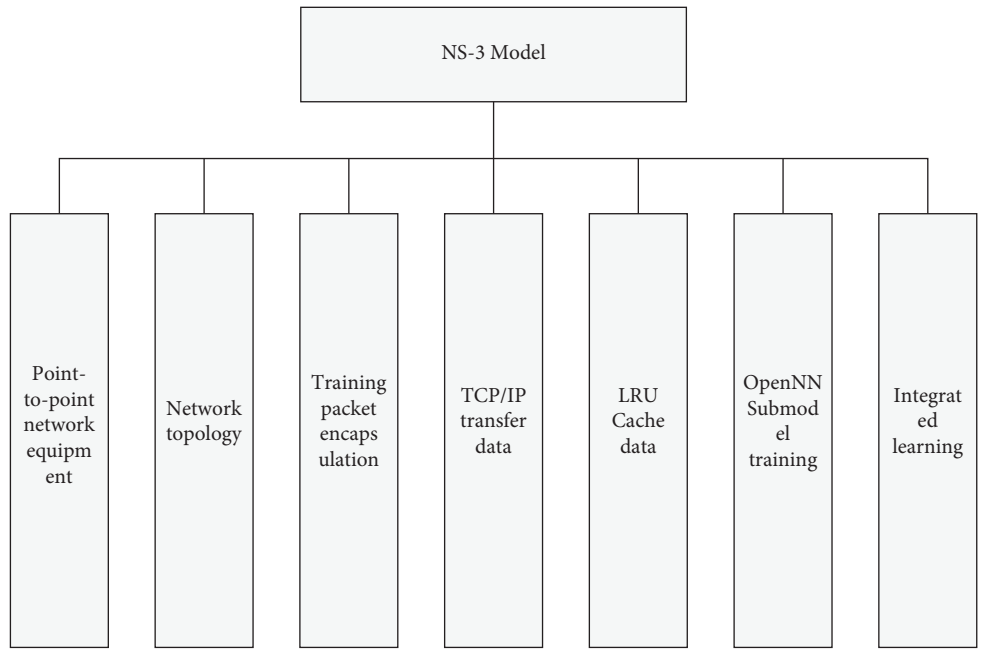


Figure 3: Edge network topology.

(3) Add the following to the header file in the code that needs neural network model training:

#include "../opennn/opennn.h."

Because the opennn.h file contains the header files required by OpenNN, only including this one header file in the source file can include all the required header files.

(4) NS-3 contains a fixed Vector vector. When using the Vector vector defined by the Eigen library in OpenNN, it needs to be distinguished. The method adopted is as follows:

Using namespace OpenNN:

When using the OpenNN model, first include the OpenNN namespace, and then when using the OpenNN Vector vector, add "OpenNN:" in front, that is, OpenNN: : Vector *****. In this way, the Vector vector in NS-3 can be distinguished from the vector in OpenNN.

The ensemble learning process is closely related to different submodels. In the edge integrated learning scenario, different edge nodes often deploy similar models, build submodels by learning the data around the edge nodes, and finally distribute the submodels on different nodes through the central node to form an integrated model. In this case, it is necessary to provide different data for different edge nodes, so as to obtain the training results of different

submodels, so as to achieve a more accurate integrated model.

In this article, the integrated learning method based on the OpenNN design pattern obtains the results by assigning different weights to the output results of each submodel. The set output result is $H(x)$, where $H(x) = \sum_{i=1}^{n} \omega_i h_i(x)$ represents the weight of ω_i, usually with $\omega_i \geq 0$ and $\sum_{i=1}^{n} \omega_i = 1$ constraints.

The weights of these parameters in the submodel are uploaded to the central node, and the central node performs integrated learning. Specifically, for n submodels $h_1, \ldots, h_n$, the following methods are used for ensemble learning.

Assume that the output of each submodel can be written as a true value plus an error term, as shown in the following formula:

$$h_i(x) = f(x) + \epsilon_i(x), i = 1, \ldots, n. \tag{5}$$

The integration error can be expressed as in the following formula:

$$\begin{aligned} \widehat{err}(H) &= \int \left(\sum_{i=1}^{n} \omega_i h_i(x) - f(x) \right)^2 p(x) d(x) \\ &= \int \left(\sum_{i=1}^{n} \omega_i h_i(x) - f(x) \right) \times \left(\sum_{j=1}^{n} \omega_j h_j(x) - f(x) \right) p(x) d(x) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j C_{ij} \int \left(h_i(x) - f(x)\right)\left(h_j(x) - f(x)\right) p(x) d(x). \end{aligned} \tag{6}$$

In the above formula, $p(x)$ is the distribution of the input and $\epsilon_i(x)$ is the error term. The optimal weight of $C_{ij} = \int (h_i(x) - f(x))(h_j(x) - f(x)) p(x) d(x)$ can be solved by the following formula:

$$C_{ij} = \int \left(h_i(x) - f(x)\right)\left(h_j(x) - f(x)\right) p(x) d(x). \tag{7}$$

By Lagrangian multiplier method, D is obtained by the following formula:

$$\omega_i = \frac{\sum_{j=1}^{n} C_{ij}^{-1}}{\sum_{k=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j C_{kj}^{-1}}. \tag{8}$$

2. Experiment

2.1. Lab Environment. The performance of the adaptive collaborative caching scheme was evaluated on the NS-3 platform. The NS-3 platform is a modular, programmable, extensible, open, open-source, and community-supported computer network simulation framework. Connect the neural network library OpenNN (Open Neural Network Library) to NS-3 for experimental simulation. OpenNN is an open-source neural network library for the construction of neural networks. It has a wide range of applications, including functional regression, pattern recognition, time series forecasting, optimal control, optimal shape design, or inverse problems. In this article, all simulation experiments are performed on a local machine. The configuration and environment of the experiment host are as follows:

(1) CPU: Intel Core i7, 3.4 G CPU;

(2) Installed memory (RAM): 16 GB;

(3) Linux operating system: Ubuntu 16.04;

(4) Kernel version: 3.19.

2.2. Dataset. In order to evaluate the performance of the collaborative caching scheme, this paper uses four datasets for learning. Specifically, two text datasets (D1 and D2) are used to train the MLP model. In order to train the VGG model, a tigerface image dataset (D3) and a human face dataset (D4) are applied.

(1) Covertype dataset (D1): this dataset includes the forest vegetation types of Roosevelt National Forest. There are 4 types of soil, corresponding to 7 types of vegetation. The 581,012 data-item forest vegetation is divided into four soil types. The number of data items for different soil types is uneven. The number of type 4 is less than 3,000, and the number of type 5 is close to 10,000. The quantity of any other type is greater than 10,000.

(2) Healthy elderly dataset (D2): the sequential exercise data of 14 healthy elderly aged 66 to 86 years who used sensors to identify clinical environmental activities. Participants were assigned to two clinical room environments (S1 and S2). S1 (Clinical Room (1)) and S2 (Clinical Room (2)) are equipped with different sensor receiving numbers and positions. The number of data items is 75128, which is divided into 6 different behaviors on average.

(3) Reid-tigerface dataset (D3): Atrw Reid-tigerface image captured. After the picture is edited, the image resolution is adjusted to 128×128. There are 500 tigers in total, each of which has 10 photos. According to the active region (Russia Far East and Northern India), the dataset is divided into two scenarios.

(4) Casia-face dataset (D4): obtain face images of human faces. After the picture is edited, the image resolution is adjusted to 128×128. There are 500 people in total,

and each of them has 10 facial photos. According to the shooting angle (front position and side 45°), the dataset is divided into two scenes.

2.3. Validation Model. This paper implements the two following learning models: the multilayer perceptron (MLP) model is used to train two text datasets, and the Visual Geometry Group (VGG) network model is used to train two image datasets. The model is introduced in detail.

2.3.1. Multilayer Perceptron (MLP) Model. MLP is a feedforward artificial neural network that can map multiple input data to output data. Each layer of MLP is a fully connected layer. This paper implements a six-layer MLP model, including an input layer, four hidden layers, and an output layer. The following describes the internal structure of the multilayer perceptron.

Neurons can be combined into a neural network. The structure of a neural network refers to the number, arrangement, and connectivity of neurons. Any kind of network structure can be represented by a directed label graph, where nodes represent neurons, and edges represent connections between neurons. The edge labels represent the parameters of the neuron and indicate the inflow of the neuron. Most neural networks, even biological neural networks, present a hierarchical structure. In this case, the working layer is the basis for determining the structure of the neural network. Therefore, a neural network usually consists of a set of perception nodes that constitute the input layer, one or more hidden layers of neurons, and a set of neurons that constitute the output layer. As mentioned above, the characteristic neuron model of the multilayer perceptron is the perceptron. On the other hand, the multilayer perceptron has a feedforward network structure. The feedforward structure does not contain cycles; that is, the structure of the feedforward neural network can be expressed as an acyclic graph. Therefore, the neurons in the feedforward neural network are divided into a series of layer $h+1$ neurons $L^{(1)}, \ldots, L^{(h)}, L^{(h+1)}$, so that the neurons in any layer are only connected to the neurons in the next layer. The input layer is composed of n external inputs, not a neuron layer; the hidden layer $L^{(1)}, \ldots, L^{(h)}$, respectively, contains a hidden neuron in $s^{(1)}, \ldots, s^{(h)}$; the output layer $L^{(h+1)}$ is composed of m output neurons. Figure 4 shows the network structure of the multilayer perceptron. There are n inputs, h hidden layers, $s^{(i)}$ neurons, and $i = 1, \ldots, h$ and neurons are in the output layer. In this chapter, the superscript is used to identify the layer.

The multilayer perceptron neural network can be regarded as a parameterized function space V from input $X \subset \mathbb{R}^n$ to output $Y \subset \mathbb{R}^m$. The element form of V is y: $X \longrightarrow Y$. They are parameterized by neural parameters, which can be combined in a d-dimensional vector $\zeta = (\zeta_1), \ldots, (\zeta_d)$. Therefore, the dimension of the function space V is d.

For the first hidden layer L (1), by formula (9), the combined function is obtained by adding the dot product of the weight and the input to the deviation, thereby obtaining

$$c^{(1)} = b^{(1)} + w^{(1)} \cdot x. \tag{9}$$

According to formula (10), the output of this layer $a^{(1)}$ is obtained by the combination of conversion and activation function:

$$y^{(1)} = a^{(1)}\left(c^{(1)}\right). \tag{10}$$

Similarly, for the last hidden layer, the combined function is given by the following formula:

$$c^{(h)} = b^{(h)} + w^{(h)} \cdot y^{(h-1)}. \tag{11}$$

The output of this layer is found by formula (12) by using the activation function:

$$y^{(h)} = a^{(h)}\left(c^{(h)}\right). \tag{12}$$

The output of the neural network is obtained by transforming the output of the last hidden layer by the neurons in the output layer F. Therefore, the combined form of the output layer is shown in the following formula:

$$c^{(h+1)} = b^{(h+1)} + w^{(h+1)} \cdot y^{(h)}. \tag{13}$$

The output of the output layer is transformed by formula (14) through the combination of the layer and activation into

$$y^{(h+1)} = a^{(h+1)}\left(c^{(h+1)}\right). \tag{14}$$

Combining the above equations, an explicit expression of the multilayer perceptron function is obtained in the following form:

$$y = a^{(h+1)}\left(b^{(h+1)} + w^{(h+1)} \cdot a^{(h)}\left(b^{(h)} + w^{(h)} \cdot a^{(h-1)}\left(\ldots a^{(1)}\left(b^{(1)} + w^{(1)} \cdot x\right)\right)\right)\right). \tag{15}$$

In this way, the multilayer perceptron function is represented by formula (16) as the composition of the layer output function:

$$y = y^{(h+1)} \circ y^{(h)} \circ \ldots \circ y^{(1)}. \tag{16}$$

Multilayer perceptron can be regarded as a function of multiple variables formed by the superposition and addition of functions of one variable. Different activation functions produce different function families, and multilayer perceptrons can define these function families. Similarly, different neural parameter sets cause different elements in the function space defined by a particular multilayer perceptron.

2.3.2. Visual Group Network (VGG) Model. VGG is a deep convolutional neural network for computer vision. The implementation of this paper includes 5 convolutional

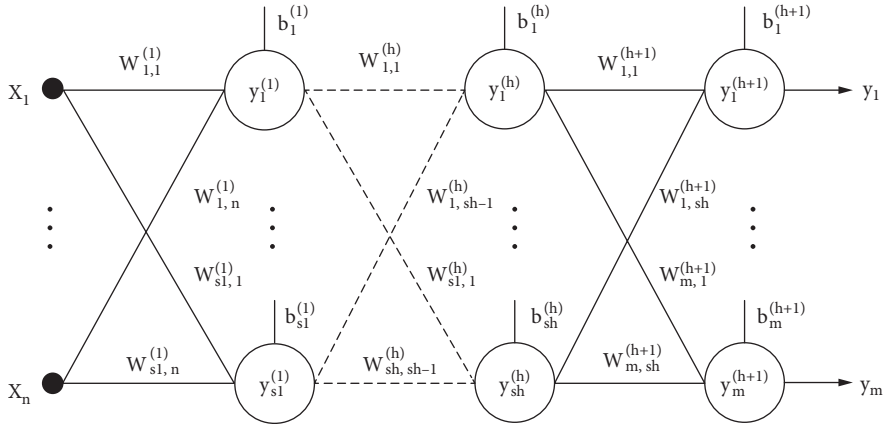


Figure 4: Multilayer perceptron.

blocks, each of which consists of 2–4 convolutional layers. At the same time, a maximum pooling layer is connected to the end of each block to reduce the size of the picture. The number of convolution kernels in each block is the same, and the number of convolution kernels in the later block is larger. For five convolution blocks, each layer contains 64-128-256-512 convolution kernels, and, in this article, 10 convolutional layers are used and 4 pooling layers are alternately performed, and the specific arrangement is shown in Figure 5. Among them, the convolution layer uses a 3∗3 convolution kernel, the activation function uses the ReLU activation function, that is, $F(x) = \max(0, x)$, and the training algorithm uses the Adam algorithm that can adaptively adjust the learning rate. The following describes the Adam optimization algorithm.

The Adam optimization algorithm is an extension of the stochastic gradient descent algorithm. Recently, it is widely used in deep learning applications, especially tasks such as computer vision and natural language processing. Adam is different from the classic stochastic gradient descent method. Stochastic gradient descent maintains a single learning rate (called α) for all weight updates, and the learning rate does not change during the training process. The Adam optimization algorithm maintains a learning rate for each network weight (parameter) and adjusts it individually as the learning expands. This method calculates the adaptive learning rate of different parameters from the budget of the first and second moments of the gradient. The following describes the Adam parameter configuration:

α: it is called the learning rate or step size. It controls the weight update rate (such as 0.001). A larger value (such as 0.3) will have faster initial learning before the learning rate is updated, while a smaller value (such as 1.0E-5) will make the training converge to better performance.

β_1: it is the exponential decay rate of the first moment estimation (such as 0.9).

β_2: it is the exponential decay rate of the second moment estimation (such as 0.999). This hyperparameter should be set to a number close to 1 in sparse gradients (such as in NLP or computer vision tasks).

ε: this parameter is a very small number, which is to prevent division by zero in implementation (such as 10E-8).

2.4. Classification Accuracy of Neural Network Model. Table 1 describes the classification accuracy of the MLP and VGG models trained on different schemes. For different training models and datasets, the collaborative caching scheme and the centralized scheme both achieve similar high performance in accuracy. This is because the collaborative caching solution can provide more valuable training data to support model training, while the centralized solution can collect all training data to support model training. On the contrary, the solution of periodically requesting cached data cannot provide enough training data in a short time, which affects the training of the edge nodes by the submodel, thereby reducing the performance of the integrated result.

2.5. Training Delay of Neural Network Model. Figure 6 describes the learning delay of different models under the three schemes. It can be seen from the figure that both MLP and VGG can use cooperative caching to achieve rapid convergence. There is a maximum difference of 7000 seconds in the learning delay between the periodic request cached data scheme and the collaborative cache scheme. Within one to two hours, the collaborative caching solution provided enough cached data items for the submodel learning and integration process. Since the centralized solution collects all training data to support model training, the centralized model learning delay is less than that of the solution that periodically requests cached data. On the other hand, the centralized transmission delay is large, which also reduces the efficiency of centralized model learning.

2.6. Network Data Transmission Traffic Load. The network data transmission traffic load is shown in Figure 7. It can be seen from the figure that regardless of the model or dataset, the network data transmission traffic load of the collaborative caching scheme is always the smallest, and more

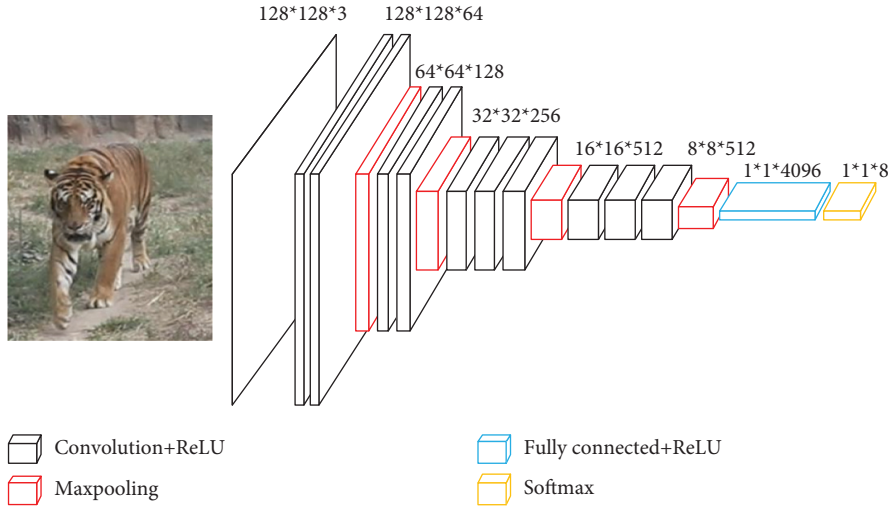


Figure 5: VGG model.

Table 1: Classification accuracy of neural network model.

Method	MLP		VGG	
	D1	D2	D3	D4
Centralized	0.848	0.968	0.917	0.923
Periodically requested cache	0.789	0.947	0.827	0.852
Cooperative caching	0.847	0.968	0.917	0.923

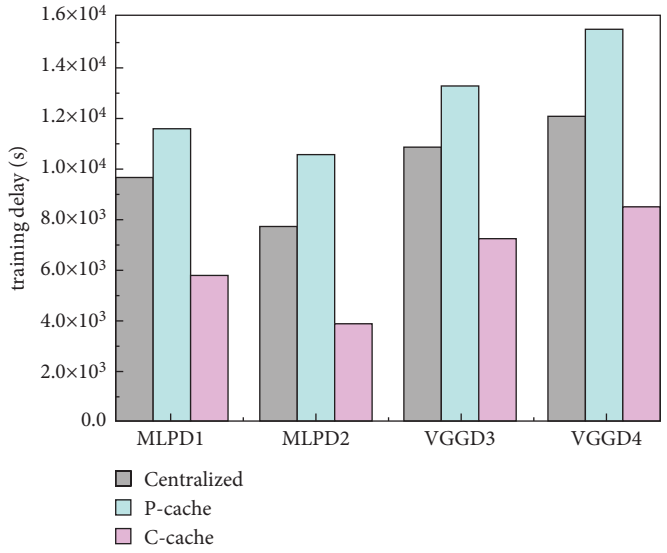


Figure 6: Training latency of neural network model.

powerful models such as VGG will consume more communication resources. The network data transmission traffic load of the centralized solution is twice that of the collaborative cache solution. Because all learning data needs to be sent to the data center, the network data transmission traffic load of the centralized solution is the largest. In addition, data request and cooperative caching are beneficial to the cooperative caching scheme in terms of transmission overhead. More valuable data is cached on edge nodes, thereby reducing redundant data transmission between different edge nodes and reducing the transmission traffic load in the network.

2.7. Cache Hit Rate. Since the centralized scheme trains the model in the data center and does not cache the data at the edge nodes, we only compare the cache hit rates of Centralized, P-cache, and the proposed C-cache. The local learning hit rate is shown in Figure 8. The overall learning hit rate is shown in Figure 9. The local learning hit rate of

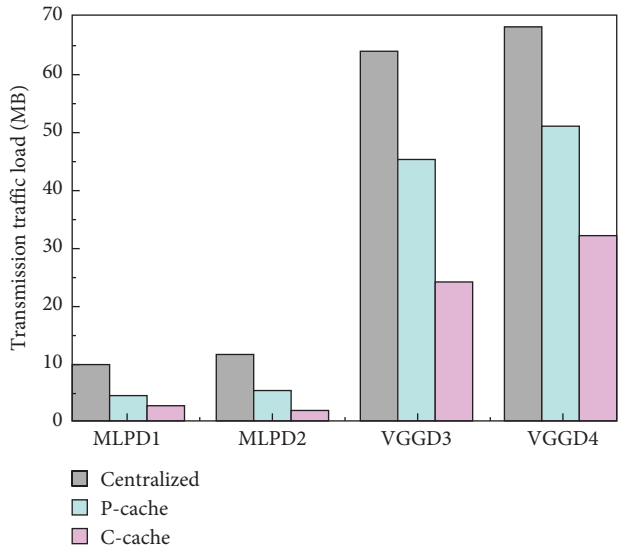


Figure 7: Network data transmission overhead.

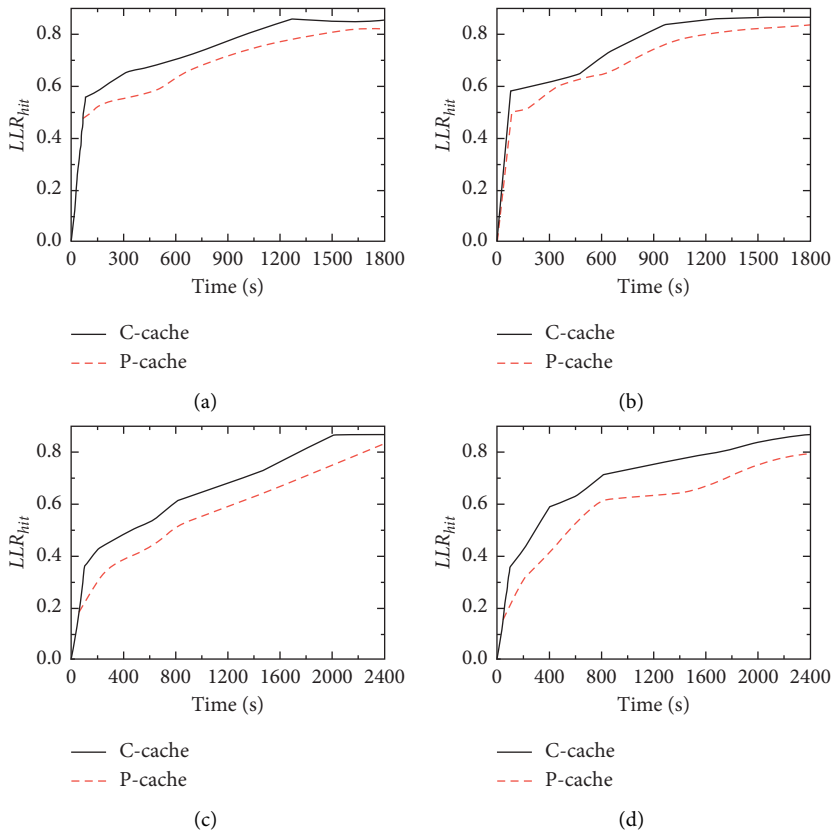


Figure 8: Local learning hit ratio. (a) LLR_{hit} during training MLP on D1. (b) LLR_{hit} during training MLP on D2. (c) LLR_{hit} during training VGG on D3. (d) LLR_{hit} during training VGG on D4.

C-cache and P-cache is increased to the maximum stable values of 0.87 and 0.85, respectively. The global learning hit rate of C-cache and P-cache is increased to the maximum stable values of 0.83 and 0.81, respectively, and the learning data is generated and cached at different edge nodes.

Figure 10 depicts the hit rate of background traffic data. The cache hit rate of background traffic data first increases with the passage of time. When the learning data increases, more background traffic data is switched from the cache of edge computing nodes. Therefore, the cache hit rates of

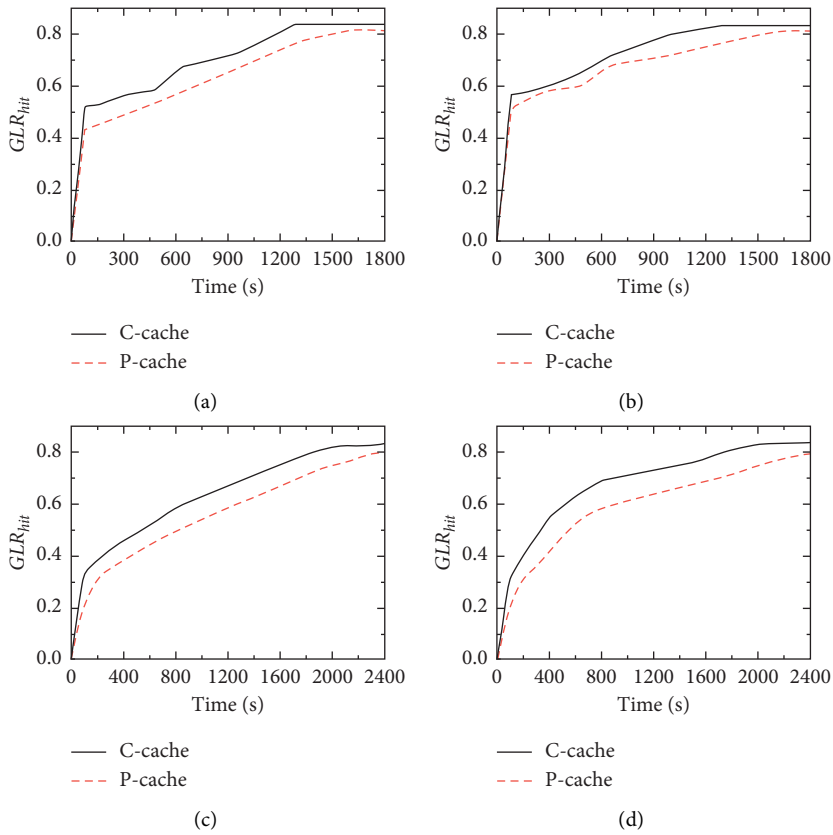


Figure 9: Global learning hit ratio. (a) GLR_{hit} during training MLP on D1. (b) GLR_{hit} during training MLP on D2. (c) GLR_{hit} during training VGG on D3. (d) GLR_{hit} during training VGG on D4.

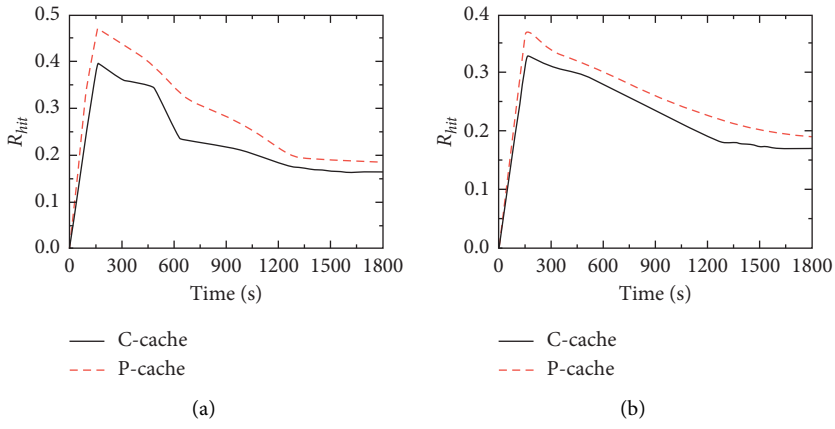


Figure 10: Continued.

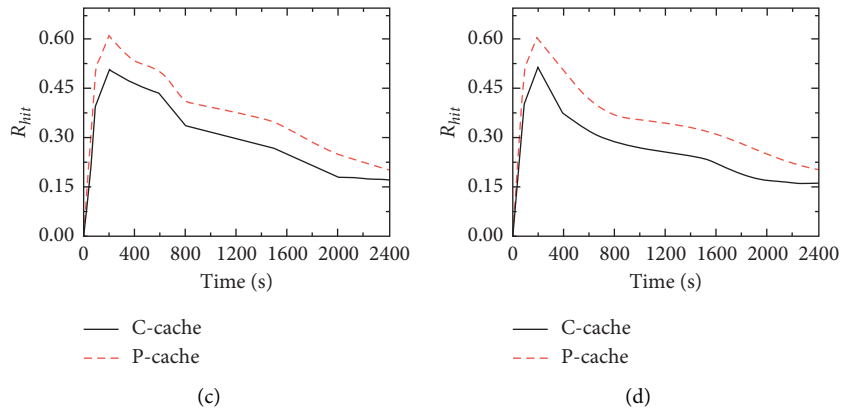


Figure 10: Background traffic data hit ratio. (a) R_{hit} during training MLP on D1. (b) R_{hit} during training MLP on D2. (c) R_{hit} during training VGG on D3. (d) R_{hit} during training VGG on D4.

C-cache and P-cache middle and background traffic data are reduced to 0.17 and 0.19, respectively. For different training models and datasets, the cache hit rate under C-cache decreases faster than that under P-cache. This is because C-cache can use learning data better than P-cache and reserves less available cache space for caching background traffic data.

3. Conclusion

As a complementary extension of cloud computing technology, mobile edge computing will reduce the computing power of the previous cloud to the edge nodes of the network. Through the cooperation between computing nodes, the number of nodes can be calculated, the type can be more comprehensive, and the calculation range can be larger. The emergence of mobile edge computing makes up for the shortcomings of cloud computing technology. Aiming at the problem of network edge cache computing and intelligent scheduling of resource allocation, in order to reduce network traffic load and delay, a simulation framework is finally established within the framework of effective recording cache data collaboration and edge computing learning framework to verify the network collaborative cache solution proposed in this paper. A large number of simulation results show that the collaborative caching scheme proposed in edge network can significantly reduce the learning delay and transmission cost of ensemble learning. In future studies, more datasets and more complex integrated learning models can be used to further improve the experiment. The edge integrated learning framework designed in this paper can be further optimized.

Data Availability

No data were used to support this study.

Conflicts of Interest

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

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] Setting the standard[Z]. https://www.itu.int/en/ITU-T/wtsa16/Documents AVTSASna pshotReport. pdf.

[3] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for IoV in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[4] M. Chiang and T. Zhang, "Fog and IoT: an overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[5] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He, and X. Zhou, "TripRes," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 2, pp. 1–21, 2021.

[6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proceedings of the 1st Edition MCC Workshop Mobile Cloud Comput*, pp. 13–16, NY, USA, 2012.

[7] W. Shi, J. Cao, Q. Zhang et al., "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[8] X. Xu, D. Zhu, X. Yang, S. Wang, L. Qi, and W. Dou, "Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–17, 2021.

[9] W. Wen, C. Xu, F. Yan et al., "Terngrad: ternary gradients to reduce communication in distributed deep learning," in *Proceedings of the Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1509–1519, Long Beach California, USA, December 2017.

[10] H. Hussain, S. U. R. Malik, and A. Hameed, "A survey on resource allocation in high performance distributed computing systems," *Parallel Computing*, vol. 39, no. 11, pp. 709–736, 2013.

[11] J. Bellendorf and Z. Á Mann, "Classification of optimization problems in fog computing," *Future Generation Computer Systems*, vol. 107, no. 1, pp. 158–176, 2020.

[12] A. Brogi, S. Forti, C. Guerrero, and I. Lera, "How to place your apps in the fog: state of the art and open challenges," *Software: Practice and Experience*, vol. 1, no. 1, pp. 1–8, 2019.

[13] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, 2017.

[14] D. Rahbari and M. Nickray, "Scheduling of fog networks with optimized knapsack by symbiotic organisms search," in *Proceedings of the 2017 21st Conference of Open Innovations Association (FRUCT)*, pp. 278–283, Helsinki, Finland, November 2017.

[15] T. Zhang, J. Deng, and J. Wang, "Progressive damage analysis (PDA) of carbon fiber plates with out-of-plane fold under pressure," *Computer Modeling in Engineering and Sciences*, vol. 124, no. 2, pp. 545–559, 2020.

[16] M. Redowan, R. Kotagiri, and B. Rajkumar, "Latency-aware application module management for fog computing environments," *ACM Transactions on Internet Technology*, vol. 19, no. 1, pp. 1–21, 2018.

[17] T. Y. Kan, Y. Chiang, and H. Y. Wei, "Task offloading and resource allocation in mobile-edge computing system," in *Proceedings of the 2018 27th Wireless and Optical Communication*, pp. 1–4, Hualien, Taiwan, May 2018.

[18] T. Zheng, Y. Chang, and S. Zhang, "Quantum risk assessment model based on two three-qubit GHZ states," *Computer Modeling in Engineering and Sciences*, vol. 124, no. 2, pp. 573–584, 2020.

[19] T. Kagan and J. Ghosh, "Theoretical foundations of linear and order statistics combiners for neural pattern classifiers," *IEEE Transactions on Neural Networks*, vol. 7, pp. 1–35, 1996.

[20] Y. Qin, D. Wu, Z. Xu, J. Tian, and Y. Zhang, "Adaptive in-network collaborative caching for enhanced ensemble deep learning at edge," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–14, Article ID 9285802, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6974757, 9 pages
https://doi.org/10.1155/2021/6974757

Research Article

Multicamera Calibration Optimization Method Based on Improved Seagull Algorithm

Shuai Du, Jianyu Wang, and Jia Guo

School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China

Correspondence should be addressed to Shuai Du; dushuai@njust.edu.cn

Received 12 August 2021; Accepted 1 December 2021; Published 21 December 2021

Academic Editor: Xuyun Zhang

There are some problems in the process of camera calibration, such as insufficient accuracy and poor accuracy. Based on the seagull algorithm, the adaptive differential evolution algorithm is combined with the seagull algorithm to optimize the multicamera calibration. The seagull algorithm can achieve good results on multiparameter problems and effectively avoid falling into local optima. In this paper, the adaptive differential search algorithm is adopted to improve the local search ability and optimize the local search and global search ability. According to Zhang Zhengyou's method, the calibrated parameter is obtained, in which the parameter is used as the initial value. Then, taking the minimum mean error as the criterion, the improved seagull algorithm (SOA-SaDE) is used to establish the objective function, and the internal parameters and distortion coefficient of the camera are further solved. Verification experiments showed that the fusion algorithm has less reprojection error and higher calibration accuracy gull algorithm.

1. Introduction

Multiple cameras are widely used in various fields. Multicamera fusion can provide a wider range of real-time scene information. As an important tool of machine vision, improving camera calibration accuracy is the focus of its research. It has more accurate camera calibration and is more conducive to the camera image matching, recognition positioning, and other follow-up operation accuracy.

At present, the field of deep-sea exploration is developing in a deeper and farther direction. Underwater robots are widely used in underwater exploration due to their wide range of activities and strong autonomy. Compared with other underwater detection methods, robot visual observation is at a close range. Object detection has unparalleled advantages. It can not only track and detect underwater targets in real time but also record underwater video materials for researchers to study deep-sea objects after landing. Therefore, improving the accuracy of camera calibration and eliminating image distortion are of great significance to the research in the deep-sea field.

The camera calibration uses the calibration object in the space and obtains the calibration object different space positions through the position change. The mathematical model can be constructed by the relation between the object space coordinate system and the camera image coordinate system, Then, the internal parameters and distortion coefficients of the camera are calculated and solved.

Camera calibration is a multidimensional nonlinear problem. Each calibration image corresponds to different external parameters. Therefore, it is very difficult to find the concrete calculation equation in the process of back-projection to the calibration point. Common optimization algorithms, such as the pseudo-Newton method, cannot be applied to the optimization of camera calibration because they must depend on the specific functional form. Therefore, the researchers focus on using optimization algorithm to optimize the camera calibration results. Huang et al. [1] used the classical particle swarm optimization (PSO) algorithm to optimize the camera calibration results, took the absolute value of average relative error as the objective function, optimized the camera parameters, and improved the calibration accuracy. Qin et al. [2] proposed a full-parameter

adaptive mutation PSO. They used the algorithm to optimize the camera intrinsic parameters and improve the adaptive mutation rate of the particles according to the average particle distance of the particle swarm to optimize the camera calibration results. Xu et al. [3] introduced a diffusion mechanism to improve the local optimal solution problem of particle swarm optimization. Xiang et al. [4] proposed a calibration method based on depth learning. It can be calibrated by inputting the coordinates of the original image by improving the approximation ability of the DNN network. It can be used in large areas, multiple camera angles, and other complex environments. However, deep learning network training requires GPU acceleration and requires high computer configuration, and training takes time, so it is difficult to calibrate quickly. Lei et al. [5] combined PSO with simulated annealing (SA) algorithm, obtained the initial parameters of camera calibration by least squares, and optimized the camera parameters by the hybrid algorithm. This method improves the calibration accuracy of the camera. Based on the Levenberg–Marquardt algorithm, Liu Jiachen [6] corrected the reprojection error by using the improved beam adjustment method to reduce the error of 3D reconstruction. However, in this process, the formula is tedious and the computation is complex.

The seagull algorithm is a kind of metaheuristic, which is suitable for solving multiparameter optimization problems. When the traditional seagull algorithm is used, the initial seagull swarm has strong randomness, but the optimization process is inefficient and easy to fall into local optimization. A hybrid optimization algorithm is proposed in [7]. The algorithm is based on chaotic differential evolution and distribution estimation, which can obtain a high-precision solution. A hybrid algorithm of adaptive gravity search and differential evolution (DE) is proposed in [8]' which keeps the diversity of the population. The DE is used for local search and plays a big role. Inspired by the successful application of the above hybrid algorithm, seagull algorithm used in this paper, and seagulls algorithm compared to traditional optimization algorithm, the principle is simple and easy to implement, is suitable for multiobjective optimization, and does not coincide with the position of the population in an iterative process, reducing repetitive iterations and improving the effectiveness of iterations. To avoid the local optimization in the calibration calculation, this paper combines the seagull algorithm and the adaptive differential evolution algorithm and improves the seagull algorithm by absorbing the strong local searching ability of ADE, improving the accuracy and stability of camera calibration.

This article focuses on the key issues that need to be solved for camera calibration. Aiming at the problem of low calibration accuracy and obvious reprojection errors, a fusion algorithm (SOA-SaDE) is proposed based on the seagull algorithm and the adaptive differential algorithm. The internal parameters and distortion coefficients calibrated based on the pinhole camera model are optimized by the SOA-SaDE algorithm. Experiments show that the algorithm proposed in this paper effectively reduces the error of camera reprojection; the reprojection error is reduced by 63.03%, which is 16.75% higher than the effect of the seagull algorithm, and provides a feasible method for reducing the camera reprojection error.

2. Basic Principles of the Seagull Algorithm

In 2018, the seagull algorithm (SOA) proposed a new population-based intelligent optimization algorithm, which simulates the migrating and foraging behavior of seagulls to optimize the target [9].

The seagull algorithm is divided into two parts; they are migration and foraging. Migration is the behavior that is the movement of seagulls from their current position to a more livable position. Migration behavior affects the global exploration ability of the seagull algorithm. Foraging is the behavior of seagulls attacking the food in the current sea area during the flight. The foraging behavior affects the ability of the seagull algorithm for local exploitation.

There are three important points to be paid attention to during a Gull's migration from one place to another: avoiding collisions between individuals, the best orientation of its position, and its proximity to the best position. To avoid the collision with the seagulls, the algorithm uses the additional variable A to adjust the seagulls' position:

$$\overrightarrow{C_s} = A \times \overrightarrow{P_s(\mathbf{x})}, \tag{1}$$

where A represents the migration behavior of seagulls in each given search space The size of A is controlled by B:

$$A = f_c - \left(\left(\mathrm{t} \times \left(\frac{f_c}{\mathrm{Max}_{\mathrm{iteration}}}\right)\right)\right). \tag{2}$$

The final size of A decreases linearly from $2 \longrightarrow 0$ according to the number of iterations. After ensuring that individual gulls do not collide with each other, move all gulls closer to the best:

$$\overrightarrow{M_s} = B * \left(\overrightarrow{\mathbf{P_{best}}(\mathbf{x})} - \overrightarrow{P_s(\mathbf{x})}\right), \tag{3}$$

where $\mathbf{M}_s$ represents the convergence direction of the individual toward the optimal seagull and B is an important parameter for balancing the exploration and development capability of the algorithm. It changes according to

$$B = 2 * A^2 * \mathrm{rand}, \tag{4}$$

where rand is a random number in the range [0,1].

After calculating the direction of convergence of each gull, each gull began to move toward this position:

$$\overrightarrow{D_s} = \left|\overrightarrow{C_s} + \overrightarrow{M_s}\right|, \tag{5}$$

where $\overrightarrow{D_s}$ is the position of the seagull.

Seagulls can constantly change their angle and speed of attack during the migration, when attacking prey, seagulls will carry out the spiral movement. The position of the

seagull in the 3D is

$$x = r * \cos(k), \tag{6}$$

$$y = r * \sin(k), \tag{7}$$

$$z = r * k, \tag{8}$$

$$r = \mathrm{u} * e^{kv}, \tag{9}$$

where k is a random number at $[0, 2\pi]$. The algorithm controls the spiral radius r by u and v, and they are usually 1. According to the new position of seagull, the updated formula of the whole position of the seagull is as follows:

$$\overrightarrow{P_s(x)} = \left(\overrightarrow{D_s} \times x \times y \times z\right) + \overrightarrow{\mathbf{P}_{\mathbf{best}}(\mathbf{x})}, \tag{10}$$

where $\overrightarrow{\mathbf{P}_s(\mathbf{x})}$ is the attack position of the seagull.

SOA: the flow of the algorithm is as follows:

Step 1: initialization parameter

Step 2: calculate the fitness value for each seagull and the objective function value

Step 3: calculate $\overrightarrow{\mathbf{D}_s}$ according to formulas (1)–(5)

Step 4: calculate according to formulas (6)–(10)

Step 5: update position information and fitness values for the best seagull, internation = internation + 1

Step 6: if internation > $\mathrm{Max}_{\mathrm{interation}}$, skip to Step 7, or slip to Step 3

Step 7: output the optimal seagull position and fitness value

3. Adaptive Differential Evolution Algorithm

3.1. Differential Evolution. Differential evolution algorithm (DE) is a kind of evolutionary algorithm (EA), which is a search strategy for solving polynomial fitting problems put forward by R. Storn and K. Price. This algorithm is based on genetic algorithm and other evolutionary ideas; its essence is to optimize the multiobjective and multidimensional space to achieve the overall optimal solution of the goal. The differential evolution algorithm retains the crossover, mutation, and copy operations in the genetic algorithm. It differs from GA in which the variation vector is generated by the parent difference vector, which crosses with the incidental individuals to generate new individuals and then selects among them. Therefore, it has a better iterative approximation effect than GA. The differential algorithm is divided into two stages: population initialization and iteration [10].

3.1.1. Population Initialization. Suppose that $G = 0, 1, 2 \ldots G_{\max}$ stands for evolution algebra. Then, the 1st individual in the population under the current algebra is represented as

$$\overrightarrow{X}_{\mathrm{i,G}} = \left(\mathbf{x}^1_{\mathrm{i,G}}, \mathbf{x}^2_{\mathrm{i,G}}, \ldots, \mathbf{x}^j_{\mathrm{i,G}}\right), \quad j = 1, 2, \ldots, D, \tag{11}$$

where D is the dimension of the individuals of the population. In population initialization, the initial population is required to cover the entire search space R^D, and the initialization formula is shown in (8):

$$x_{j,i,0} = x_{j,\min} + \mathrm{rand}_{i,j}[0,1] * \left(x_{j,\max} - x_{j,\min}\right), \tag{12}$$

where $\mathrm{rand}_{i,j}[0,1]$ is a uniformly distributed random number in the interval $[0, 1]$ and $x_{j,\min}$ and $x_{j,\max}$ are the lower and upper bounds of the individual optimization variables $\overrightarrow{X}$, j, respectively.

3.1.2. Differential Mutation Operation. In each iteration, three individual vectors are randomly selected from the population $\overrightarrow{\mathbf{X}}_{\mathbf{r1,G}}$, $\overrightarrow{\mathbf{X}}_{\mathbf{r2,G}}$, and $\overrightarrow{\mathbf{X}}_{\mathbf{r3,G}}$, and $\mathrm{r1} \neq \mathrm{r2} \neq \mathrm{r3}$, according to (13), a new individual $\overrightarrow{V}_{i,G}$ can be created; this individual is a variation vector:

$$\overrightarrow{\mathbf{V}}_{\mathbf{i,G}} = \overrightarrow{\mathbf{X}}_{\mathbf{r1,G}} + \mathrm{F} * \left(\overrightarrow{\mathbf{X}}_{\mathbf{r1,G}} - \overrightarrow{\mathbf{X}}_{\mathbf{r3,G}}\right), \tag{13}$$

where F is the variation scale factor, which is used to scale the difference vector to control the search step. In general, the variation scale factor F is in the [0,2] interval.

3.1.3. Cross Operation. In the crossover step, the algorithm adopts discrete crossover. The test vector $\overrightarrow{U}_{i,G}$ is generated by crossing mutation vector $\overrightarrow{\mathbf{V}}_{\mathbf{i,G}}$ and target vector $\overrightarrow{\mathbf{X}}_{\mathbf{r1,G}}$ according to the binomial method. The specific operation is shown in

$$u^i_{i,G} = \begin{cases} v^j_{i,G}, & \text{if } \mathrm{rand}_{i,j}[0,1] \le C_r, \\ x^j_{i,G}, & \text{otherwise}, \end{cases} \tag{14}$$

where C_r is the crossover probability factor, and the crossover operator can enhance the diversity of the population. The value of C_r is generally in the interval [0,1], which is a random number uniformly distributed in the interval [0,1]. In the j dimension, if the random generating number is less than C_r, the test vector inherits the variation vector and vice versa.

3.1.4. Select Operation. The selection process selects the more adaptable child from each iteration as the next generation by comparing the child with the corresponding parent based on the value of the fitness function; its selection method is shown as

$$\overrightarrow{\mathbf{X}}_{\mathbf{i,G}} = \begin{cases} \overrightarrow{\mathbf{U}}_{\mathrm{i,G}}, & \text{if } \left(\overrightarrow{\mathbf{U}}_{\mathbf{i,G}}\right) \le f\left(\overrightarrow{\mathbf{X}}_{\mathbf{i,G}}\right), \\ \overrightarrow{\mathbf{X}}_{\mathrm{i,G}}, & \text{otherwise}. \end{cases} \tag{15}$$

3.2. Adaptive Differential Evolution. According to formulas (13) and (14), F and C_r are two important parameters in DE, and the choice of their values will affect the optimization

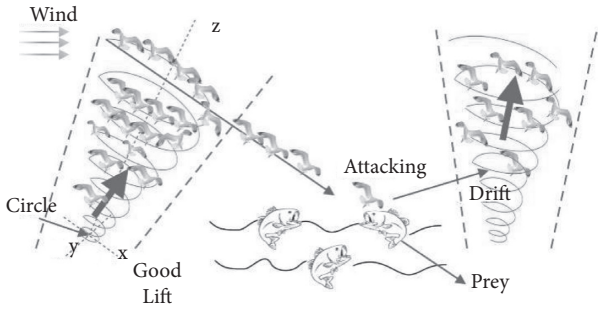


Figure 1: The biological principle of the seagull algorithm.

effect. However, in the DE algorithm, the values are all constant and cannot be well adapted to various problems, especially, for complex high-dimensional problems. Therefore, Janez Brest introduced adaptive control parameters in 2006. The improved algorithm is called the adaptive differential evolution algorithm (SaDE) [11]. The adaptive control parameters F and C_r are represented as

$$F_i^{t+1} = \begin{cases} F_l + \text{rand}_1 \times F_u, & \text{if rand}_2 \le \tau_1, \\ F_i^t, & \text{otherwise}, \end{cases}$$
$$CR_i^{t+1} = \begin{cases} \text{rand}_3, & \text{if rand}_4 < \tau_2, \\ CR_i^t, & \text{if rand}_4 \ge \tau_2, \end{cases} \tag{16}$$

where rand1, rand2, rand3, and rand4 is the random number in [0, 1], $\tau 1$ and $\tau 2$ indicate the probability of conversion, and F_l and Fu are the boundary scaling factor.

4. Camera Internal Parameter Optimization Design and Application Based on the Hybrid Algorithm

4.1. Design of Hybrid Algorithms. This paper proposes a combination of the SOA algorithm and the SaDE algorithm. It aims to improve the search precision, avoid the population falling into the local extremum, and maintain the population diversity in the later iteration. In the minimization problem, if the fitness of the iteration is greater than that of the previous generation, the location region of the iteration is not good, so the randomness of the population should be strengthened to improve the search range. If the fitness intelligence of the iterated individuals is less than that of the optimal individuals of the previous generation, then the region has the potential value, so we should continue searching along the region.

Combining the SOA algorithm with the SaDE algorithm to realize the internal parameter optimization, in each population iteration, the minimum fitness value of the population is calculated as $f_{\min}^i$. In the i+1 iteration, when $f_{\min}^{i+1} < f_{\min}^i$, SOA is used for optimization, when $f_{\min}^{i+1} > f_{\min}^i$, SaDE is used for optimization.

4.2. Establishment of the Objective Function. The objective function of the camera calibration problem is established as follows:

$$f = \sum_{i=1}^{N} \left\| p_{ij} - p\left(f_x, f_y, u_0, v_0, k_1, k_2, k_3, p_1, p_2, R, T\right) \right\|, \tag{17}$$

where p_{ij} is the actual pixel coordinates of the j corner, N is the number of corners, P is the calculated pixel coordinates, f_x, f_y, u_0, and v_0 are the camera's internal parameter, k_1, k_2, k_3, p_1, and p_2 are the radial distortion coefficient and the tangential distortion coefficient, and R and T are the rotation translation matrix of the image.

4.3. Parameter Initial Value Solution. The camera imaging relationship is

$$z_c \begin{bmatrix} \mathbf{x}_d \\ \mathbf{y}_d \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{f}_{c1}}{d_x} & \mathbf{0} & \mathbf{u}_0 & \mathbf{0} \\ \mathbf{0} & \frac{\mathbf{f}_{c2}}{\mathbf{d}_y} & \mathbf{v}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{T}_{3\times1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_w \\ \mathbf{y}_w \\ \mathbf{z}_w \\ \mathbf{1} \end{bmatrix}. \tag{18}$$

The above formula represents the transformation of point $[\mathbf{x}_w \ \mathbf{y}_w \ \mathbf{z}_w \ \mathbf{1}]^{\mathbf{T}}$ in the world coordinate system to point $[\mathbf{x}_d, \mathbf{y}_d, \mathbf{1}]^T$ in the pixel coordinate system in the linear model. M_A is the internal parameter matrix which represents the intrinsic geometry of the camera. The mathematical model is

$$\mathbf{M_A} = \begin{bmatrix} \mathbf{f}_x & \mathbf{0} & \mathbf{u_0} \\ \mathbf{0} & \mathbf{f}_y & \mathbf{v_0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}, \tag{19}$$

where $f_x = f_{c1}/d_x$ and $f_y = f_{c2}/d_y$, f_{c1} and f_{c2} are the focal length of the camera, d_x and d_y are the physical lengths of the pixels, and u_0 and v_0 are the intersections of the camera's optical axis and the image plane.

According to the above expression, the camera internal parameters can mainly solve 4 parameters. They are f_x, f_y, u_0, and v_0. We can obtain the initial value of f_x, f_y, u_0, and v_0 by the imaging relation. The initial value is obtained under the ideal condition, but the actual lens has distortion, so it needs to introduce distortion coefficient k_1, k_2, k_3, p_1, and p_2 to correct it.

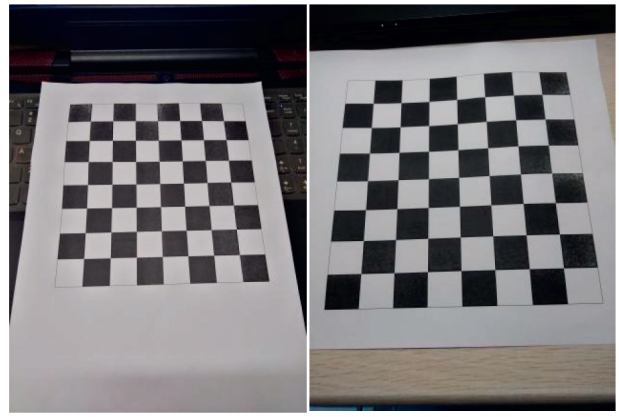


Figure 2: Calibration pictures.

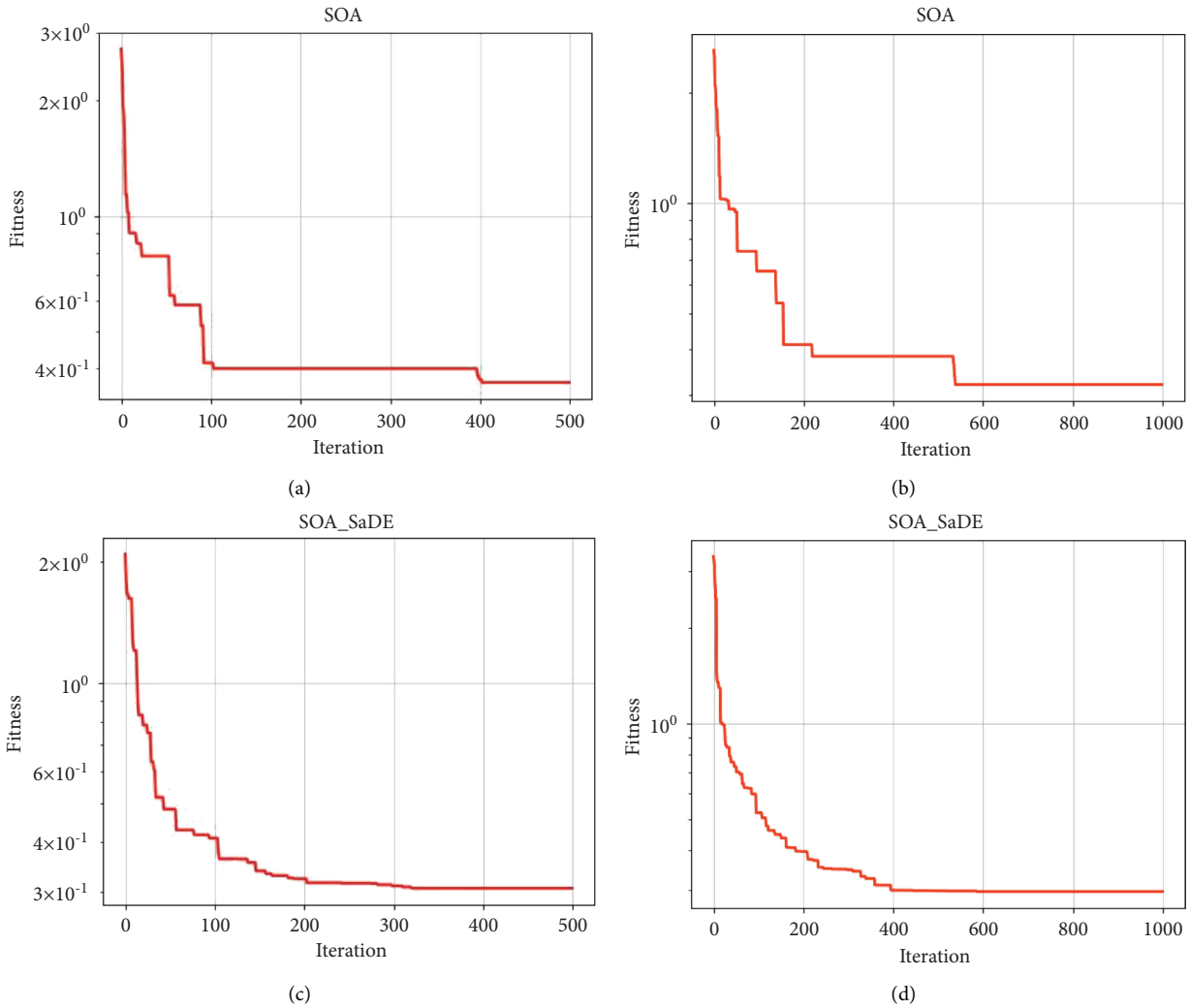


Figure 3: Object function curve. (a) SOA iterates 500 objective function curves. (b) SOA iterates 1000 objective function curves. (c) SOA-SaDE iterates 500 objective function curves. (d) SOA-SaDE iterates 1000 objective function curves.

Table 1: Calibration results of the Zhang Zhengyou calibration method.

Parameter	Zhang Zhengyou calibration method
f_x	3575.57253
f_y	3558.46976
u_0	1541.29510
v_0	1885.83655
k_1	0.14037782
k_2	−1.12437053
p_1	0.0079399
p_2	−0.00243523
k_3	1.45905723
Error	0.828057702

Table 2: Calibration results of camera internal parameters for 500 iterations.

Parameter	Method	
	SOA	SOA-SaDE
f_x	3572.85291	3640.19021
f_y	3556.14008	3614.61743
u_0	1546.41000	1540.28375
v_0	1886.40299	1888.24897
k_1	−0.00130089	−0.25517776
k_2	0.02047008	0.96152329
p_1	−0.00114037	0.00163706
p_2	−0.00115161	−0.00013975
k_3	−2.72448436	1.34834219

Table 3: Iterative calibration results of 1000 camera internal parameters.

Parameter	Method	
	SOA	SOA-SaDE
f_x	3581.17874	3623.59017
f_x	3564.41244	3601.84834
u_0	1544.83354	1538.79344
v_0	1886.52839	1888.07675
k_1	0.00972433	−0.23023645
k_2	−0.00264705	1.33036931
p_1	0.00050738	0.00148013
p_2	−1.09968486	0.00344624
k_3	0.00202380	−0.75344467

Table 4: Reprojection errors.

Number of iterations	Method	
	SOA	SOA-SaDE
500	0.36774533	0.30615074
1000	0.32056968	0.29615753

The mathematical model of radial distortion is

$$\begin{cases} x_d = x_u\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right), \\ y_d = y_u\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right). \end{cases} \tag{20}$$

The mathematical model of tangential distortion is

$$\begin{cases} x_d = x_u + \left[2p_1 y_u + p_2\left(r^2 + 2x_u^2\right)\right], \\ y_d = y_u + \left(2p_2 x_u + p_1\left(r^2 + 2y_u^2\right)\right). \end{cases} \tag{21}$$

In the above formula,

$$r^2 = x_u^2 + y_u^2. \tag{22}$$

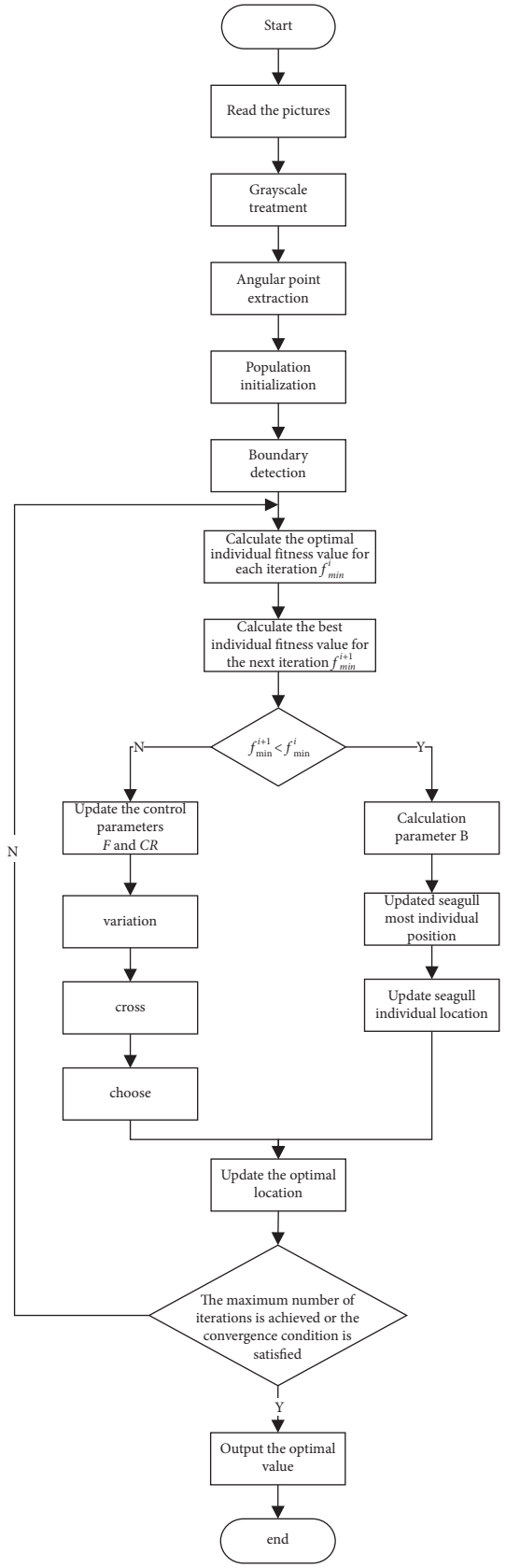


Figure 4: Algorithm flow of internal parameter optimization.

Contacting formulas (20)–(22), we can obtain

$$\begin{cases} x_d = x_u\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + 2p_1 x_u y_u + p_2\left(3x_u^2 + y_u^2\right), \\ y_d = y_u\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + 2p_2 x_u y_u + p_1\left(3x_u^2 + y_u^2\right), \end{cases} \tag{23}$$

where (x_d, y_d) is the image coordinate of the ideal camera model and (x_u, y_u) is the real image coordinate of the nonlinear camera model. The initial value under the distortion is obtained by formula (23).

4.4. Hybrid Algorithm Application. First, the initial value of the internal parameter $f_x, f_y, u_0, v_0, k_1, k_2, k_3, p_1$, and p_2 then initializes the seagull population to generate N different seagull individuals. Initializing population position, parameters A, B, and $\text{Max}_{\text{internation}}$, set the appropriate parameters $f_c = 2$, $u = 1, v = 1$, and initialize the current iteration number $t = 0$. The fitness function is obtained from the objective function and is defined as

$$\text{fitness} = \min \sum_{i=1}^{m} \sqrt{(x-u)^2 + (y-v)^2}, \tag{24}$$

where (x, y) is the actual pixel coordinates obtained by the corner extraction algorithm, (u, v) is the pixel coordinates calculated by camera imaging relations, and m is the total number of corners.

The algorithm flow of camera internal parameter optimization using the hybrid algorithm is shown in Figure 1.

5. Experiment

5.1. Experimental Design. The experiment uses camera RealSense D435i as the hardware platform and uses *Python* as the software development platform. In this experiment, 16 images were taken, and the camera internal parameters and distortion coefficients were calibrated based on these images. The pictures are shown in Figure 2.

5.1.1. Specific Calibration Steps

Step 1: the camera calibration is realized based on OpenCV-Python

Step 2: according to the calibration parameters obtained by the traditional method, the upper and lower interval of the parameters are set, the scope is limited, and the parameters are initialized

Step 3: the results of 300, 500, and 1000 iterations in the seagull algorithm are brought in

Step 4: the initial parameters are brought into the SOA-SaDE fusion algorithm to calculate the iterative results of 300 times, 500 times, and 1000 times, respectively

Step 5: compare the results of the traditional method and the improved algorithm

5.2. Analysis of Experimental Results. Figures 3(a)–3(d) show the results of 500 and 1000 iterations of the general SOA algorithm and the SOA-SaDE algorithm. As can be seen from the graph that the SOA algorithm converges fast, but it is easy to fall into the local optimum. However, the SOA-SaDE algorithm is effective in dealing with the local optimum and can jump out of the local optimum and continue to converge toward the global optimum. When the image curve is flat, it is close to the optimal target value.

Table 1 is the calibration result of Zhang Dingyou's method; Tables 2–4 are the optimization results of the SOA algorithm and SOA-SaDE fusion algorithm after 500 and 1000 iterations, respectively. According to the reprojection error after optimization is calculated, the results of Zhang's method can be well optimized by these two optimization algorithms. Moreover, the SOA-SaDE algorithm proposed in this paper is generally superior to the SOA algorithm. The fusion algorithm proposed in this paper is robust and reusable and can improve the local convergence of the SOA algorithm and get better results, as shown in Figure 4.

6. Conclusion

In order to solve the local convergence problem, SaDE algorithm is not easy to fall into local convergence. When the optimal individual fitness of each iteration is less than the value of the last iteration, local convergence may have occurred. SaDE was used to optimize population parameters and increase population diversity. This paper presents a new optimization method for camera internal parameters. The algorithm is based on seagull algorithm and adaptive parametric differential evolution algorithm. In this paper, the two are integrated into a framework according to certain mechanisms. By comparing the reprojection errors of Zhang Zhengyou's calibration algorithm, SOA algorithm, and SOA-SADE fusion algorithm, it can be seen that the gull differential evolution algorithm can get smaller errors. The calibration accuracy is improved to a certain extent. The experimental results show that the gull difference algorithm has good accuracy and feasibility for camera internal parameters optimization. The algorithm can be combined with practical engineering cases to solve multidimensional nonlinear optimization problems accurately and effectively. There are many similar bionic algorithms, such as monarch butterfly optimization (MBO), earthworm optimization algorithm (EWA), elephant herding optimization (EHO), moth search (MS) algorithm, slime mould algorithm (SMA), and Harris hawks optimization (HHO). [12, 13] These bionic algorithms have their own unique characteristics and can play a very good role in specific engineering fields. We believe that, in the follow-up research, we can comprehensively consider the advantages and disadvantages of each algorithm, merge different algorithms, learn from each other, apply it to the vision of underwater robots for deep-sea exploration, and provide a solution for improving the accuracy of underwater robot vision detection. The underwater environment is complicated. How to solve the problem of underwater imaging should consider the refraction of light brought by water, and the problem of

floating objects in the water affecting image clarity is the next issue we need to consider.

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.

References

[1] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for IoV in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[2] R. Qin, Y. Yang, and F. Li, "Calibration of monocular camera based on full-parameter adaptive mutation particle swarm optimization algorithm," *Journal of Southeast University (Natural Science Edition)*, vol. 47, pp. 193–198, 2017.

[3] C. Xu, Y. Liu, and Yi Xiao, "etc. Optimization methods of camera internal parameters based on improved particle swarm algorithm," *Progress in Laser and Optoelectronics*, vol. 57, no. 6, Article ID 061501, 2020.

[4] X. Xu, Z. Fang, J. Zhang et al., 2021.

[5] L. Yang, H. Zhang, and C. Wang, "Hybrid particle swarm optimization method for accurate camera calibration," *Progress in Laser and Optoelectronics*, vol. 56, no. 21, pp. 171–179, 2019.

[6] J. Liu, *Research and Implementation of Multi-Camera Calibration Algorithm Based on One-Dimensional Calibration Rod*, Hefei University of Technology, Hefei, China, 2020.

[7] F. Zhao, F. Xue, Y. Zhang, W. Ma, C. Zhang, and H. Song, "A hybrid algorithm based on self-adaptive gravitational search algorithm and differential evolution," *Expert Systems with Applications*, vol. 113, pp. 515–530, 2018.

[8] M. A. Elaziz, S. Xiong, K. P. N. Jayasena, and L. Li, "Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution," *Knowledge-Based Systems*, vol. 169, pp. 39–52, 2019.

[9] G. Dhiman and V. Kumar, "Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems," *Knowledge-Based Systems*, vol. 165, pp. 169–196, 2019.

[10] L. Chen, *Improved Adaptive Differential Evolution Algorithm and its Application Research*, Donghua University, Shanghai, China, 2012.

[11] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 646–657, 2006.

[12] W. Li, G.-G. Wang, and A. H. Gandomi, "A survey of learning-based intelligent optimization algorithms," *Archives of Computational Methods in Engineering*, vol. 28, no. 5, pp. 3781–3799, 2021.

[13] G.-G. Wang, A. H. Gandomi, A. H. Alavi, and D. Gong, "A comprehensive review of krill herd algorithm: variants, hybrids and applications," *Artificial Intelligence Review*, vol. 51, no. 1, pp. 119–148, 2019.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5648168, 8 pages
https://doi.org/10.1155/2021/5648168

Research Article

Time-Aware Cross-Platform IoT Service Recommendation with Privacy Preservation

Can Zhang, Junhua Wu, Chao Yan, and Guangshun Li

School of Computer Science, Qufu Normal University, Rizhao, Shandong, China

Correspondence should be addressed to Junhua Wu; shdwjh@163.com

Received 3 May 2021; Revised 26 June 2021; Accepted 8 November 2021; Published 6 December 2021

Academic Editor: Dou Wanchun

IoT service recommendation techniques can help a user select appropriate IoT services efficiently. Aiming at improving the recommendation efficiency and preserving the data privacy, the locality-sensitive hashing (LSH) technique is adopted in service recommendation. However, existing LSH-based service recommendation methods ignore the intrinsic temporal feature of IoT services. In light of this challenge, we integrate the temporal feature into the conventional LSH-based method and present a time-aware approach with the capability of privacy preservation for IoT service recommendation across multiple platforms. Experiments on a real-world dataset are conducted to validate the advantage of our proposed approach in terms of accuracy and efficiency in recommendation.

1. Introduction

The rapidly increasing number of IoT devices and services are continuously producing a vast amount of data. Among those data, the quality of service (QoS) data is generated when a user invoked a service. Service recommendation techniques can employ the historical quality data to reduce the users' service selection burden and help them find out appropriate services efficiently. Typically, taking collaborative filtering recommendation algorithm for example, through predicting the quality value when a target user invokes a service according to its similar users' quality data, a recommendation system can suggest a list of optional services for a target user.

However, in the IoT environment, traditional recommendation algorithms are often nonavailable in practice [1]. One fundamental reason is that users may choose services from different platforms. For example, in the context of a smart home, users may use service provided by Hikvision to capture video data, service from Huawei to monitor the air quality, and service from Xiaomi to control household appliances. Therefore, the historical QoS data of users is not centralized, but stored across different platforms [2, 3]. Moreover, because of conflicts in economic interests and data privacy concerns, service providers are unwilling to share their data with each other. Furthermore, the volume of the historical QoS data is often massive, thus it is impractical to share such large amount of data with other platforms.

Considering these challenges, the locality-sensitive hashing (LSH) algorithm [4] is adopted in service recommender system. Specifically, through hashing high dimensional historical quality data of users into scalar hash value, user privacy is preserved. Besides, as the hashing process is conducted individually in each platform, there is no need to transfer historical data.

However, most LSH-based service recommender systems view the historical QoS data as static and unique, rarely taking other context factors (e.g., time) into account. This may lead to less reasonable and accurate recommended results.

In light of the abovementioned challenges, we integrate temporal dimension into the conventional LSH-based recommendation method and present a time-aware recommendation approach for IoT services across multiple platforms. The proposed approach achieves higher accuracy than conventional recommendation approaches that have the capability of privacy preservation. The contributions of this paper are summarized as follows.

(1) We improve the conventional LSH-based recommender system by incorporating it with a time factor, so as to adapt the intrinsic feature of IoT services, and achieve better performance in IoT service recommendation.

(2) We conduct extensive experiments on a real-world dataset to validate the advantage of our method. Experimental results demonstrate that the proposed method outperforms the other state-of-the-art approaches in both recommendation accuracy and efficiency, while preserving data privacy among multiple platforms.

The rest of this article is structured as follows. Section 2 reviews recent work on recommender system for IoT service. Section 3 formulates the problem of IoT service recommendation and presents our motivation. Section 4 describes a time-aware cross-platform IoT service recommendation approach with privacy-preserving capability. Section 5 demonstrates the implementation and results of experiments. Section 6 summarizes the whole paper and addresses future work.

2. Related Work

In this section, we review the related research work on time-aware IoT services recommendation with privacy preservation from the following three aspects.

2.1. IoT Service Recommendation. Most existing researches on recommender system for IoT services can be classified into three groups: content-based filtering, collaborative filtering, and link-based methods [1]. In [5, 6], Mashal et al. formulated the IoT recommendation as a hypergraph model, which connects users, objects, and services with hyperedges and presents a graph-based recommender system considering the unique feature of IoT services. Yao et al. put forward a unified framework based on probabilistic factor; they calculated user similarity and device similarity and then fused them together to make more accurate recommendation [7, 8]. In [9], Mashal et al. proposed a multiagent approach to establish a distributed recommender system in IoT environment. However, in the above literature, the time factor, which is one of the most common features of IoT service, is not considered in IoT recommendation system. This may decrease the accuracy of recommendation result.

2.2. Time-Aware Service Recommendation. A number of research works have taken the time factor into account to obtain more accurate recommendation results. In [10, 11], Wang and Zhu present a spatial-temporal QoS value prediction approach. Temporal sequences of historical QoS data are employed to build feature models, while the spatial information of web services is exploited to reduce the searching space. In [12], Zhong and Fan built a time-aware recommender system for mashup creation. An extraction method for service pattern based on LDA and time series prediction is presented. In [13], Yu and Huang took both time and location factors into account; they represent the temporal quality data as a three-dimensional matrix and use CF techniques to make prediction and recommendation.

All of the abovementioned methods employed the time factor to enhance the performance of recommendation system; however, none of those methods take privacy preservation into account, which is necessary when QoS data is collected from different platform to obtain more comprehensive user preference.

2.3. Privacy-Preserving Service Recommendation. As IoT service data from different platform contains sensitive user information, it is crucial to preserve user's privacy while sharing valuable data across platforms [14–19]. In [20], Ma et al. proposed K-anonymity method to protect user privacy through hiding sensitive user identification information. However, this may influence the data availability and decrease the performance of recommendation systems accordingly. In [21], Dou et al. suggest not publishing all the observed QoS data but the optimal data; however, users may still leak some sensitive information. In [4], Qi et al. first introduced LSH technique into service recommendation; by hashing high dimensionality QoS data into low dimensionality indices, data privacy can be protected in an efficient way. However, time factor is still not considered in the approach.

As a conclusion, existing IoT service recommendation methods fail in taking time factor and privacy preservation into account simultaneously [22–26]. In light of this challenge, we improve the conventional LSH method and present a time-aware cross-platform IoT service recommendation algorithm with privacy preservation.

3. Formulation and Motivation

3.1. Problem Formulation. Concretely, our time-aware cross-platform IoT service recommendation model can be formulated as a five-tuple $\text{IoTSerRec}_{\text{t-LSH}}$ (SP, U, IS, H, u_{target}), where

(1) $SP = \{sp_1, \ldots, sp_c\}$: $sp_k (1 \le k \le c)$ represents the k-th IoT service platform. Each platform provides a part of a user's QoS data.

(2) $U = \{u_1, \ldots, u_m\}$: $u_k (1 \le k \le m)$ denotes the k-th IoT service user. As a user may invoke IoT service from different platforms, his/her historical QoS data is stored across multiple platforms.

(3) $IS = \left\{is_{1,1}, \ldots, is_{1,n_1}, \ldots, is_{c,1}, \ldots, is_{c,n_c}\right\}$: $is_{i,j}$ $(1 \le i \le c, 1 \le j \le n_i)$ denotes the j-th IoT service on the i-th platform; n_i means the number of IoT services provided by the i-th platform.

(4) $Q = \{Q_1, \ldots, Q_t\}$: $Q_k (1 \le k \le t)$ denotes the historical user-service quality data at the k-th time slot. $Q_k = \left\{q_{k,1,1}, \ldots, q_{k,1,n}, \ldots, q_{k,m,1}, \ldots, q_{k,m,n}\right\}$: $q_{k,i,j}$ denotes the quality value of the j-th IoT service when the i-th user invokes at the k-th time slot.

(5) $u_{\text{target}} \in U$: the user who needs recommendation service.

3.2. Motivation. As shown in Figure 1, there are three IoT service platforms, Alibaba (denoted as sp_1), Huawei (denoted as sp_2), and Google (denoted as sp_3). IoT services $is_{1,1}, \ldots, is_{1,n_1}$ are deployed in Alibaba, $is_{2,1}, \ldots, is_{2,n_2}$ are deployed in Huawei, and $is_{3,1}, \ldots, is_{3,n_3}$ are deployed in Google. Users $\{u_1, u_2, u_3\}$ keep on invoking IoT services in the above platforms to finish certain tasks.

For conventional user-based CF recommendation approach, if we want to make recommendation for u_{target}, we need to find similar users of u_{target} according to the historical QoS data first [27–29]. However, there are three challenges in the collaboration process among Alibaba, Huawei, and Google. (1) Considering user privacy, each platform cannot share their own QoS data to each other [30–33]. (2) Since the user-service quality data keeps updating overtime, its volume becomes increasingly massive, which significantly reduces the collaboration efficiency and scalability [19, 34]. (3) As a user often invokes an IoT service constantly, a user-service pair is made up of a series of QoS values, which makes traditional user-based recommendation approach unsuitable for this situation [35, 36].

Considering the abovementioned challenges, we present a novel LSH-based IoT service recommendation method. The method will be elaborated in Section 4.

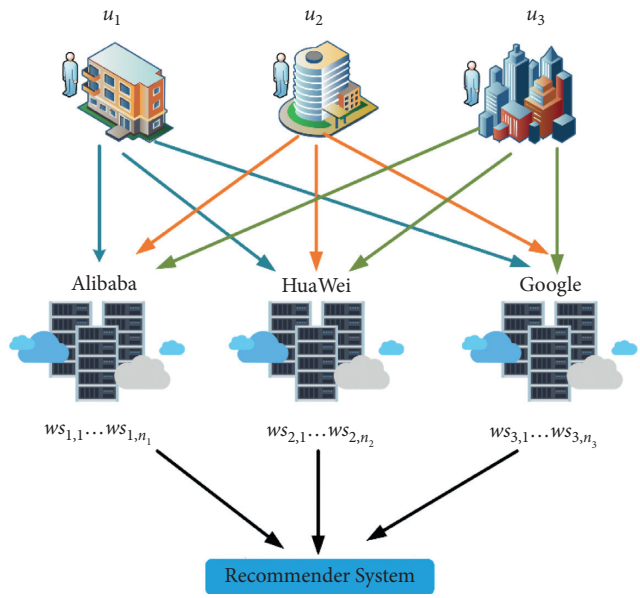


Figure 1: Cross-platform IoT service recommendation: an intuitive example.

4. Privacy-Preserving and Time-Aware LSH-Based IoT Service Recommendation Algorithm: IoTSerRec$_{\text{t-LSH}}$

In this section, we present a time-aware LSH-based recommendation algorithm for IoT services across multiple platforms, which is denoted as IoTSerRec$_{\text{t-LSH}}$. In summary, our proposed algorithm is divided into three steps:

Step 1: calculating user indices at each time slots based on LSH. For an IoT service platform, $sp_k (1 \le k \le z)$, each user u's QoS data in sp_k at the t-th time slot is mapped into $H_{t,k}(u)$, which denotes the k-th subindex of user u at the t-th time slot. User u's complete index at the t-th time slot $H_t(u)$ is merged as $(H_{t,1}(u), \ldots, H_{t,c}(u))$ offline. The set of indices of all users at each time slot is regarded as a hash table.

Step 2: finding out top-K users most similar to u_{target}. Compute u_{target}'s complete index at each time slot according to step 1. Then, calculate the similarity between u_{target} and other users according to the user indices at each time slot, and return the top-K users with the highest similarity score with u_{target}.

Step 3: recommending IoT service for u_{target}. Predict the quality value of services never invoked by u_{target} based on u_{target}'s top-K neighbors' quality data. Then, retrieve the quality-optimal one to u_{target}.

Next, we demonstrate the implementation of each step in details.

4.1. Step 1: Calculating User Indices at Each Time Slots Based on LSH. In the IoT environment, a user (i.e., u_i) usually invokes a IoT service (i.e., is_j) at intervals and continuously generates a sequence of QoS data (i.e., $(\ldots, q_{t-1,i,j}, q_{t,i,j})$) [37]. To reduce the scale of historical QoS data, we only employ the historical QoS data of the latest t time slots (i.e., $[q_{1,i,j} \ldots, q_{t-1,i,j}, q_{t,i,j}]$). Therefore, we denote the QoS data in the p-th platform at the t_k-th time slot as a matrix shown in (1), where the i-th row describes the quality value of all n_p services provided in the p-th platform invoked by the i-th user at the t_k-th time slot. Note that if the i-th user does not employ the j-th IoT service at the t_k-th time slot, $q_{t_k,i,j} = 0$.

$$Q_{t_k,p} = \begin{bmatrix} q_{t_k,1,1} & \cdots & q_{t_k,1,n_p} \\ \vdots & \ddots & \vdots \\ q_{t_k,m,1} & \cdots & q_{t_k,m,n_p} \end{bmatrix}. \quad (1)$$

Next, to preserve the user's privacy, we utilize LSH technique to map a user's QoS data to less-sensitive user index. Since classical CF-based recommendation algorithms often employ Pearson correlation coefficient (PCC) as a measure of user similarity, we adopt the LSH function for the PCC distance to realize the transformation.

For each platform sp_p and each time slot t_k, we randomly generate a n_p-dimensional vector $\vec{v}_{t_k,p}$ from the range $[-1, 1]$. For user u_i, the LSH function for time slot t_k is

defined in (2). Here, $Q_{t_k}(u_i) = [Q_{t_k,1}(u_i), \ldots, Q_{t_k,c}(u_i)]$, and $Q_{t_k,p}(u_i) = [q_{t_k,i,1}, \ldots, q_{t_k,i,n_p}]$.

$$h\left(Q_{t_k}(u_i)\right) = \begin{cases} 1, & \text{if } \sum_{p=1}^{c}\left(Q_{t_k,p}(u_i) \circ \overrightarrow{v_{t_k,p}}\right) > 0, \\ 0, & \text{if } \sum_{p=1}^{c}\left(Q_{t_k,p}(u_i) \circ \overrightarrow{v_{t_k,p}}\right) > 0. \end{cases} \tag{2}$$

Thus, according to (2), a user's observed IoT service data at the p-th platform is firstly mapped into a less-sensitive one-dimensional float value; then, all the float values at each platform are accumulated and mapped into a binary value.

Because LSH is a neighbor search method based on probability, one hash function usually leads to less accurate search results. Thus, we adopt amplified LSH through employing multiple hash functions and hash tables. Concretely, for each time slot t_k, we define r hash functions based on r vectors randomly generated from the range [−1, 1]. Afterwards, we obtain a t ∗ r 0-1 matrix, represented as $H(Q(u_i))$ in (3). Indices of all users in U are stored in a hash table.

$$H(Q(u_i)) = \begin{bmatrix} h_{1,1}(Q_1(u_i)) & \cdots & h_{1,r}(Q_1(u_i)) \\ \vdots & \ddots & \vdots \\ h_{t,1}(Q_t(u_i)) & \cdots & h_{t,1}(Q_t(u_i)) \end{bmatrix}. \tag{3}$$

4.2. Step 2: Finding Out Top-K Users Most Similar to u_{target}. In step 1, we have generated a hash table; a user's quality data in recent t time slots is mapped into a t ∗ r 0-1 matrix. For convenience, we denote the matrix (in (3)), as shown in (4), where $\overrightarrow{h_{t_k}(u_i)}$ is a vector referring to the t_k-th row in the matrix. Thus, the hash values of all users' quality data in recent t time slots can be represented by t ∗ m matrix (shown in (5)).

$$HQ(u_i) = \begin{bmatrix} \overrightarrow{h_1(u_i)} \\ \cdots \\ \overrightarrow{h_t(u_i)} \end{bmatrix}, \tag{4}$$

$$H(Q) = \begin{bmatrix} \overrightarrow{h_1(u_1)} & \cdots & \overrightarrow{h_1(u_m)} \\ \vdots & \ddots & \vdots \\ \overrightarrow{h_t(u_1)} & \cdots & \overrightarrow{h_t(u_m)} \end{bmatrix}. \tag{5}$$

Because of the probability-based feature of LSH, it is too strict to find neighbors according to single hash table. Thus, we build L $L > 1$ hash tables through repeating step 1 L times, which can reduce the number of possible "false-negative" neighbors.

Next, we compute the similarities between u_i and the other users at the t_k-th time slot by comparing $\overrightarrow{h_{t_k}(u_i)}$ with the other elements in the t_k-th row in (5). The similarity between u_i and u_j at time slot t_k is denoted as $\mathrm{sim}_{t_k}(u_i, u_j)$. We first initialize $\mathrm{sim}_{t_k}(u_i, u_j)$ to 0. If $\overrightarrow{h_{t_k}(u_i)} = \overrightarrow{h_{t_k}(u_j)}$ holds in any hash table T_p $(1 \le p \le L)$, we increment $\mathrm{sim}_{t_k}(u_i, u_j)$ by one. The similarity between u_i and u_j is calculated with the formula in (6), where ω_{t_k} is a tunable parameter to control the influence of different time slot. The tunable parameters meet the constraint defined in (7).

$$\mathrm{sim}(u_i, u_j) = \sum_{t_k=1}^{t} \mathrm{sim}_{t_k}(u_i, u_j), \tag{6}$$

$$\sum_{t_k=1}^{t} \omega_{t_k} = 1. \tag{7}$$

Finally, the K users with the highest $\mathrm{sim}(\mathrm{u}_i, u_j)$ values are returned as the similar user set of u_i, denoted as SIM_U_Set.

4.3. Step 3: Recommending IoT Service for u_{target} at Time Slot t. In step 3, we have obtained the top-k most similar users of u_{target}, i.e., SIM_U_Set(u_{target}). Next, we recommend the optimal IoT service for u_{target} at time slot t based on SIM_U_Set(u_{target}). In detail, we first predict the quality value of IoT services which have not been employed by u_{target} at time slot t according to formula (8); then, we make recommendation for u_{target} according to the predicted quality value.

$$q_{t,\text{target},j} = \frac{\sum_{u_i \in sIM\_U\_Set\left(u_{\text{target}}\right)} \mathrm{sim}\left(u_{\text{target}}, u_i\right) * q_{t,i,j}}{\sum_{u_i \in sIM\_U\_Set\left(u_{\text{target}}\right)} \mathrm{sim}\left(u_{\text{target}}, u_i\right)}. \tag{8}$$

5. Experiments

To testify the feasibility of our proposed method, extensive experiments are conducted on a real-world QoS dataset, i.e., WS-DREAM [17]. The dataset consists of quality values (i.e., response time and throughput) of 4532 web services invoked by 142 users at 64 different time slots. In our experiments, we consider only one quality dimension, i.e., response time. Moreover, each country that owns services is used to simulate a geographically distributed IoT service platform.

Next, we generate test dataset by randomly removing a proportion of data in the dataset. If P% QoS values have been erased, the sparsity of the test dataset is defined as P %.

Because of the intrinsic nature of LSH, the privacy-preservation capability of our approach is not testified here. Specifically, we testify and compare three evaluation measures.

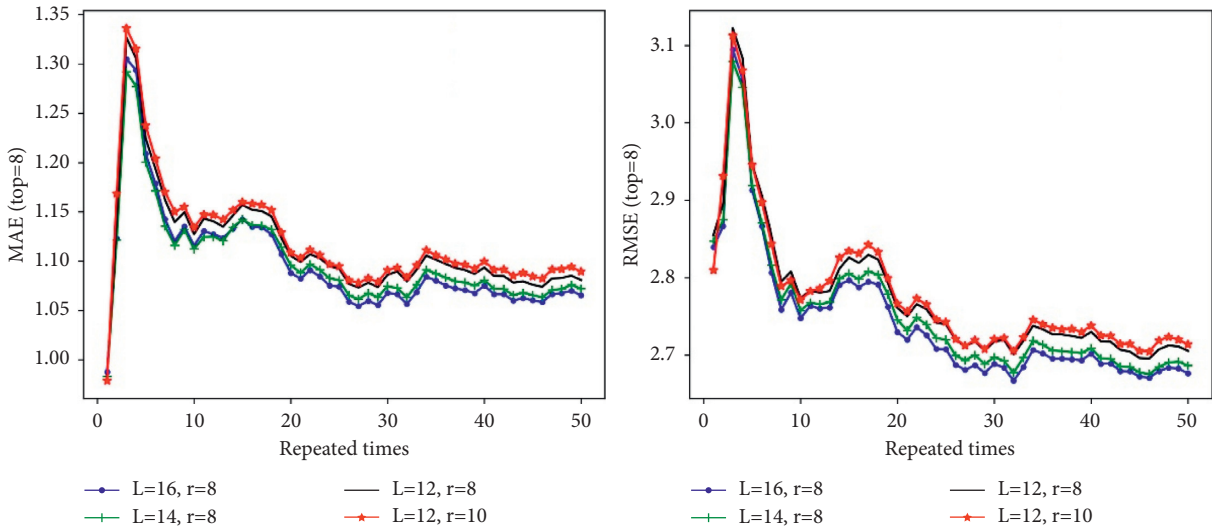


Figure 2: MAE and RMSE w.r.t experiment times.

(1) Time cost: time consumption of our proposed approach, which is employed to measure the efficiency and performance of a recommendation system.

(2) MAE (mean absolute error): average difference between predicted value and real value, which is employed to measure the accuracy of a recommender system.

(3) RMSE (root mean square error): square root of the average of squared error between predicted value and real value, which is also employed to measure the accuracy of a recommendation system.

Concretely, in our experiments, we testify and compare five profiles in our experiments.

5.1. Profile 1: Determining the Best Value of Parameters L and r. In our approach, the number of hash tables (i.e., L) and the number of hash functions in a hash table (i.e., r) can significantly influence the accuracy of a recommender system. Thus, we compare the prediction accuracy (i.e., RMSE and MAE) under different L-r settings and find out the best value of L and r in this profile. For each L-r setting, we repeat our approach on 50 different test datasets and all of the test datasets' sparsity if fixed to 10%. The reason why we choose 50 different test datasets is that we find the average accuracy tends to be convergent after repeated 50 times (as shown in Figure 2).

Table 1 demonstrates the MAE and RMSE values of our algorithm under different combination of L, r, and K when the sparsity of test dataset is set to 10%. Results reveal that our algorithm has the best accuracy when parameters are set as follows: $L = 16$ and $r = 8$. This holds when *top* varies from 5 to 10.

5.2. Profile 2: Finding the Proper Value for Parameter K. In our proposed method, we need to find out a target user's K most similar users. To this end, we conduct experiment with various values of K, where parameters L and r are fixed to 16 and 8, respectively. Therefore, we can find out the proper value of K in our method. Figure 3 shows that, after repeating the experiment on different test datasets 50 times, our method achieves the highest average accuracy (both MAE and RMSE) when top = 8.

5.3. Profile 3: Accuracy Comparison of Three Recommendation Algorithms. To demonstrate the feasibility of our method, we compare IoTSerRec$_{\text{t-LSH}}$ with two state-of-the-art algorithms UPCC [18] and SerRec$_{\text{distr-LSH}}$ [4] within different dataset sparsity. Based on the findings in Profile 1 and Profile 2, parameters in IoTSerRec$_{\text{t-LSH}}$ are defined as follows: $L = 16$, $r = 8$, and $K = 8$. The following observations are made based on the results presented in Figure 4:

(1) The MAE and RMSE values of all three approaches decrease as the dataset sparsity increases. When the sparsity is high, less useful QoS values are fetched, which leads to a less precise prediction.

(2) SerRec$_{\text{distr-LSH}}$ performs much worse than our approach since it does not consider the historical temporal information.

(3) When the dataset sparsity is lower than 50%, the MAE value of our approach is a little lower than UPCC. In spite of this, our algorithm still outperforms other algorithms in most cases. The reason is that the quality values at current time slot are sufficient to find the most similar users for UPCC when the dataset sparsity is low. However, temporal information is necessary within sparser dataset.

5.4. Profile 4: Efficiency Comparison of Three Recommendation Algorithms. In this profile, we make a comparison among the three recommendation algorithms in terms of the efficiency. All of the experiments are performed on a computer with Intel i5 processor and 16.0 GB RAM, which runs

Table 1: Accuracy comparisons of IoTSerRec$_{t\text{-}LSH}$ under different combination of parameters.

Parameter	$K=5$		$K=8$		$K=10$	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
$L=16,\ r=6$	1.0782	2.7286	1.0771	2.6885	1.0787	2.6828
$L=16,\ r=8$	1.0647	2.717	1.0651	2.6762	1.0702	2.6808
$L=16,\ r=10$	1.0835	2.7429	1.0726	2.6912	1.0723	2.682
$L=14,\ r=6$	1.0804	2.7323	1.0727	2.684	1.0783	2.6836
$L=14,\ r=8$	1.0733	2.7297	1.0719	2.6864	1.0771	2.6906
$L=14,\ r=10$	1.0892	2.7558	1.082	2.7033	1.0788	2.6943
$L=12,\ r=6$	1.087	2.7451	1.0816	2.6939	1.0842	2.691
$L=12,\ r=8$	1.0808	2.7411	1.0807	2.7053	1.0824	2.6979
$L=12,\ r=10$	1.0908	2.7573	1.0893	2.7138	1.0934	2.709

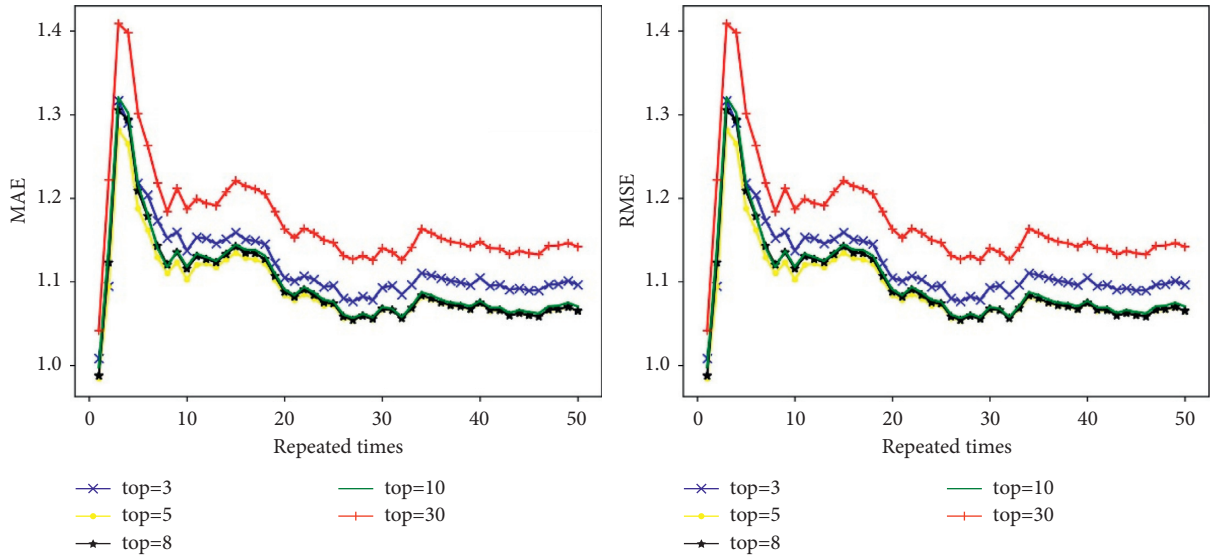


Figure 3: MAE and RMSE of IoTSerRec$_{t\text{-}LSH}$ w.r.t. top.

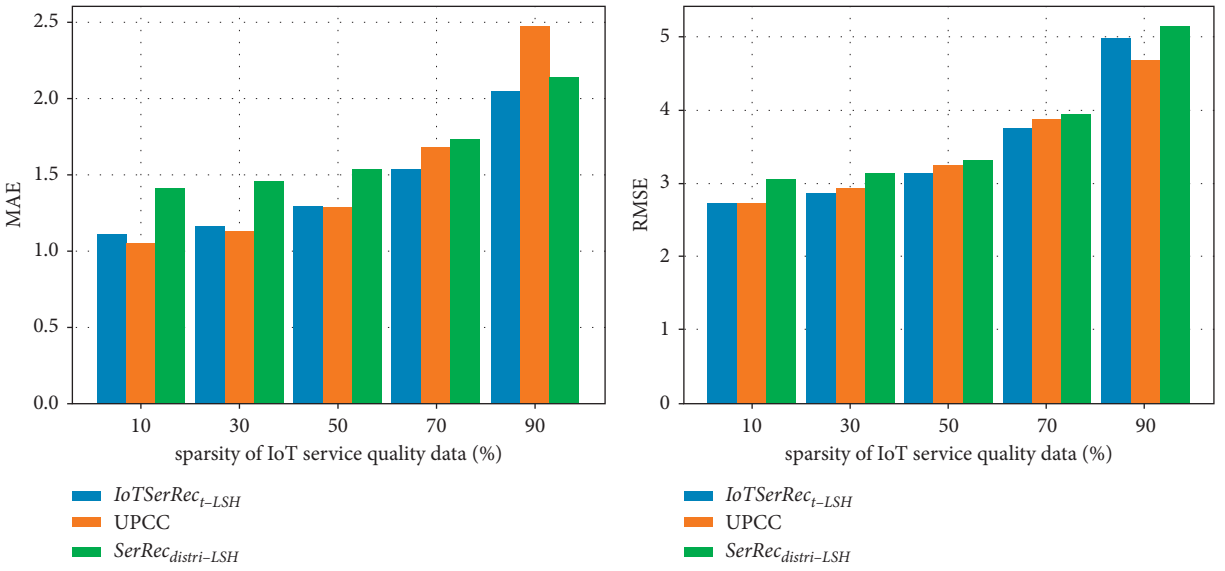


Figure 4: Recommendation accuracy comparison.

Windows 10 and Python 3.6. The experimental results (in Figure 5) show that the time consumption of UPCC is much higher than the other approaches since the UPCC method calculates correlation coefficients online. Furthermore, the time cost of our method is close to that of SerRec$_{distr\text{-}LSH}$ since they both adopt the offline recommendation strategy.

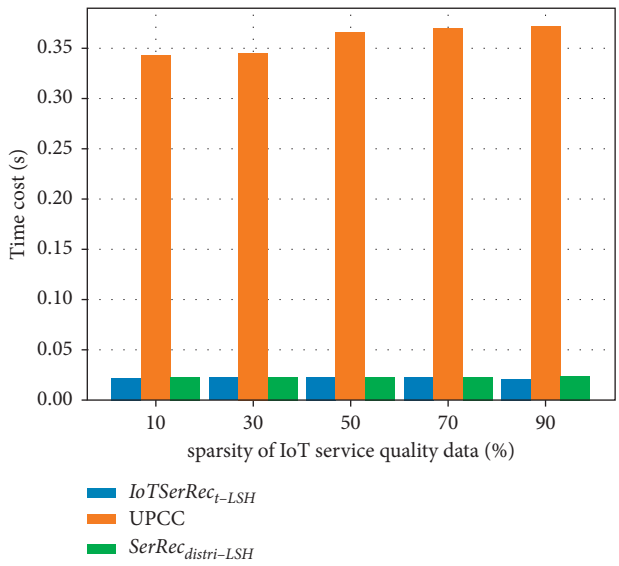


Figure 5: Recommendation efficiency comparison.

6. Conclusion

The number of IoT services is rapidly growing. Recommendation technology can significantly relieve the target users' selection burden. However, because of user privacy concerns, it is impractical for different platforms to directly share their data with each other. In this paper, we proposed a novel cross-platform LSH-based IoT service recommendation method with privacy preservation. Through LSH technique, high dimensionality historical quality data is hashed into less-sensitive indices. Moreover, historical data at different time slots are employed to make more accurate recommendation. Finally, a number of experiments are conducted on real-world dataset WS-DREAM. The experimental results demonstrated the advantage of our approach in terms of accuracy, efficiency, and the capability of privacy preservation.

However, some limitations should be addressed. As we did not consider the periodicity of the quality value of IoT service, the number of time slots is fixed in the current method. Furthermore, spatial information of both users and service providers is not employed in our approach. Therefore, we will further enhance our algorithm by taking more context factors such as periodicity [38, 39] and location [40, 41] into account.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. Yao, X. Wang, Q. Z. Sheng, S. Dustdar, S. Zhang, and S. Dustdar, "Recommendations on the internet of things: requirements, challenges, and directions," *IEEE Internet Computing*, vol. 23, no. 3, pp. 46–54, 2019.

[2] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multi-dimensional mobile service quality prediction and recommendation in distributed fog environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3075849, 8 pages, 2018.

[3] Q. Yu, Z. Zheng, and H. Wang, "Trace norm regularizedmatrix factorization for service recommendation," in *Proceedings of the IEEE International Conference on Web Services*, pp. 34–41, Santa Clara, CA, USA, October 2013.

[4] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2616–2624, 2017.

[5] I. Mashal, T. Y. Chung, and O. Alsaryrah, "Toward service recommendation in Internet of Things," in *Proceedings of the the Seventh International Conference on Ubiquitous and Future Networks*, pp. 328–331, Sapporo, Japan, July 2015.

[6] L. Yao, Q. Z. Sheng, A. H. H. Ngu, and X. Li, "Things of interest recommendation by leveraging heterogeneous relations in the Internet of Things," *ACM Transactions on Internet Technology*, vol. 16, no. 2, pp. 1–25, 2016.

[7] L. Yao, Q. Z. Sheng, A. H. H. Ngu, H. Ashman, and X. Li, "Exploring recommendations in internet of things," in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 855–858, Gold Coast Queensland, Australia, July 2014.

[8] A. Forestiero, "Multi-agent recommendation system in internet of things," in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 772–775, Madrid, Spain, May 2017.

[9] I. Mashal, O. Alsaryrah, and T.-Y. Chung, "Testing and evaluating recommendation algorithms in Internet of Things," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 6, pp. 889–900, 2016.

[10] X. Wang, J. Zhu, Z. Zheng, W. Song, Y. Shen, and M. R. Lyu, "A spatial-temporal qos prediction approach for time-aware web service recommendation," *ACM Transactions on the Web*, vol. 10, no. 1, pp. 1–25, 2016.

[11] Y. Zhang, Z. Zheng, and M. R. Lyu, "WSPred: a time-aware personalized QoS prediction framework for web services," in *Proceedings of the IEEE 22nd International Symposium on Software Reliability Engineering*, pp. 210–219, Hiroshima, Japan, December 2011.

[12] Y. Zhong, Y. Fan, K. Huang, W. Tan, and J. Zhang, "Time-aware service recommendation for mashup creation," *IEEE Transactions on Services Computing*, vol. 8, no. 3, pp. 356–368, 2015.

[13] C. Yu and L. Huang, "A web service QoS prediction approach based on time- and location-aware collaborative filtering," *Service Oriented Computing and Applications*, vol. 10, no. 2, pp. 135–149, 2016.

[14] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[15] 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.

[16] B. S. Jena, C. Khan, and R. Sunderraman, "High performance frequent subgraph mining on transaction datasets: a survey

and performance comparison," *Big Data Mining and Analytics*, vol. 2, no. 3, pp. 159–180, 2019.

[17] 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.

[18] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the 4th Conference on Uncertainty in Artificial Intelligence,*, pp. 43–52, Madison, Wisconsin, July 1998.

[19] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1145–1153, Article ID 2969489, 2021.

[20] T. Ma, Y. Zhang, J. Cao et al., "KDVEM: a k-degree anonymity with vertex and edge modification algorithm," *Computing*, vol. 70, no. 6, pp. 1336–1344, 2015.

[21] W. Dou, X. Zhang, J. Liu, and J. Chen, "HireSome-II: towards privacy-aware cross-cloud service composition for big data applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 455–466, 2015.

[22] Y. Yu, M. Li, L. Liu, Y. Li, and J. Wang, "Clinical big data and deep learning: applications, challenges, and future outlooks," *Big Data Mining and Analytics*, vol. 2, no. 4, pp. 288–305, 2019.

[23] Y. Hu, Q. Peng, X. Hu, and R. Yang, "Time aware and data sparsity tolerant web service recommendation based on improved collaborative filtering," *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 782–794, 2015.

[24] G. Li, S. Peng, C. Wang, J. Niu, and Y. Yuan, "An energy-efficient data collection scheme using denoising autoencoder in wireless sensor networks," *Tsinghua Science and Technology*, vol. 24, no. 1, pp. 86–96, 2019.

[25] L. Liu, X. Chen, Z. Lu, L. Wang, and X. Wen, "Mobile-Edge computing framework with data compression for wireless network in energy internet," *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 271–280, 2019.

[26] X. Zhou, X. Xu, W. Liang et al., "Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1377–1386, 2022.

[27] Y. Zhang, J. Pan, L. Qi, and Q. He, "Privacy-preserving quality prediction for edge-based IoT services," *Future Generation Computer Systems*, vol. 114, pp. 336–348, 2020.

[28] Y. Liu, A. Pei, F. Wang et al., "An attention-based category-aware GRU model for the next POI recommendation," *International Journal of Intelligent Systems*, vol. 36, no. 7, pp. 3174–3189, 2021.

[29] Q. Zhu, X. Ma, and X. Li, "Statistical learning for semantic parsing: a survey," *Big Data Mining and Analytics*, vol. 2, no. 4, pp. 217–239, 2019.

[30] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, "Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5790–5798, 2021.

[31] D. Kim, J. Son, D. Seo, Y. Kim, H. Kim, and J. T. Seo, "A novel transparent and auditable fog-assisted cloud storage with compensation mechanism," *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 28–43, 2020.

[32] L. Qi, C. Hu, X. Zhang et al., "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, p. 1, 2020.

[33] X. Zhou, Y. Li, and W. Liang, "CNN-RNN based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, 2021.

[34] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5213–5222, 2021.

[35] S. Kumar and M. Singh, "A novel clustering technique for efficient clustering of big data in hadoop ecosystem," *Big Data Mining and Analytics*, vol. 2, no. 4, pp. 240–247, 2019.

[36] Y. Huang, Y. Chai, Y. Liu, and J. Shen, "Architecture of next-generation E-commerce platform," *Tsinghua Science and Technology*, vol. 24, no. 1, pp. 18–29, 2019.

[37] L. Qi, H. Song, X. Zhang, G. Srivastava, X. Xu, and S. Yu, "Compatibility-aware web API recommendation for mashup creation via textual description mining," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 1s, pp. 1–19, 2021.

[38] X. Xu, B. Shen, S. Ding et al., "Service offloading with deep Q-network for digital twinning-empowered internet of vehicles in edge computing," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1414–1423, 2022.

[39] X. Xu, R. Mo, X. Yin et al., "PDM: privacy-aware deployment of machine-learning applications for industrial cyber-physical cloud systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5819–5828, 2021.

[40] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3469–3477, 2021.

[41] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 9126833, 17 pages
https://doi.org/10.1155/2021/9126833

Research Article

A Graph Optimization-Based Acoustic SLAM Edge Computing System Offering Centimeter-Level Mapping Services with Reflector Recognition Capability

Zou Zhou,[1] **Guoli Zhang**,[1] **Fei Zheng**,[1] **Tuyang Wang**,[2] **Longjie Chen**,[1] **and Nan Duan**[1]

[1]*Ministry of Education Key Laboratory of Cognitive Radio and Information Processing, Guilin University of Electronic Technology, Guilin 541004, China*
[2]*National Demonstration Center for Experimental Electronic Circuit Education, Guilin University of Electronic Technology, Guilin 541004, China*

Correspondence should be addressed to Fei Zheng; zhengfei@guet.edu.cn and Tuyang Wang; gdwty@foxmail.com

Received 4 September 2021; Revised 4 October 2021; Accepted 2 November 2021; Published 3 December 2021

Academic Editor: Xuyun Zhang

Robots can use echo signals for simultaneous localization and mapping (SLAM) services in unknown environments where its own camera is not available. In current acoustic SLAM solutions, the time of arrival (TOA) in the room impulse response (RIR) needs to be associated with the corresponding reflected wall, which leads to an echo labelling problem (ELP). The position of the wall can be derived from the TOA associated with the wall, but most of the current solutions ignore the effect of the cumulative error in the robot's moving state measurement on the wall position estimation. In addition, the estimated room map contains only the shape information of the room and lacks position information such as the positions of doors and windows. To address the above problems, this paper proposes a graph optimization-based acoustic SLAM edge computing system offering centimeter-level mapping services with reflector recognition capability. In this paper, a robot equipped with a sound source and a four-channel microphone array travels around the room, and it can collect the room impulse response at different positions of the room and extract the RIR cepstrum feature from the room impulse response. The ELP is solved by using the RIR cepstrum to identify reflectors with different absorption coefficients. Then, the similarity of the RIR cepstrum vectors is used for closed-loop detection. Finally, this paper proposes a method to eliminate the cumulative error of robot movement by fusing IMU data and acoustic echo data using graph-optimized edge computation. The experiments show that the acoustic SLAM system in this paper can accurately estimate the trajectory of the robot and the position of doors, windows, and so on in the room map. The average self-localization error of the robot is 2.84 cm, and the mapping error is 4.86 cm, which meet the requirement of centimeter-level map service.

1. Introduction

With the arrival of intelligent society, mobile robots have been widely used in people's life and work, which greatly facilitates people's life and work. In the unknown indoor space, robots realize positioning and navigation services need to know the surrounding environment map and their own position in the map. However, the robot does not know the indoor map information. Simultaneous Localization and Mapping (SLAM) is the service of detecting and sensing the map (contour) of the surrounding environment for a moving subject in an unknown environment, relying only on the mounted sensors, while determining its own position in the map [1]. After decades of continuous development, SLAM technology services have been widely used in mobile robotics [2], virtual/augmented reality [3], autonomous driving [4], and so on. The current mainstream SLAM techniques are classified based on the differences in the sensors used and can be divided into LIDAR SLAM techniques and visual SLAM techniques. Sensors such as LIDAR and cameras have the advantage of accuracy and high resolution but they also have disadvantages: LIDAR is a very

expensive sensor and poses health and safety issues in operation [5]. Cameras, although cost getting lower, require high processing power in low-light environments as well as low signal-to-noise ratios [6]. In addition, the above systems are computationally complex and usually use cloud-based processing, which is costly and involves privacy and security. In contrast, acoustic sensors as the standard for mobile robots can be used in low-light and dark environments [7]. Map reconstruction work can be achieved using the robot's own arithmetic power, which not only has significant cost advantages but also offloads privacy-aware services to MEC (mobile edge computing), avoiding the leakage of private information such as indoor images. Therefore, researchers have begun to explore the implementation of acoustic SLAM.

In indoor environments, the propagation of acoustic signals is obscured and reflected by buildings resulting in multipath effects, which to some extent reflect information about the room arrangement and geometry and can be used to estimate environmental maps. Researchers have started to estimate room shapes from the room impulse response (RIR) of acoustics. Labelling the first-order echoes in the RIR to the walls that generate them is the key for room shape estimation. Since the RIR itself does not specify which reflections come from which walls, there is an echo labelling problem (ELP) [8]. Literatures [9, 10] solved the ELP using the properties of the Euclidean distance matrix (EDM), but the algorithm must traverse the TOA combinations of all echoes, which has a high computational complexity. References [8, 11, 12] improved the computational complexity of their work based on EDM using graph theory, subspace filtering, and greedy iteration, respectively, but the overall computational complexity is still large. There is also a method of solving ELP using elliptic constraints. Literatures [13, 14] solved the series of reflective points of walls based on an elliptic constraint model and finally used the Hough transform for the estimation of each reflective wall. This method needs to arrange many anchor nodes to obtain enough data, which is complex to implement and extremely computationally intensive. Literature [15] proposed an algorithm to reduce the computational complexity based on elliptic constraints for iterative echo marking. The above method uses a stationary distributed microphone array, which requires the sound source and microphone array to be arranged in the room in advance and is not applicable to the practical application scenario of SLAM.

In addition to the static deployment of sources and microphones scheme in the above work, there is another scheme that embeds acoustic sensors on a mobile robot, which is more in line with the practical needs of SLAM and is more relevant for research. Whether the robot is equipped with an acoustic source can be classified as active acoustic SLAM and passive acoustic SLAM. References [16, 17] used robots equipped with multichannel microphones for their own localization as well as localization of acoustic sources by sensing the ambient sound sources around them. However, suitable sound sources that can be detected are not always available in the actual environment. Another option is to use a robot equipped with both sound sources and microphones for simultaneous localization and mapping. In literatures [18–20], a mobile robot with a juxtaposition of an acoustic source and a single-channel microphone was used to collect first-order echoes. Due to the weak spatial perception of the single-channel microphone, the robot moved at least three times for a reflective wall estimation, and the movement error may affect the accuracy of the reflective wall estimation. Multichannel microphones have better spatial perception capability than single microphones and can obtain more information about the interior geometry for the same number of measurements. Literature [21] achieved room shape estimation using a robot equipped with an acoustic source and a four-channel microphone by estimating the location of the first-order image source by clustering, but the robot needs to collect a large amount of RIR data at each movement. All the above acoustic SLAM schemes can achieve the estimation of room shape but ignore the effect of cumulative errors in the measurement of the robot's moving state, that is, on the reconstruction of the room contour. The room map has only room shape information and lacks the location information of doors and windows because it cannot distinguish between different reflective materials.

In this paper, a robot active sounding scheme equipped with both sound sources and microphone arrays is used for simultaneous localization and mapping. To address the ELP problem mentioned above and the cumulative error of robot movement measurement affecting the accuracy of map building, this paper proposes a graph-optimized acoustic SLAM edge computing system based on graph optimization that can identify room detail information. The main contributions of this paper are as follows:

(1) A robot system prototype is designed and implemented that can be used for acoustic SLAM

(2) A first-order echo labelling algorithm based on RIR cepstrum is proposed, which solves ELP by distinguishing different reflective materials

(3) A graph optimization-based method is proposed for correcting the pose estimated by the trapezoidal constraint

The sections of this paper are organized as follows. Section 1 introduces the background and current research status of acoustic SLAM, and Section 2 describes the problem setup and the architecture of the graph optimization-based acoustic SLAM system. In Section 3, a prototype of our designed robotic system for acoustic SLAM is presented. Section 4 introduces the related acoustic SLAM methods. Section 5 shows the simulations and experiments, and Section 5 concludes.

2. Problem Description and System Structure

In an indoor environment, the sound signal received by a microphone consists of the direct sound from the source and the reflected sound that is reflected by the walls. In the image source model [22, 23], the reflected sound from the actual sound source was replaced by the direct sound from the image source, as shown in Figure 1(a). For a first-order echo

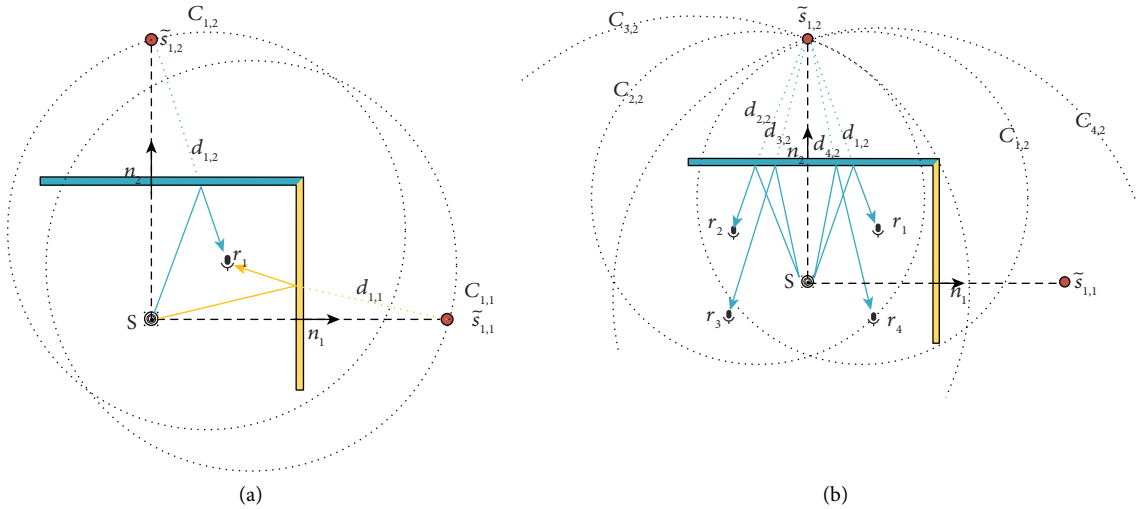


FIGURE 1: Schematic diagram of the first-order image sound source: (a) single source with a single microphone and (b) single source with a multichannel microphone.

described by the unit normal n_k and an arbitrary wall point p_k and the kth reflected wall, the first-order image source $\tilde{s}_{1,k}$ of the real sound source s with microphone r_j is calculated according to $\tilde{s}_{1,k} = r_j + 2\langle p_k - r_j, n_k\rangle n_k$. The sound propagation process was described in terms of the room impulse response (RIR), which consists of a series of Dirac pulses δ:

$$h_j(t) = \sum_i \alpha_i \delta(t - \tau_i) + \varepsilon(t), \tag{1}$$

where $\varepsilon(t)$ is the noise, α_i is the amplitude of the received pulse, and its magnitude depends on the absorption coefficient of the wall and the distance from the image source to the microphone [18]. τ_i is the arrival time of the corresponding pulse, which is proportional to the distance from the image source to the microphone r_j. The room impulse response can be represented as a dataset $\text{data}_j = \{(\alpha_i, \tau_i), i = 1, 2, \ldots, n\}$. ELP finds the data (α_i, τ_i) associated with the first-order image source $\tilde{s}_{1,k}$ from the dataset data_j. The label_i was defined as the label corresponding to data (α_i, τ_i). The data association process for a first-order image source $\tilde{s}_{1,k}$ can be represented by the function $L(\alpha_i, \tau_i)$:

$$L(\alpha_i, \tau_i) = \begin{cases} 1, & \text{label}_i = k, \\ 0, & \text{else}. \end{cases} \tag{2}$$

The distance from the first-order image source $\tilde{s}_{1,k}$ to the microphone r_j can be known from the marked first-order echo data. For a single-channel microphone, as shown in Figure 1(a), the position of the image source $\tilde{s}_{1,k}$ is on a circle $C_{1,k}$ with the position of microphone r_1 as the center and $d_{j,k}$ as the radius, and the exact position of the image source $\tilde{s}_{1,k}$ on the circle cannot be determined due to the lack of spatial information. For multichannel microphones, as shown in Figure 1(b), the position of the image source $\tilde{s}_{1,k}$ is at the intersection point between circles $C_{j,k}$. In 2D space, it is known from the TOA localization algorithm [24] that the uniqueness of the image source s location can be guaranteed when the number of microphones is greater than 3. The midpoint of the line connecting the real source s and the image source $\tilde{s}_{1,k}$ is the location of the reflecting wall.

In this paper, an omnidirectional sound source was defined, and an omnidirectional 4-channel microphone array are installed on the robot, and the location of the microphone array in relation to the sound source is shown in Figure 2(a). The robot travels around the room in a circle, and for each step the robot takes, the sound source generates a pulse while the microphone array records an echo. The room was defined as a 2D polygon for the sake of descriptive simplicity, and the approach in this paper can be easily extended to 3D.

The robot can estimate the position of each wall relative to itself in the room using echo information, as shown in Figure 2(b). Based on the above description, the difficulty of first-order image source position estimation is solving the echo labelling problem (ELP) [8]. In this paper, taking advantage of the strong spatial perception of multichannel microphones, an RIR cepstrum feature that can distinguish reflective walls with different absorption coefficients was proposed, and based on this feature, this paper proposes a solution for first-order echo labelling based on the RIR cepstrum.

The robot travels around the room in a circle and uses the echo information from different locations to estimate the distance from the wall to itself for the shape estimation of the indoor room, as shown in Figure 2(b). Since there is a cumulative error in the robot position estimated by IMU data, this can lead to inaccurate estimation of the wall position. To solve this problem, a graph optimization-based acoustic SLAM method was proposed, which uses graph optimization to fuse the robot pose estimated by the sound echo signal with the pose estimated by IMU to eliminate the cumulative error, and the system block diagram of this method is shown in Figure 3. Referring to other graph optimization structures [25–27], the graph optimization system in this paper is also mainly divided into two parts: front end and back end. The front end establishes the graph

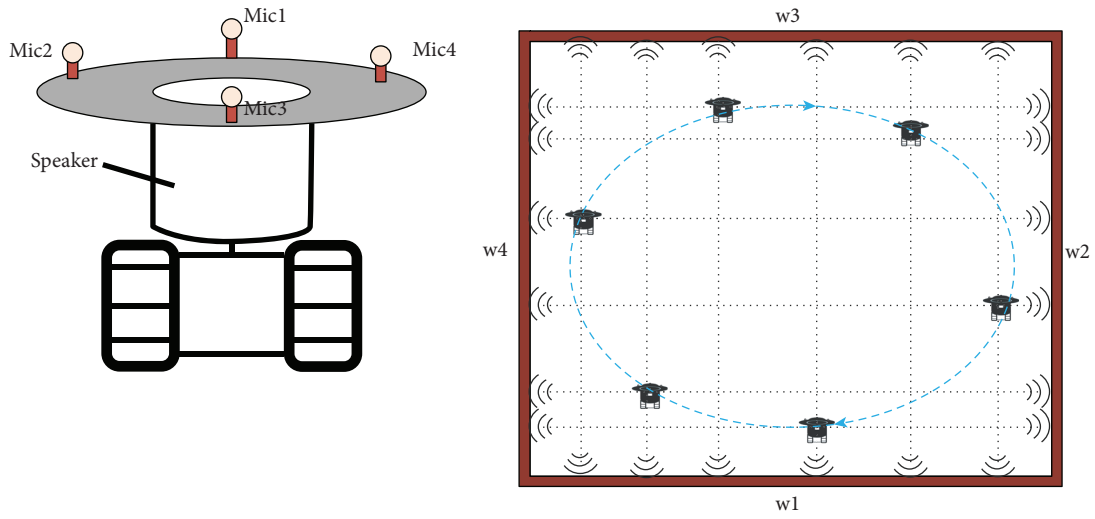


Figure 2: (a) Layout of the sound source and microphone on the robot. (b) Schematic diagram of the robot moving in the room.

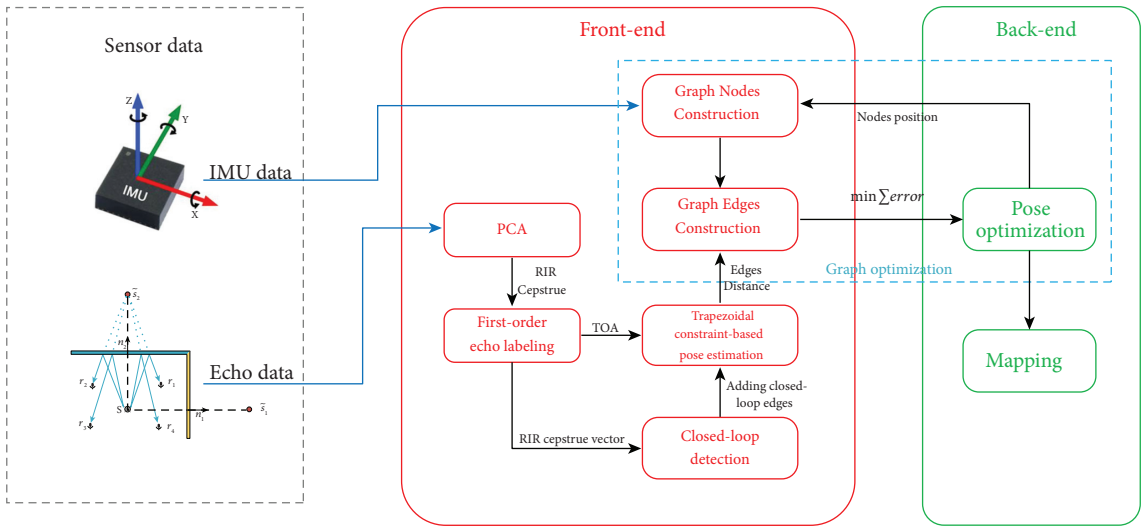


Figure 3: Framework of an acoustic SLAM system based on graph optimization.

vertices and the positional constraint relations between graph vertices based on the IMU sensor and acoustic sensor data, and the back end optimizes the positional graph based on the closed-loop constraints added by the closed-loop detection and the constraint relations between graph vertices to finally obtain globally consistent robot trajectories and indoor room maps.

3. Acoustic SLAM Method

According to the system framework of acoustic SLAM, the acoustic SLAM method can be divided into two parts: echo labelling and pose correction. Echo labelling extracts the first-order echo signal from the echo signal and then estimates the position of the first-order image sound source based on the first-order echo signal. The pose correction is to eliminate the accumulated errors during the robot movement globally using graph optimization methods.

3.1. Echo Labelling Based on the RIR Cepstrum Feature. The ELP problem is often solved using the Euclidean distance matrix approach, which requires traversing all possible combinations of echoes with high complexity. In this section, an echo labelling method based on RIR cepstrum is proposed, which can achieve fast and accurate echo labelling.

3.1.1. RIR Cepstrum. Multichannel microphones are more spatially aware than single-channel microphones, and a spatial cepstrum feature is proposed in literature [28], which

can represent the relative position of the sound source in the room. Inspired by this, the room impulse response cepstrum feature was proposed, which can be used for first-order echo labelling and loopback detection.

Suppose the robot is equipped with M omnidirectional microphones and an omnidirectional sound source s. As shown in Figure 4, the robot moves N steps from x_1 to x_N, and each time it moves, the robot's sound source generates a pulse, while the microphone acquires the room impulse response at the current position. The robot can acquire M room impulse responses $h_{i,j}(n)$, $j = 1, 2, \ldots, M$ at x_i. According to equation (1), $h_{i,j}(n)$ consists of a series of pulses, and the average energy feature $r_{i,j,k}$ of the kth pulse is extracted:

$$r_{i,j,k} = \sqrt{\frac{1}{2L+1} \sum_{n=\tau_{i,j,k}-L}^{\tau_{i,j,k}+L} h_{i,j}(n)^2}, \tag{3}$$

where $\tau_{i,j,k}$ is the TOA value of the kth pulse in $h_{i,j}(n)$ and $2L+1$ is the width of the rectangular window.

The time delay feature of the kth pulse in $h_{i,j}(n)$ is

$$t_{i,j,k} = \tau_{i,j,k} \cdot c, \tag{4}$$

where c is the speed of sound propagation in the air.

Log operations are performed on the above two features separately to obtain the log energy vector $p_{i,k}$ and the log time delay vector $q_{i,k}$:

$$\begin{aligned} p_{i,k} &= \left(\log\left(r_{i,1,k}\right) \ \log\left(r_{i,2,k}\right) \ \cdots \ \log\left(r_{i,M,k}\right) \right)^T, \\ q_{i,k} &= \left(\log\left(t_{i,1,k}\right) \ \log\left(t_{i,2,k}\right) \ \cdots \ \log\left(t_{i,M,k}\right) \right)^T. \end{aligned} \tag{5}$$

The robot is obtained from $x_1 \longrightarrow x_N, N > M$; the matrix of the average amplitude logarithm of the impulse response about the room A_k and the matrix of the logarithm of the arrival distance B_k can be obtained as follows:

$$\begin{aligned} A_k &= \left(p_{1,k} \ \ p_{2,k} \ \cdots \ p_{n,k} \right)^T, \\ B_k &= \left(q_{1,k} \ \ q_{2,k} \ \cdots \ q_{n,k} \right)^T. \end{aligned} \tag{6}$$

As in the method for extracting the spatial cepstrum [29], we also use PCA instead of DFT or DCT. R_{A_k} can be obtained from A_k.

$$R_{A_k} = A_k A_k^T. \tag{7}$$

Since R_{A_k} is a symmetric matrix, R_{A_k} the eigenvalue decomposition can be expressed as follows:

$$R_{A_k} = E_{A_k} D_{A_k} E_{A_k}^T, \tag{8}$$

where E_{A_k} is the eigenvector matrix, D_{A_k} is the diagonal matrix, and the diagonal elements are the eigenvalues, in descending order.

After PCA dimensionality reduction, the data d_{A_k} can be expressed as

$$d_{A_k} = E_{A_k}\left(A_k - \overline{A_k}\right). \tag{9}$$

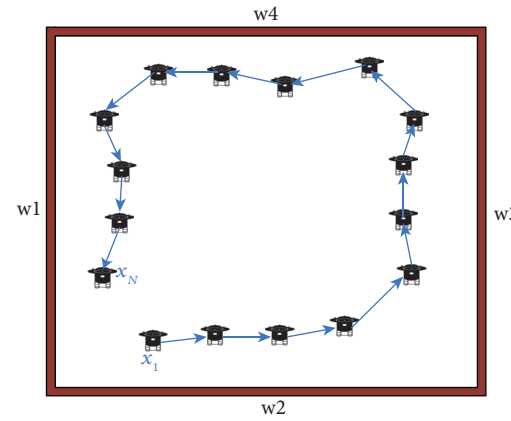


Figure 4: Schematic diagram of robot movement.

Similarly, for the matrix B_k, PCA dimensionality reduction is performed.

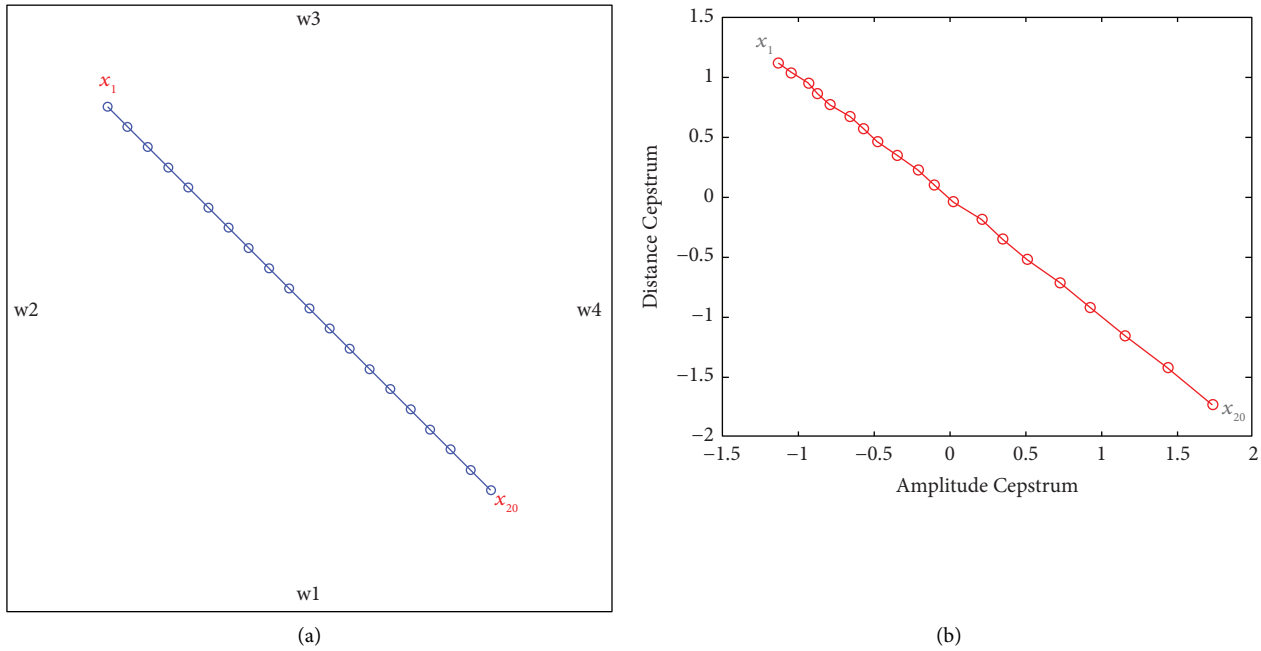
$$d_{B_k} = E_{B_k}\left(B_k - \overline{B_k}\right). \tag{10}$$

The principal component components are selected from d_{A_k} and d_{B_k}, and they are the components with the largest eigenvalues d_{a_k} and d_{b_k}, which form the room impulse response cepstrum d_{h_k}.

$$d_{h_k} = \left[d_{a_k} \ \ d_{b_k} \right], \tag{11}$$

where d_{h_k} is the matrix, d_{h_k} is defined as the room impulse response cepstrum, d_{a_k} is the amplitude cepstrum, and d_{b_k} is the distance cepstrum.

The amplitude cepstrum corresponds to the average amplitude of the pulses observed by the microphone array. When the pulses observed by the microphone array are consistent, i.e., the first-order echoes come from the same reflecting wall, the magnitude of the amplitude cepstrum is inversely proportional to the distance of the robot from the reflecting wall. Similarly, when the pulses observed by the microphone array are consistent, the magnitude of the distance cepstrum is proportional to the distance of the robot from the reflecting wall. As shown in Figure 5(a), in a 6 m * 6 m rectangular room, the robot moves from x_1 to x_{20}, and to ensure the consistent observation of the microphone matrix, the reflection coefficients of walls w1–w4 to 0.8, 0, 0, and 0 were defined. At this time, the robot can only receive the echo from wall w1. We select the second pulse in the room impulse response and extract the RIR cepstrum d_{h_2} according to the above method and represent the RIR cepstrum in a two-dimensional Cartesian coordinate system, as shown in Figure 5(b). We can observe that when the pulses observed by the microphone matrix are consistent, the cepstrum of the room impulse response d_{h_k} of d_{a_k} and d_{b_k} approximates a linear relationship. When the value of the RIR cepstrum of the microphone array pulse combination is in the vicinity of the straight line corresponding to the reflective wall and then combined with the size of the microphone array, it can be determined whether the microphone array is observed consistently.



(a) (b)

FIGURE 5: (a) Schematic diagram of the robot's trajectory in a 6 m ∗ 6 m room; (b) the RIR cepstrum d_{h_2} mapping schematic in the two-dimensional coordinate system.

3.1.2. Echo Labelling Based on the RIR Cepstrum Feature. Based on the feature that the one-dimensional component of the RIR cepstrum approximately satisfies a linear relationship with the two-dimensional component when the RIR cepstrum is observed consistently, a method for first-order echo labelling was proposed. According to the image source model, it is known that the 2nd and 3rd pulses received by the microphone must be the first-order echoes reflected from the wall. The robot follows the trajectory in Figure 4 from x_1 moving to x_N. Since the robot moves against the wall and the spacing between the microphones is small, it can be guaranteed that the 2nd and 3rd pulses observed by the microphones are mostly from the same virtual source, i.e., the observations are consistent. Assuming that the acoustic reflection coefficients of each of the four walls of the room in Figure 4 are different, the robot takes the 2nd and 3rd pulses from $x_1 \longrightarrow x_N$. The second and third pulses received by the microphone are taken to find the cepstrum of the room impulse response; the matrix of the logarithm of the room impulse response amplitude at this time $A_{2,3}$ and the matrix of the logarithm of the arrival distance $B_{2,3}$ are as follows:

$$A_{2,3} = \begin{bmatrix} A_2 \\ A_3 \end{bmatrix}, \quad B_{2,3} = \begin{bmatrix} B_2 \\ B_3 \end{bmatrix}. \tag{12}$$

Following the method in the previous section, PCA operations are performed on $A_{2,3}$ and $B_{2,3}$, to obtain the feature matrices $E_{A_{2,3}}$ and $E_{B_{2,3}}$, respectively. The RIR cepstrum after PCA dimensionality reduction is

$$d_{h_{2,3}} = \left[d_{a_{2,3}} \;\; d_{b_{2,3}} \right]. \tag{13}$$

According to the characteristics of the RIR cepstrum feature, $d_{h_{2,3}}$ can be fitted as a straight line l_i corresponding to the reflecting wall i with different reflection coefficients.

$$a_i x + b_i y + c_i = 0. \tag{14}$$

When the kth pulse observed by the microphone matrix is greater than 3, it is not possible to determine whether the kth pulse observed at this time is a first-order echo. The first-order echo candidates from wall i can be obtained from the TOA values of the first and second echoes and the relationship between the microphone positions. These candidates can be combined to obtain a new combination of room pulses, which corresponds to the RIR cepstrum d_τ as follows:

$$d_\tau = \left[d_{a,\tau} \;\; d_{b,\tau} \right]. \tag{15}$$

If the new combination is a first-order echo from wall i, the corresponding room impulse response cepstrum d_τ should be near the straight line l_i, and the distance from d_τ to the straight line l_i satisfies the following equation:

$$D_{i,\tau} = \frac{\left| a_i d_{a,\tau} + b_i d_{b,\tau} + c_i \right|}{\sqrt[2]{a_i^2 + b_i^2}} < \Delta\varepsilon, \tag{16}$$

where $\Delta\varepsilon$ is the Euclidean distance threshold. Following the above method, the first-order echoes of different walls can be distinguished, and thus, the location of the image source can be estimated.

3.2. Pose Correction Based on Graph Optimization. The pose correction method consists of three parts: closed-loop

detection, pose estimation, and pose correction. In this paper, the pose at different locations of the robot is used as nodes of the graph. The constraint relationship between graph nodes is established using closed-loop detection and pose estimation. Finally, the graph optimization method is used to correct the robot's pose.

3.2.1. Closed-Loop Detection. Closed-loop detection determines whether the robot has reached the previous position, and it is extremely important for back-end optimization. After the above echo labelling method, the robot at the x_i location can obtain the RIR cepstrum consistent with the k-sided wall observation, and these cepstrums can be combined into a 2k-dimensional RIR cepstrum vector X_i as follows:

$$X_i = [\, x_{i1} \;\; y_{i1} \;\; x_{i2} \;\; y_{i2} \;\; \dots \;\; x_{ik} \;\; y_{ik} \;\;], \tag{17}$$

where $[\, x_{i,k} \;\; y_{i,k} \,]$ is the RIR cepstrum from wall k.

Similarly, the robot can obtain RIR cepstrum vector X_j at x_j as follows:

$$X_j = \left[\, x_{j1} \;\; y_{j1} \;\; x_{j2} \;\; y_{j2} \;\; \dots \;\; x_{jk} \;\; y_{jk} \;\; \right]. \tag{18}$$

The vectors X_i and X_j can be used to express the Euclidean distance between them to express the similarity of their spaces. When x_i and x_j are in the same position or close to each other, the Euclidean distance between the two vectors should satisfy the following formula:

$$\text{distance}_{ij} = \left| X_j - X_i \right| < \delta, \tag{19}$$

where δ is the Euclidean distance threshold.

3.2.2. Trapezoidal Constraint-Based Pose Estimation. In indoor space, the actual position of the robot and the position movement of the image source satisfy the isosceles trapezoidal constraint [18, 29], as shown in Figure 6. Based on this constraint, a robot positional estimation method was proposed.

In world coordinates, let the robot's pose at x_{i-1} be $X_{i-1} = (x_{i-1}, y_{i-1}, \theta_{i-1})$ and the robot's pose at x_i be $X_i = (x_i, y_i, \theta_i)$. The change in pose dX_i of the robot moving from x_{i-1} to x_i is

$$dX_i = X_i - X_{i-1} = (x_i - x_{i-1}, y_i - y_{i-1}, \theta_i - \theta_{i-1}), \tag{20}$$

where dX_i is expressed in polar coordinates as

$$dX_i = (r \cos \alpha, r \sin \alpha, \theta). \tag{21}$$

In equation (21), r is the displacement variable, α is the angle of the displacement direction to the X-axis of the world coordinate system, and θ is the rotation angle of the robot coordinate system.

The coordinates of the image source at x_{i-1} and x_i are expressed in polar coordinates in the robot coordinate system, respectively, as follows:

$$\begin{cases} \widetilde{x}_{i-1,1} = \left(r_{i-1,1}, \theta_{i-1,1}\right), \\ \widetilde{x}_{i-1,2} = \left(r_{i-1,2}, \theta_{i-1,2}\right), \\ \widetilde{x}_{i-1,3} = \left(r_{i-1,3}, \theta_{i-1,3}\right), \\ \widetilde{x}_{i-1,4} = \left(r_{i-1,4}, \theta_{i-1,4}\right), \end{cases} \longrightarrow \begin{cases} \widetilde{x}_{i,1} = \left(r_{i,1}, \theta_{i,1}\right), \\ \widetilde{x}_{i,2} = \left(r_{i,2}, \theta_{i,2}\right), \\ \widetilde{x}_{i,3} = \left(r_{i,3}, \theta_{i,3}\right), \\ \widetilde{x}_{i,4} = \left(r_{i,4}, \theta_{i,4}\right). \end{cases} \tag{22}$$

The robot rotation angle θ is related only to the angle of the polar coordinates of the image source.

$$\theta = \theta_{i,1} - \theta_{i-1,1} = \theta_{i,2} - \theta_{i-1,2} = \theta_{i,3} - \theta_{i-1,3} = \theta_{i,4} - \theta_{i-1,4}. \tag{23}$$

The estimated value of the robot rotation angle is

$$\widehat{\theta} = \frac{1}{4} \sum_{w=1}^{w=4} \theta_{i,w} - \theta_{i-1,w}. \tag{24}$$

Robots from x_{i-1} move to x_i, and the length of the image acoustic source polar coordinates changes as follows:

$$\begin{cases} r_{i,1} = r_{i-1,1} + 2r \, \sin\left(\alpha + \alpha_{i-1,1}\right), \\ r_{i,2} = r_{i-1,2} + 2r \, \sin\left(\alpha + \alpha_{i-1,2}\right), \\ r_{i,3} = r_{i-1,3} + 2r \, \sin\left(\alpha + \alpha_{i-1,3}\right), \\ r_{i,4} = r_{i-1,4} + 2r \, \sin\left(\alpha + \alpha_{i-1,4}\right), \end{cases} \tag{25}$$

where $\alpha_{i-1,n} = \theta_{i-1,1} - \theta_{i-1,n}, \quad n = 1, 2, 3, 4.$

Let $s = (r, \alpha)$, the following vector function can be obtained:

$$\begin{aligned} d(s) &= \left[\left\|r_{i,1} - r_{i-1,1}\right\|, \left\|r_{i,2} - r_{i-1,2}\right\|, \left\|r_{i,3} - r_{i-1,3}\right\|, \left\|r_{i,4} - r_{i-1,4}\right\|\right], \\ &= \left[2r \, \sin\left(\alpha + \alpha_{i-1,1}\right), 2r \, \sin\left(\alpha + \alpha_{i-1,2}\right),\right. \\ &\quad \left. 2r \, \sin\left(\alpha + \alpha_{i-1,3}\right), 2r \, \sin\left(\alpha + \alpha_{i-1,4}\right)\right]. \end{aligned} \tag{26}$$

The estimated value of the acoustic sensor $\widehat{d}$ has a random error ζ, and the variance of the error is σ^2.

$$\widehat{d} = d(s) + \varsigma. \tag{27}$$

Weighted error function (s) is as follows:

$$\varepsilon(s) = (\widehat{d} - d(s))^T \Phi_n^{-1} (\widehat{d} - d(s)), \tag{28}$$

where $\Phi_n = \sigma^2 I$ and I is the unit matrix.

The objective optimization function is obtained by minimizing the weighted error function $\varepsilon(s)$.

$$\widehat{s} = \arg\min_s \varepsilon(s). \tag{29}$$

The Levenberg–Marquardt algorithm is used to solve equation (29).

Linearize the error function $\varepsilon(s)$ by linearly expanding $d(s)$ through a first-order Taylor series:

$$d(s + \Delta s) = d(s) + J(s)\Delta s, \tag{30}$$

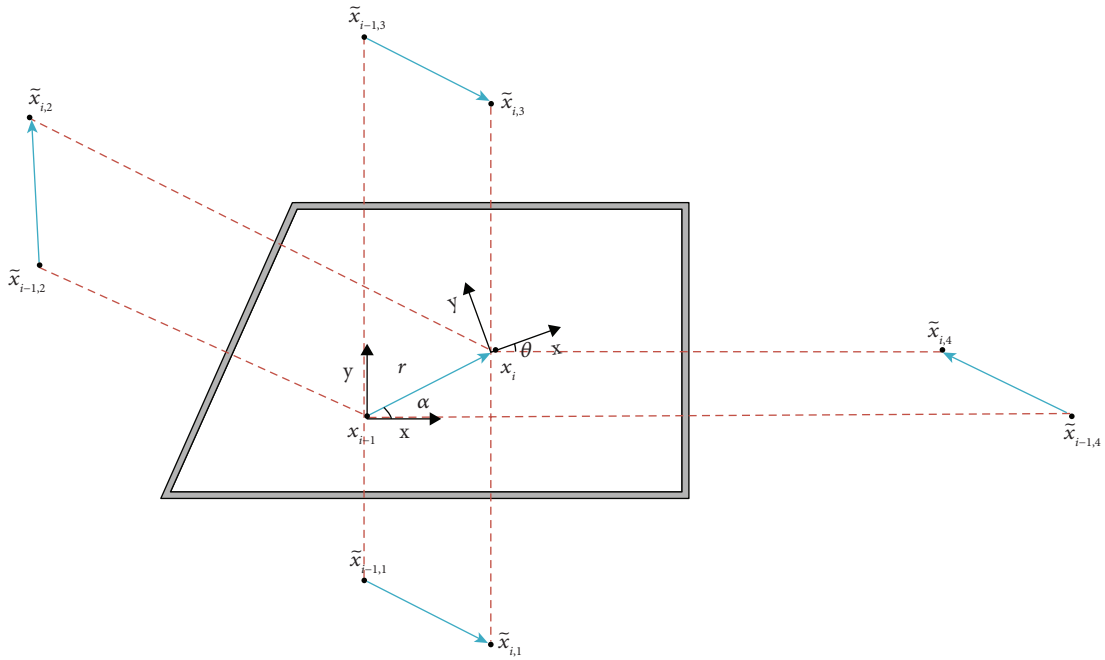


Figure 6: Isosceles trapezoidal constraint diagram.

where $J(s)$ is the Jacobi matrix, which is expressed as follows:

$$J(s) = \begin{bmatrix} 2r\sin\left(\alpha + \alpha_{i-1,1}\right) & 2r\cos\left(\alpha + \alpha_{i-1,1}\right) \\ 2r\sin\left(\alpha + \alpha_{i-1,2}\right) & 2r\cos\left(\alpha + \alpha_{i-1,2}\right) \\ 2r\sin\left(\alpha + \alpha_{i-1,3}\right) & 2r\cos\left(\alpha + \alpha_{i-1,3}\right) \\ 2r\sin\left(\alpha + \alpha_{i-1,4}\right) & 2r\cos\left(\alpha + \alpha_{i-1,4}\right) \end{bmatrix}. \quad (31)$$

Equation (30) is brought into equation (28) to solve for the extreme value of the weighted error function $\varepsilon(s)$. According to the Levenberg–Marquardt method:

$$(H + \lambda I)\Delta s = g, \quad (32)$$

where $H = J(s)^T J(s)$ and $g = J(s)(\hat{d} - d(s))$.

The above equation allows to find the step size Δs_k for each iteration. The value $s_0 = (r_0, \alpha_0)$ estimated by the IMU is used as the initial value for the iterative calculation.

$$\hat{s} = s_0 + \sum_{k=1}^{n} \Delta s_k, \quad (33)$$

where n is the number of iterations.

According to the above method, it is possible to use the acoustic signal to accurately estimate the pose change between different positions of the robot.

3.2.3. Pose Correction Based on Graph Optimization. According to the above method, we can construct the graph. Every time the robot moves a certain distance or rotates a certain arc, a vertex is added to the graph, and the constraint relationship between the vertices is established according to the pose estimation algorithm. The structure of the graph is shown in Figure 7.

Let $x = (x_1, x_2, \ldots, x_T)$ be a vector of parameters, where x_i describes the pose of node i. The robot moves from the pose node xi to the pose node x_j, $\hat{z}_{ij}$ is the pose transformation estimated by the IMU and z_{ij} is the pose transformation observed by the acoustic sensor. Let $e(x_i, x_j)$ be the error function from x_i to x_j, which is the difference between the robot's predicted observation $\hat{z}_{ij}$ and the actual observation z_{ij}.

$$e_{ij}\left(x_i, x_j\right) = z_{ij}\left(x_i, x_j\right) - \hat{z}_{ij}\left(x_i, x_j\right). \quad (34)$$

The dashed box in Figure 7 shows the constraint relationship between node x_2 to node x_p.

Let C be the set of constraint pairs of nodes in the graph and the set of nodes in the graph of trajectory points x of the robot. The goal of the maximum likelihood method is to find the configuration of nodes x^* that minimizes the negative log likelihood $F(x)$ of all observations.

$$F(x) = \sum_{(i,j)\in c} e_{ij}^T \Omega_{ij} e_{ij}, \quad (35)$$

where Ω_{ij} is the measurement information matrix of the error function e_{ij}.

The objective optimization function is

$$x^* = \arg\min_x F(x). \quad (36)$$

Using the first-order Taylor expansion error function $e_{ij}(x_i, x_j)$.

$$e\left(x_i + \triangle x_i, x_j + \triangle x_j\right) = e_{ij}\left(x_i, x_j\right) + J_{ij}\Delta x. \quad (37)$$

The error function $e_{ij}(x_i, x_j)$ is only related to x_i and x_j. Its Jacobi matrix J_{ij} is

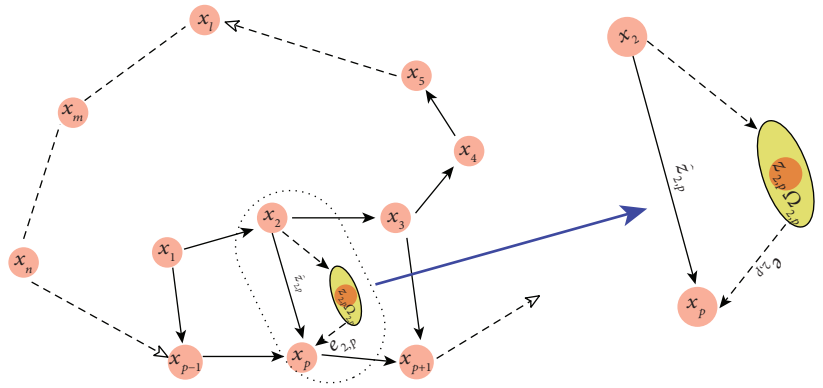


Figure 7: Schematic diagram of the connection between nodes.

$$J_{ij} = \frac{\partial e_{ij}\left(x_i, x_j\right)}{\partial x} = \left(0 \ldots 0, \frac{\partial e_{ij}\left(x_i, x_j\right)}{\partial x_i}, 0 \ldots 0, \frac{\partial e_{ij}\left(x_i, x_j\right)}{\partial x_j}, 0 \ldots 0\right). \tag{38}$$

The nonlinear optimization algorithm is used to iteratively solve for the minimum value of $F(x)$. The step size of each iteration can be solved using equation (32).

$$\Delta x = -(H + \lambda I)^{-1} b, \tag{39}$$

where $H = \sum_{(i,j)\in c} {J_{ij}}^T \Omega_{ij} J_{ij}$ and $b = \sum_{(i,j)\in c} {J_{ij}}^T \Omega_{ij} e_{ij}$.

The optimal robot trajectory point x^* is obtained by iterative calculation.

$$x^* = x_0 + \sum_{k=1}^{n} \Delta x_k, \tag{40}$$

where x_0 is the initial trajectory point of the robot estimated by the IMU and n is the number of iterations.

The robot travels around the room once, constructs the vertices and edges of the graph according to the method in this paper, and solves equation (36) using a nonlinear optimization algorithm. The optimal robot movement trajectory is obtained by optimizing the length of the edges of the graph to minimize $F(x)$.

4. System Implementation and Experimental Verification

This section introduces our self-designed robot prototype and then experimentally verifies the performance of the acoustic SLAM method in this paper.

4.1. System Implementation. Our robot is based on the Turtlebot3 Waffle Pi robot, a small, low-cost, fully programmable, ROS-based mobile robot, as shown in Figure 8(b). Turtlebot3 consists mainly of Raspberry Pi3 and OpenCR control board (with IMU sensor inside). In this system, OpenCR is responsible for collecting the built-in IMU sensor data and sound data as well as driving the robot to move, Raspberry Pi 3 is responsible for processing and calculating the data, and Raspberry Pi 3 is connected to OpenCR via USB 2.0. The system architecture connection diagram of the robot is shown in Figure 8(a). Since the sound source is close to the microphone array, which may result in larger direct waves and affect the reception of other reflected waves, a sound insulation panel was designed between the microphone array and the sound source to isolate the direct sound. In addition, the sound insulation panel can also isolate the reflected waves from the upper and lower walls. The physical diagram of the robot is shown in Figure 8(c).

The robot uses an active acoustic scheme for self-localization and room contour estimation, and to prevent the sound signals emitted during the robot's work from affecting people's life and work, this paper uses sound pulse signals in the pseudoultrasonic band (16k–24k), which has a wavelength between 1.4 cm and 2.1 cm and can be guaranteed to be received by a small microphone array, and the sound signals in this band are insensitive to the human ear but can be picked up by the robot's acoustic sensors. Sound source emits a sound signal of 16 kHz–20 kHz chirp pulse signal, which can avoid the leakage of sensitive information such as indoor human voice and ensure the security of privacy, but most speakers do not support the sound signal of the band, the need for speaker selection. Sound generation equipment was used, that is, Huawei Sound is used as the sound source. Huawei Sound is a 360-degree omnidirectional speaker with a frequency response range of 55 Hz–40 kHz and supports 3.5 mm wired audio input.

The process of acquiring the sound pulse signal from the robot itself is the process of converting the sound signal from a mechanical wave to a digital signal. The sound signal is first converted into a voltage signal through a microphone, and since this voltage signal is usually small, it needs to be amplified by an amplifier; then, the amplified signal is passed through a 15k–24k bandpass filter to filter out the noise signal outside the pseudoultrasonic band. Finally, the filtered signal is converted to a digital signal by an ADC.

Based on the acquisition process of the sound signal, the microphone array signal acquisition board for the robot was designed. The microphone in the acquisition board is a 130F22 omnidirectional microphone from PCB, which has a frequency response range of 10 Hz–20 kHz, and the

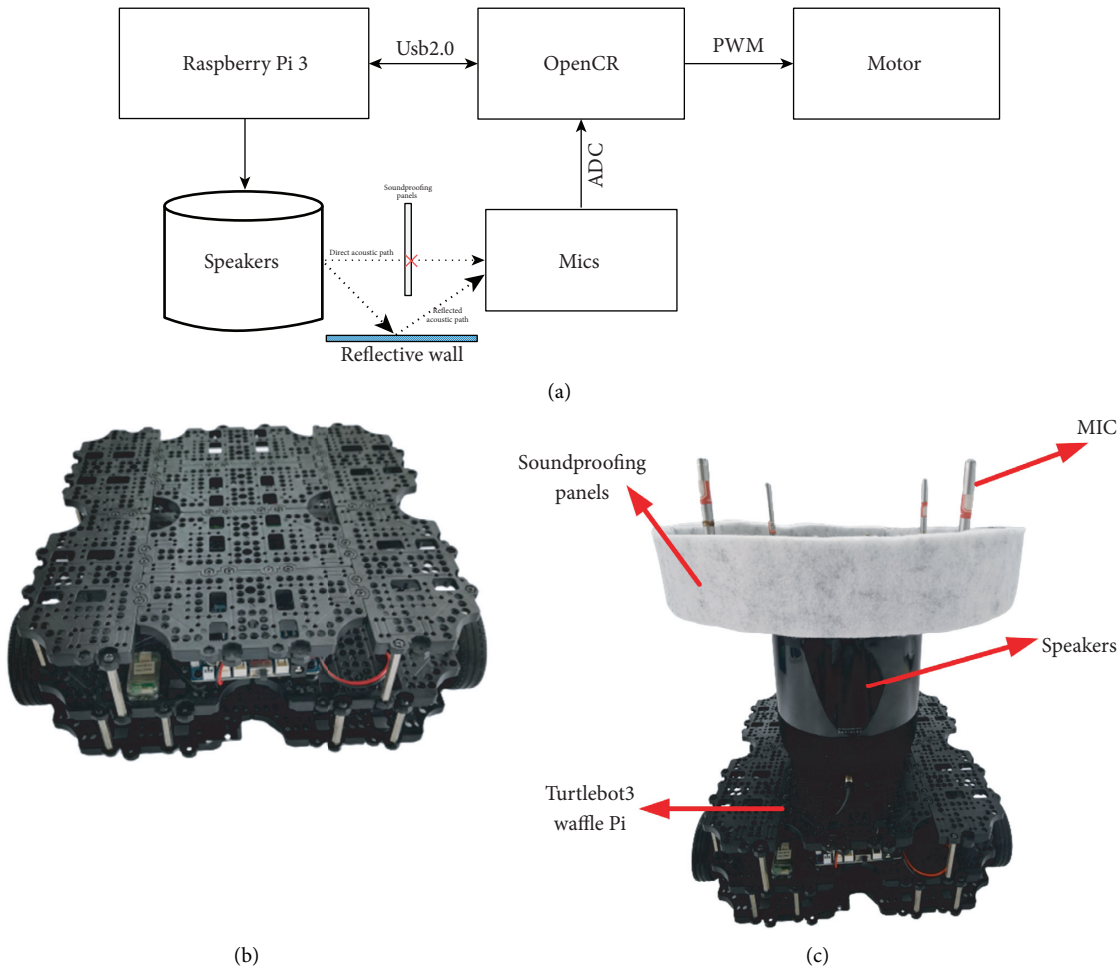


Figure 8: (a) Connection diagram of each module of the robot: (b) physical diagram of Turtlebot3 Waffle Pi; (c) physical diagram of the robot.

microphone is an SMB interface that can be plugged into the acquisition board standing up. The acquisition board is a four-channel microphone array, and the microphones are distributed at equal intervals on a circle with a radius of 0.14 m. The physical diagram of the acquisition board is shown in Figure 9.

4.2. Experimental Verification. The experimental part is divided into echo labelling experiments, pose correction and room shape estimation experiments, and real room experiments.

4.2.1. Echo Labelling Algorithm Performance Simulation. For the echo labelling experiments, echo labelling simulations were conducted in three different shapes of rooms: square, rectangular, and pipeline. The shape of the room is schematically shown in Figure 10, and w1, w2, w3, and w4 were used to denote the four walls of the room, and their corresponding reflection coefficients are $[\alpha_1, \alpha_2, \alpha_3, \alpha_4]$. The radius of the microphone array of the robot is 0.2 m, and a location in the room is randomly selected, the RIR of each microphone under that location is simulated using the image source method, and the echoes are labelled using the method in Section 3. To verify the robustness of the algorithm to noise, the Gaussian white noise was added to the propagation distance of the signal as follows:

$$\widehat{d} = d + \varepsilon, \tag{41}$$

where d is the true propagation distance and ε is the additive noise, and its standard deviation σ varies from 0.01 to 0.05 in steps of 0.005. Similar to [15], the F1-score was used to evaluate the goodness of the echo markers.

Literature [15] solved ELP by means of elliptical iterations, and for comparison, experiments in a square room with reflection coefficients for each wall of [0.8, 0.8, 0.8, 0.8] were performed. Randomly 300 points were chosen to conduct echo labelling experiments with the method of this paper and compare with the method of literature [15]. The experimental results are shown in Figure 11(b), where method 1 is the above method and method 2 is the method of this paper.

Then, the F1-score of different numbers was simulated of microphone array robotic echo markers in each of the three rooms in Figure 10, where the reflection coefficient of the room is [0.5, 0.6, 0.7, 0.8]. The experimental results are

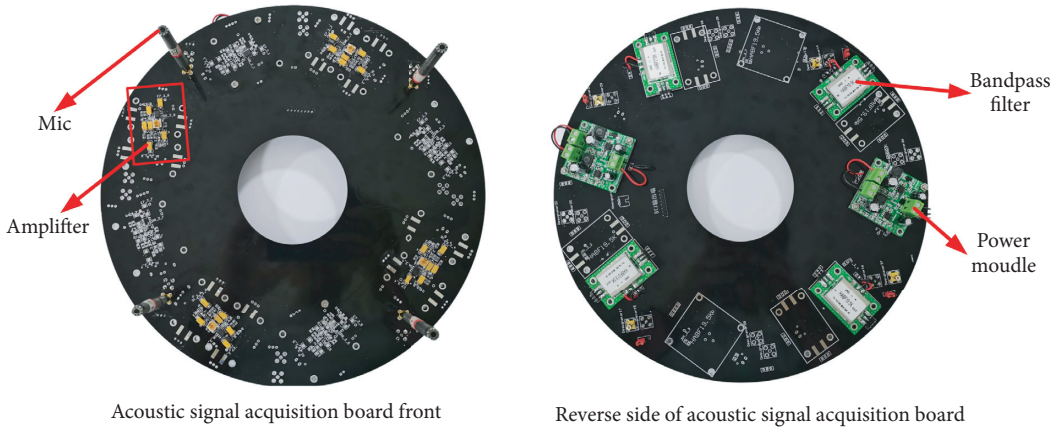


Figure 9: Physical diagram of the acoustic signal acquisition board.

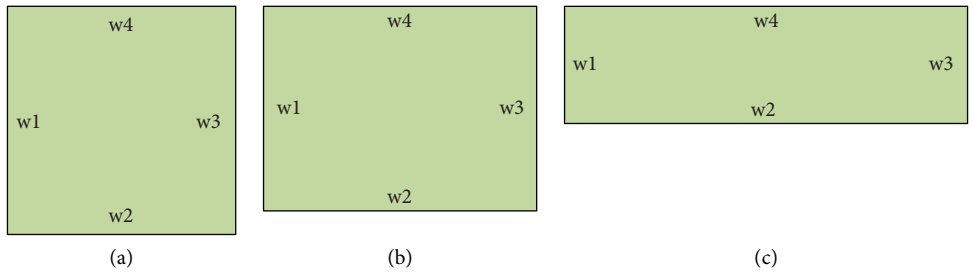


Figure 10: (a) Square room (8 ∗ 8). (b) Rectangle room (8 ∗ 6). (c) Pipeline room (10 ∗ 4).

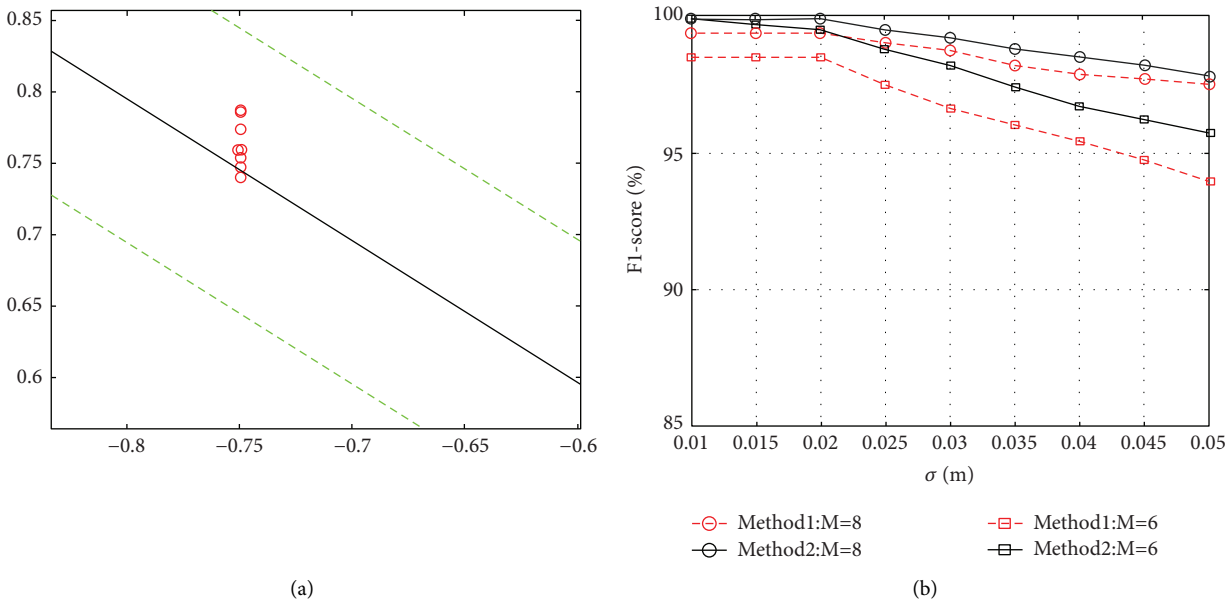


Figure 11: (a) Schematic diagram of the RIR cepstrum echo markers. The red circle is the RIR cepstrum, and the green dashed line is the upper and lower boundaries of the RIR cepstrum. (b) Comparison of the F1-scores of method 1 and method 2.

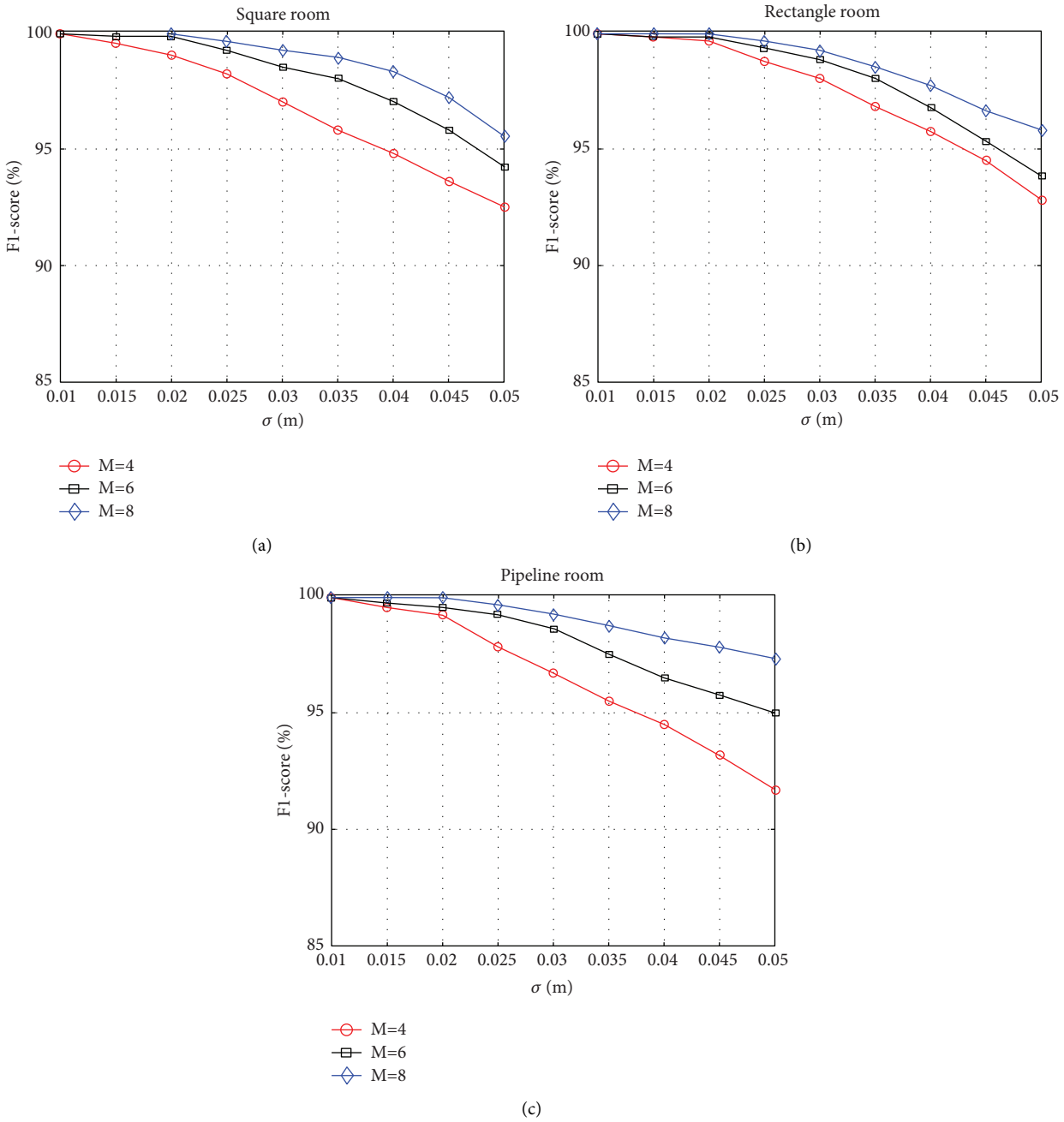


Figure 12: F1-score results for different σ values: (a) square room, (b) rectangle room, and (c) pipeline room.

shown in Figure 12. The experiments show that the F1-score can be maintained above 95% in all three rooms under low noise conditions ($\sigma \leq 0.03$ m).

4.2.2. Simulation of Pose Correction and Room Shape Estimation. Simulation experiments were conducted on robot self-localization and room position estimation in a room with wooden door and glass windows of size 7 m ∗ 7 m. Among them, the sound reflection coefficient of wooden door is 0.7, the sound reflection coefficient of glass window is 0.8, and the sound reflection coefficient of wall is 0.9. The robot's microphone array is a four-channel microphone array with a radius of 0.2 m. The robot travels around the room along the wall, and every 0.4 m, the robot actively emits sound and simulates the RIR of the current position ($\sigma = 0.05$ m).

The blue diamond line in Figure 13(b) shows the trajectory of the robot without pose correction, and the green line is the closed-loop result detected by the method in this paper. Figure 13(c) shows the trajectory after the pose correction based on the graph optimization, and the optimized path trajectory is basically consistent with the real trajectory. Figure 13(d) shows the location of the reflector estimated based on the optimized path, where the green "×" is the estimated location of the wall, the red "×" is the

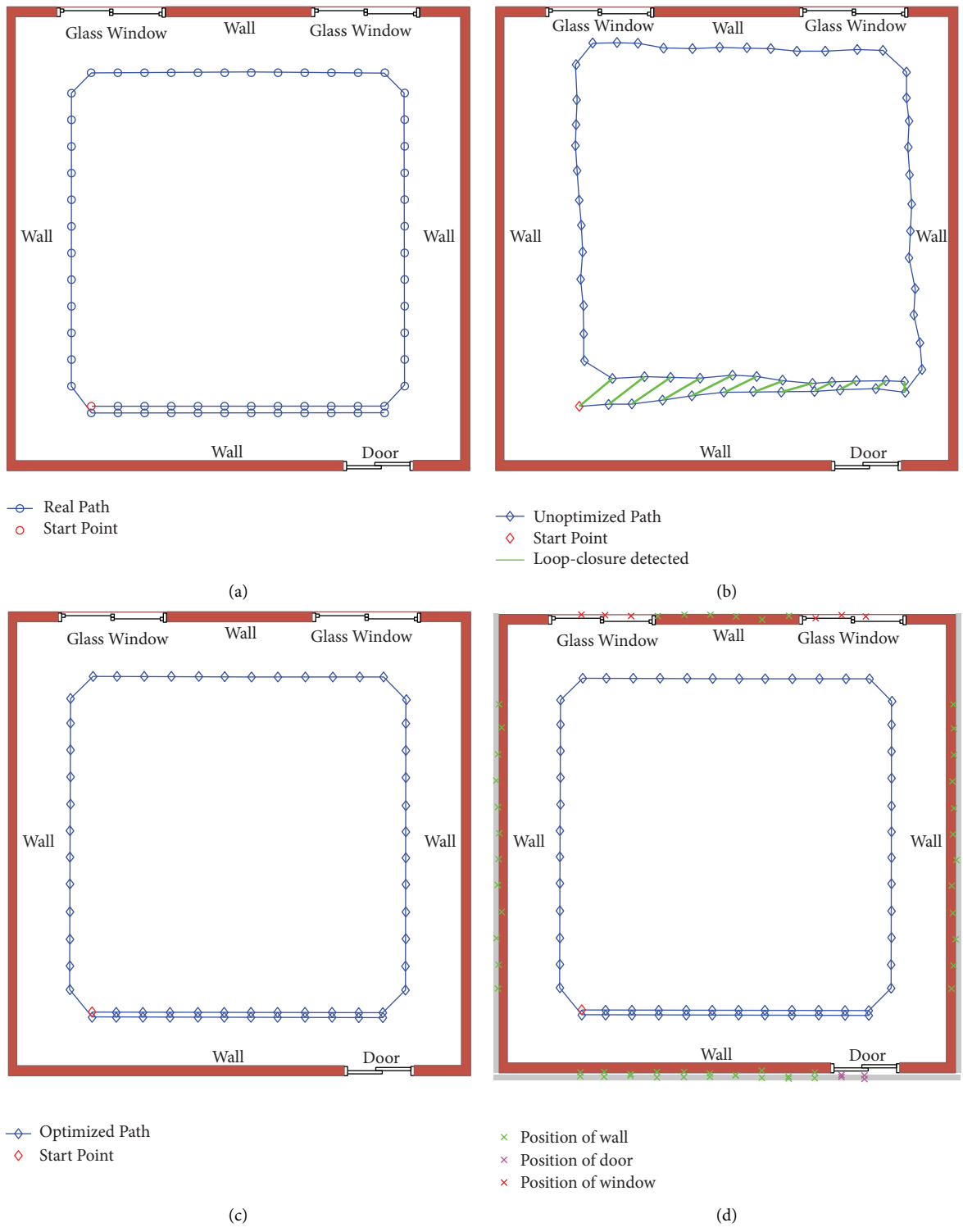


FIGURE 13: (a) Real room contour and real trajectory of the robot, (b) unoptimized robot trajectory, (c) optimized robot trajectory, and (d) estimated wall position based on the optimized trajectory.

location of the glass, and the pink "×" is the location of the door.

The position error $\text{Err} = \sqrt{X_{\text{err}}^2 + Y_{\text{err}}^2}$ was used to measure the robot's self-positioning error and mapping error, where X_{err} is the X-axis coordinate error and Y_{err} is the Y-axis coordinate error. Figure 14(a) shows the average self-localization error and mapping error statistics of the robot traveling some of the position points according to the route in Figure 13(a). The self-positioning error of the robot is less than 3.18 cm with 60% probability, and the average mapping error is less than 4.86 cm with 58% probability.

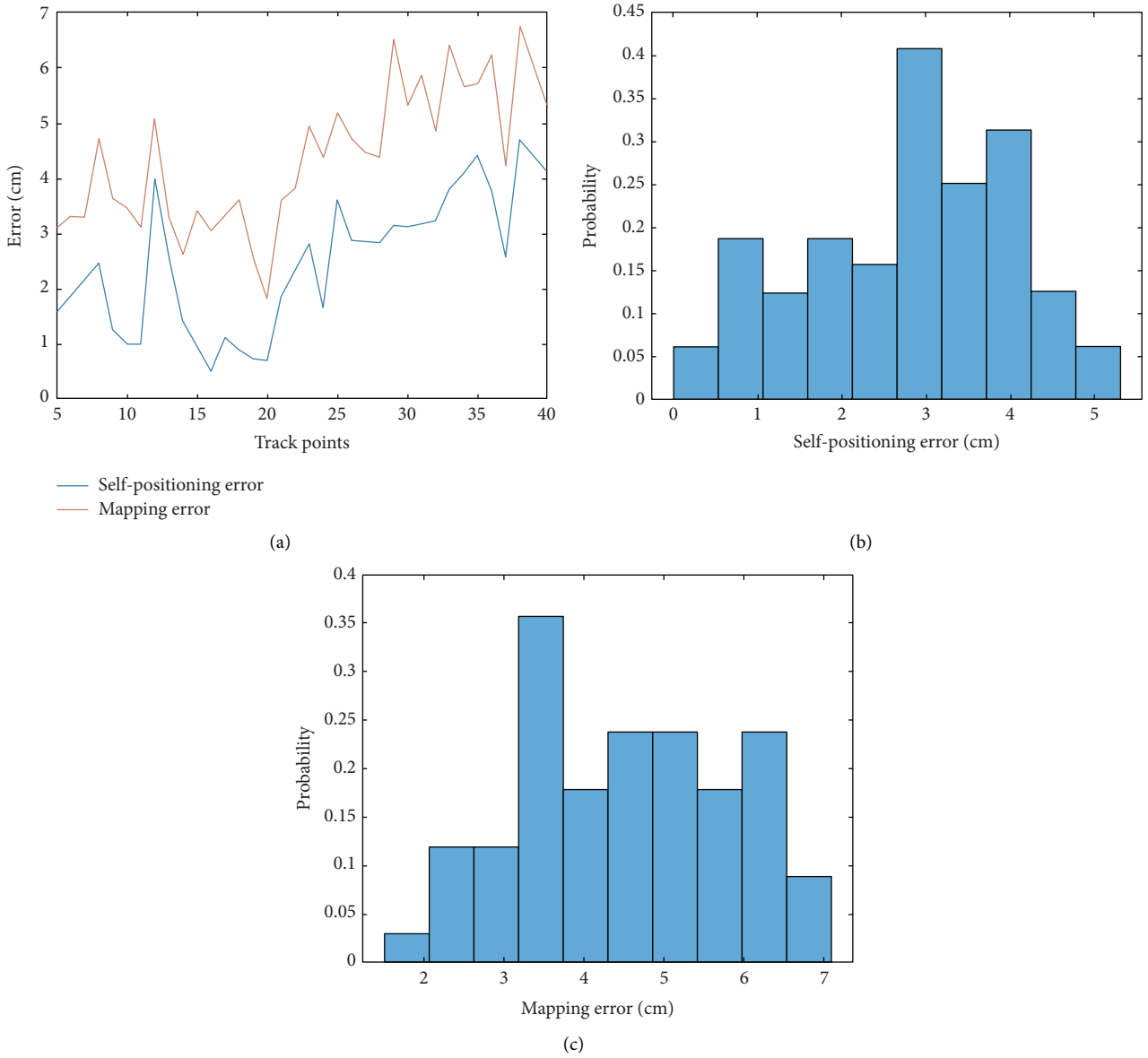


FIGURE 14: (a) Statistics of robot self-positioning error and reflected wall position estimation error. (b) Histogram of probability distribution of self-positioning error. (c) Histogram of probability distribution of mapping error.

The comparison experiments between the method in this paper and the method based on KEF filtering was done, in which the robot drove around the room randomly once in the above room and simulated 100 Monte Carlo experiments, respectively. The average self-localization error and mapping error of the experiments are shown in Table 1, where method 1 is the method without path optimization, method 2 is the method of path optimization by KEF filtering, and method 3 is the method of this paper.

4.2.3. Real Room Experiments. To verify the stability of our own designed robot and the practical performance of the method in this paper, the experiments were conducted in a real room of $4.3\,\text{m} \times 5.5\,\text{m} \times 3\,\text{m}$ (room dimensions were obtained using total station measurements). The total station was placed at the doorway and was used to measure the actual position of the robot as well as the actual position of the walls. The positions of the total station and the robot in the room are shown in Figure 15(a). Due to the height limitation of the robot, the robot can only measure the reflected wall under the red line in the right figure in Figure 15(a). The robot travels around the room close to the wall, and every time it moves, the robot actively vocalizes once (moving distance is less than 0.5 m) and records the RIR of the current position. Every time the robot moves during the experiment, the real position of the robot is measured with the total station and recorded. The RIRs obtained by the four microphones are shown in Figure 15(b) (the ambient temperature of the experiment is 30 degrees, and the corresponding sound speed is 349.75 m/s). Red marker points are first-order echoes from the wall, and red marker points of the same shape are first-order echoes from the same wall.

Table 1: Position error simulation results.

	Self-positioning error (cm)	Mapping error (cm)
Method 1	24.5	25.8
Method 2	3.01	4.53
Method 3	2.78	4.38

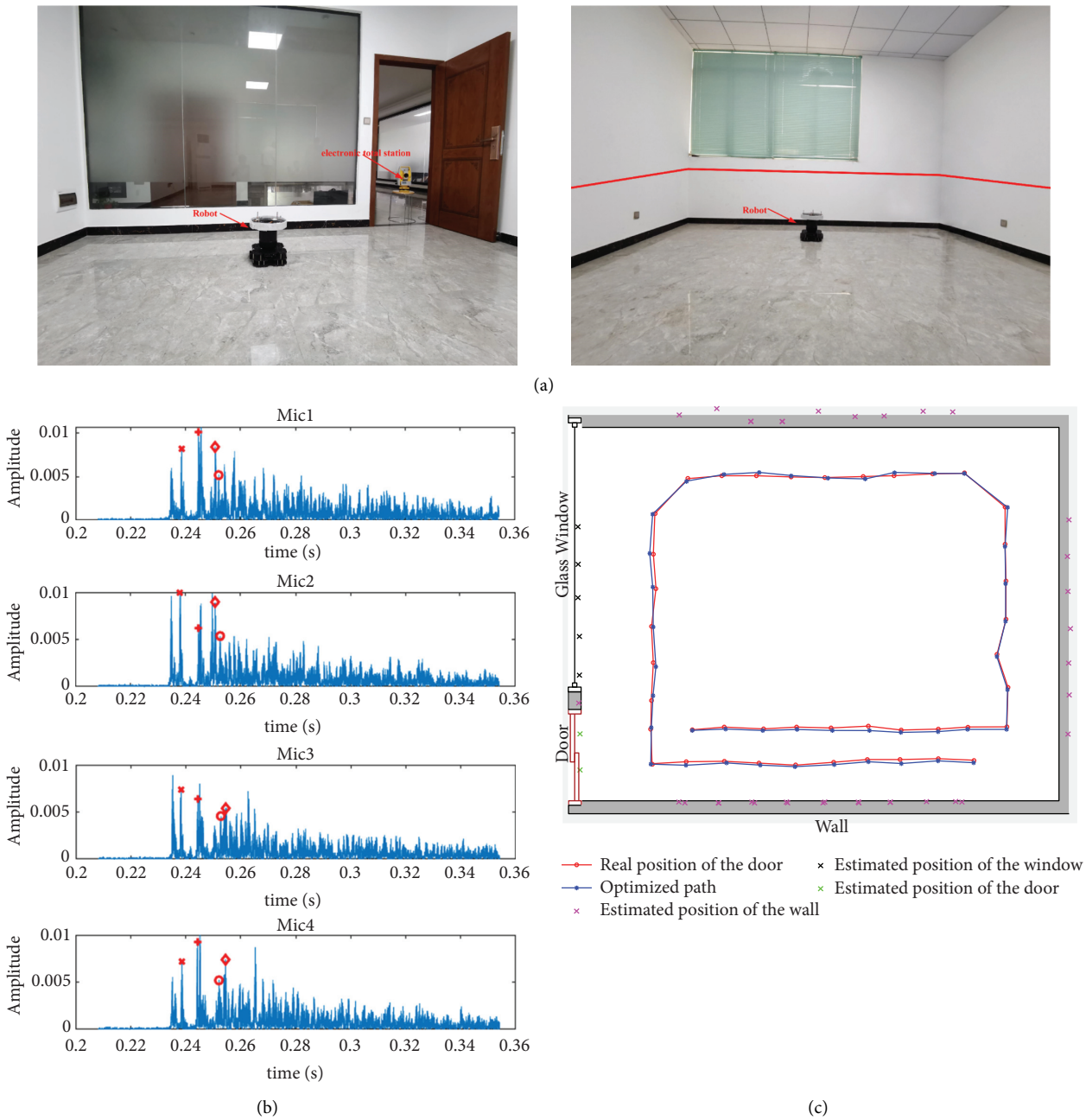


Figure 15: Experimental setup and results: (a) experimental room scene, (b) extracted first-order echo marker results in RIR, (c) experimental results graph.

Since the sound insulation panels were added between the speaker and the microphone receiver board and the sound propagation direction of the speaker is 360 degrees in the horizontal direction, there is no first-order reflection echo from the upper and lower walls. The final experimental results are shown in 2D, as shown in Figure 15(c). The

overall average self-positioning error of the robot is 2.84 cm, and the average mapping error is 4.86 cm.

5. Conclusion

This study introduces a graph optimization-based acoustic SLAM edge computing system and a method that provide new ideas for the solution of the acoustic SLAM problem. Based on the solution in this study, the robot can use acoustic signals to achieve self-localization and centimeter-level room map construction services containing door and window information. The current method in this paper has better performance in an empty room. In the future, acoustic SLAM research will be conducted in more complex indoor spaces.

Data Availability

The simulation and experimental data used to support the findings of this study have not been made available because this paper is funded by the Guangxi Science and Technology Plan Project (No. AD18281044). The grant is still in the research phase, and all research data are currently restricted to disclose within the project team.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research work was funded by the Guangxi Science and Technology Plan Project (Nos. AD18281044 and AD18281020), the Guangxi Keypoint Research and Invention Program (No. AB18221011), the Dean Project of Key Laboratory of Cognitive Radio and Information Processing, Ministry of Education (Nos. CRKL190104 and CRKL200107), and the Innovation Project of Guangxi Graduate Education (No. 2020YCXS024).

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[2] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An overview to visual o and visual SLAM: applications to mobile robotics," *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015.

[3] R. Frikha, R. Ejbali, and M. Zaied, "Camera pose estimation for augmented reality in a small indoor dynamic scene," *Journal of Electronic Imaging*, vol. 26, no. 5, Article ID 053029, 2017.

[4] C. Häne, L. Heng, G. H. Lee et al., "3D visual perception for self-driving cars using a multi-camera system: calibration, mapping, localization, and obstacle detection," *Image and Vision Computing*, vol. 68, pp. 14–27, 2017.

[5] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Stockholm, Sweden, May 2016.

[6] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, 2007.

[7] J. O'Reilly, S. Cirstea, M. Cirstea, and J. Zhang, "A novel development of acoustic SLAM," in *Proceedings of the 2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, pp. 525–531, IEEE, Istanbul, Turkey, August 2019.

[8] M. Crocco, A. Trucco, and A. Del Bue, "Room reflectors estimation from sound by greedy iterative approach," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6877–6881, IEEE, Calgary, Canada, April 2018.

[9] I. Dokmanic, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, "Acoustic echoes reveal room shape," *Proceedings of the National Academy of Sciences*, vol. 110, no. 30, pp. 12186–12191, 2013.

[10] I. Dokmanić, L. Daudet, and M. Vetterli, "From acoustic room reconstruction to slam," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6345–6349, IEEE, Shanghai, China, March 2016.

[11] I. Jager, R. Heusdens, and N. D. Gaubitch, "Room geometry estimation from acoustic echoes using graph-based echo labeling," in *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, Shanghai, China, March 2016.

[12] M. Coutino, M. B. Møller, J. K. Nielsen, and R. Heusdens, "Greedy alternative for room geometry estimation from acoustic echoes: a subspace-based method," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 366–370, IEEE, New Orleans, LA, USA, March 2017.

[13] F. Antonacci, J. Filos, M. R. P. Thomas et al., "Inference of room geometry from acoustic impulse responses," *IEEE Transactions on Audio Speech and Language Processing*, vol. 20, no. 10, pp. 2683–2695, 2012.

[14] X. Alameda-Pineda and R. Horaud, "A geometric approach to sound source localization from time-delay estimates," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 6, pp. 1082–1095, 2014.

[15] S. Park and J.-W. Choi, "Iterative echo labeling algorithm with convex hull expansion for room geometry estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1463–1478, 2021.

[16] C. Evers, A. H. Moore, and P. A. Naylor, "Acoustic simultaneous localization and mapping (a-SLAM) of a moving microphone array and its surrounding speakers," in *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6–10, IEEE, Shanghai, China, March 2016.

[17] C. Evers and P. A. Naylor, "Acoustic SLAM," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1484–1498, 2018.

[18] M. Krekovic, I. Dokmanic, and M. Vetterli, "EchoSLAM: simultaneous localization and mapping with acoustic echoes," in *Proceedings of the IEEE International Conference on Acoustics*, March 2016.

[19] A. H. Moore, M. Brookes, and P. A. Naylor, "Room geometry estimation from a single channel acoustic impulse response," in *Proceedings of the Signal Processing Conference*, pp. 1–5, IEEE, Uttar Pradesh, India, April 2014.

[20] V. Maya, N. Yair, and S. Gannot, "The hybrid Cramér-Rao lower bound for simultaneous self-localization and room geometry estimation," *EURASIP Journal on Applied Signal Processing*, vol. 2021, no. 1, pp. 1–22, 2021.

[21] L. Nguyen, J. V. Miro, and X. Qiu, "Can a robot hear the shape and dimensions of a room," 2019, https://arxiv.org/abs/1907.01169.

[22] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small - room acoustics," *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[23] J. Borish, "Extension of the image model to arbitrary polyhedra," *Journal of the Acoustical Society of America*, vol. 75, no. 6, pp. 1827–1836, 1984.

[24] H. Guo, M. Li, X. Zhang, Q. Liu, and X. Gao, "Research on indoor wireless positioning precision optimization based on UWB," *Journal of Web Engineering (JWE)*, pp. 94–116, 2020.

[25] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based Slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2011.

[26] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: a general framework for graph optimization," in *Proceedings of the IEEE International Conference on Robotics & Automation*, IEEE, Shanghai, China, May 2011.

[27] C. Cadena, L. Carlone, H. Carrillo et al., "Past, present, and future of simultaneous localization and mapping: toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[28] K. Imoto and N. Ono, "Spatial-feature-based acoustic scene analysis using distributed microphone array," in *Proceedings of the European Signal Processing Conference (EUSIPCO) 2015*, September 2015.

[29] X. Song, M. Wang, H. Qiu, and L Luo, "Indoor pedestrian self-positioning based on image acoustic source impulse using a sensor-rich smartphone," *Sensors*, vol. 18, no. 12, 2018.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 7310771, 10 pages
https://doi.org/10.1155/2021/7310771

Research Article

Efficient and Secure Cross-Domain Sharing of Blockchain Electronic Medical Records Based on Edge Computing

Yage Cheng,[1,2] **Bei Gong**,[1,3] **ZhiJuan Jia**,[1,2] **YanYan Yang**,[1,2] **Yuchu He**,[1] **and Xiaofei Zhang**[1]

[1]*College of Information Science and Technology, Zhengzhou Normal University, Zhengzhou 450044, China*
[2]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[3]*College of Computer Sciences, Beijing University of Technology, Beijing 100124, China*

Correspondence should be addressed to Bei Gong; gongbei@bjut.edu.cn and ZhiJuan Jia; jzj523@163.com

Received 22 July 2021; Accepted 4 October 2021; Published 19 November 2021

Academic Editor: Xiaolong Xu

In this article, we analysed the problems of electronic medical records (EMRs) and found that the EMRs generated by different hospitals for the same patient are mutually independent and duplication and data sharing are difficult among hospitals. In order to solve this problem, this paper proposes an efficient and secure cross-domain sharing scheme of EMRs based on edge computing. The program allows the doctor to access the personal history EMRs through the patient's authorization so that the doctor can understand the patient's history of illness and, on this basis, generate a new medical record for the patient. Then, the doctor sends the EMRs to the edge server, and the server calculates the ciphertext and adds it to the patient's personal medical record to complete the case update. Analysis shows that this solution can effectively prevent data tampering and forgery through blockchain and avoid privacy leakage problems in plaintext sharing by using searchable encryption and by relying on edge servers to solve nearby computing tasks and divert the computing capacity of cloud servers to improve efficiency. The security proof shows that the scheme satisfies the complex problem of the BDH assumption. Performance analysis shows that the scheme is feasible and efficient.

1. Introduction

With the rapid development of the Internet of Things and cloud computing, intelligent systems such as intelligent transportation and smart cities are gradually becoming a hot research topic nowadays [1–3]. At the same time, with the sharp increase in medical demand and the gradual intensification of refined hospital management, the development of the informatization of the medical system is also imperative. Compared with paper medical records, EMRs are related to each other, easy to store, more environmentally friendly, and efficient [4, 5]. It effectively solves the problems of paper medical records [3]. So, it is very popular in hospitals.

However, with the rapid growth of EMRs, the problem of data islands in hospitals has become more prominent. When patients go to different hospitals, each hospital will generate a large amount of EMRs and store them in its own hospital independently, which cannot be shared among them. For doctors, it is impossible to understand the patient's illness history in other hospitals. On this basis, doctors are prone to misdiagnosis and even cause significant problems such as medical malpractice. Moreover, it is also not conducive for the patients to master and understand their health status [5]. In addition, EMRs store the patient's personal privacy information. If they are attacked, they will face security risks, such as privacy leaks [6].

In recent years, blockchain technology has developed rapidly. Due to the characteristics of immutability, data integrity, and distributed storage, blockchain technology has been widely used in all walks of life [7–9]. Since blockchain technology can ensure privacy and security in the

application of EMRs, many scholars have proposed solutions to the current problems of EMRs. Literature [10] proposed blockchain-based healthcare data gateway architecture, enabling the patients to control and share their EMRs easily and securely without violating privacy. It provides a new potential way to improve the intelligence of healthcare systems while keeping patient data private. Literature [11] proposed a blockchain-based EMRs data-sharing framework, using immutability, and built-in autonomy properties of the blockchain sufficiently address the access control challenges associated with sensitive data stored in the cloud. Literature [12] proposed an electronic medical care system based on blockchain, which builds an alliance chain among hospitals. Using the practical Byzantine fault-tolerant algorithm reduces the computational power and ensures the safety and stability of the system, and at the same time, it prevents data tampering and privacy leakage. Literature [13] proposes a framework for sharing medical system data services based on blockchain, which does not rely on a trusted third party and realizes safe storage and privacy protection. Literature [14] used attribute-based encryption and identity-based encryption to ensure data privacy and used blockchain techniques to ensure the integrity and traceability of the EMRs. The most significant advantage of blockchain-based EMRs is that users can securely share the EMRs among hospitals and other institutions. However, most of the existing research only discusses the security search and the data sharing without considering establishing system EMRs for individual patients.

In fact, due to the limited storage space, many medical institutions and enterprises store data on cloud servers. However, with the continuous increase of cloud computing data security issues, it is imperative to upload encrypted data to the cloud server. However, it will face the problem of how to implement ciphertext search when data are shared. In this case, searchable encryption technology came into being [15–18]. It supports ciphertext search while ensuring the security of the data sharing, saving a lot of network and computing costs, and making full use of the enormous computing resources of cloud servers to search for keywords on ciphertexts. Therefore, many electronic medical record sharing schemes use searchable encryption technology to realize ciphertext sharing. Literature [19] proposed a blockchain-based searchable encryption scheme for EMRs. The solution stores the index of EMRs in the blockchain using the blockchain to ensure the integrity, tamper-proof, and traceability of the EMRs index and using searchable encryption to realize ciphertext sharing. Literature [20] constructs a framework based on the blockchain. It uses private chains and alliance chains, combined with searchable technology, to realize the safe search of EMRs while ensuring personal privacy and information security. Literature [21] proposed a blockchain-based secure and privacy-protected EMRs sharing protocol. The scheme mainly uses searchable encryption and proxy reencryption to realize data security, privacy preservation, and access control. Literature [22] combines private chain and consortium chain and uses searchable encryption technology to realize data sharing with significant storage overhead. Literature [23] uses ciphertext strategy attribute-based encryption to encrypt EMRs, and only users with the required attributes can access the data, which can achieve fine-grained access control. The above schemes solved privacy security and ciphertext search through searchable encryption technology but did not consider deduplication.

In response to the above problems, we propose a personal EMRs system with deduplication based on edge server. The plan is to update the EMRs by the doctors in time through the patient's authorization with deduplication and then complete data update. Moreover, it is through blockchain and searchable encryption to ensure data and personal privacy security, and the edge server can offload the computing tasks of cloud services to improve computing efficiency.

2. Prerequisite

2.1. Bilinearity

Definition 1. Suppose G_1 is the additive group, G_2 is the multiplicative group, and the prime order is q. Define a bilinear operation e: $G_1 \times G_1 \longrightarrow G_2$ satisfying the following properties [24]:

(1) Bilinear: for any $a, b \in Z_q^*$, there is $e(g^a, g^b) = e(g, g)^{ab}$;

(2) Nondegeneracy: there are $g_1, g_2 \in G_1$ such that $e(g_1, g_2) \neq 1$;

(3) Computable: for any $g_1, g_2 \in G_1$, $e(g_1, g_2)$ can be calculated.

2.2. Bilinear Diffie–Hellman Hypothesis. Suppose G_1 is the additive group, G_2 is the multiplicative group, and the prime order is q. Define a bilinear operation e: $G_1 \times G_1 \longrightarrow G_2$; g is the generator of group G_1. Given a four-tuple (g, g^a, g^b, g^c), it is difficult to calculate $e(g, g)^{abc} \in G_2$.

Suppose algorithm A is used to solve the BDH problem, and its advantage is defined as ε, if $\Pr|A(g, g^a, g^b, g^c) = e(g, g)^{abc}| \geq \varepsilon$.

At present, there is no effective algorithm to solve the BDH problem. Therefore, it can be assumed that the BDH problem is complex [24].

2.3. Public Key Encryption with Keyword Search (PEKS) Based on Bilinear Mapping. H_1: $\{0,1\}^* \longrightarrow G_1$ and H_2: $G_2 \longrightarrow \{0,1\}^{\log p}$ are two hash functions.

(1) KeyGen(λ). Randomly select $\alpha \in Z_p^*$ and a generator g of group G_1, and output $(sk = \alpha, pk = g^\alpha)$;

(2) Index(pk, w). Randomly select $r \in Z_p^*$ for the keyword w, Calculate $t = e(H_1(w), pk^r) \in G_2$ and output index$(pk, w) = (g^r, H_2(t))$;

(3) Trapdoor(sk, w'). Using private key sk and keyword w to generate search trapdoor $T_{w'} = H_1(w')^\alpha \in G_1$;

(4) Search$(pk, \text{Index}, T_{w'})$. Set index$(pk, w) = (I_1, I_2)$; check if there is $H_2(e(T_{w'}, I_1)) = I_1$, and output the corresponding index if they are equal [24].

2.4. System Model. This paper aims to solve the difficulties in EMRs sharing among hospitals and the problems of isolated and repeated storage of cases. The program mainly uses blockchain and searchable encryption technology to ensure EMRs data and privacy security. The overall idea of the scheme is that when a patient sees a doctor, he first registers with the hospital, and the hospital makes an appointment for the patient. Then, the patient authorizes the doctor to generate EMRs and the doctor sends the EMRs and authorization guarantee to the edge server. The edge server encrypts the EMRs and retrieval information and uploads them to the cloud server and blockchain. When the patient goes to another hospital, the doctor needs to be authorized to visit the personal EMRs. Then, the doctor generates new EMRs after understanding the patient's history of illness and sends them to the edge server. The edge server marks the repeated case and then adds the newly added case to the patient's medical record to complete the case update.

The main entities involved in the system are patients, doctors, hospitals, cloud servers, edge server, and blockchain. The system architecture is shown in Figure 1.

Definition 2. The scheme is composed of the following algorithms:

Initialization: generate system parameters;

Key generation: generate the entity's keys;

Registration: the patient registers with the hospital; the hospital makes an appointment for the doctor.

Authorization: the patient authorizes the doctor to generate EMRs.

Generation and storage of electronic medical records: the doctor generates EMRs for the patient and sends them to the edge server. Then, the edge server calculates the ciphertext and index and uploads it;

Access: the doctor views the patient's previous EMRs. The doctor applies for an access request to the edge server and the edge server accesses the blockchain and cloud to obtain the information and then returns it to the doctor.

Update: the doctor deletes duplicate EMRs and sends them to edge server; the edge server updates and uploads them to cloud storage and blockchain.

2.5. Security Model. We define the formalized security model of the proposed scheme by the following games.

2.6. Keyword Privacy Security Game. If there is no adversary $\mathscr{A}$ who can infer the plaintext of the keywords from the ciphertext or trapdoor in probabilistic polynomial time, the privacy of the keywords can be guaranteed. Define the keywords privacy and security game as follows:

(1) Initialization: given the secure parameter λ, simulation challenger $\mathscr{B}$ executes the initialization algorithm to generate par.

(2) Phase 1: adversary $\mathscr{A}$ runs the trapdoor generation algorithm multiple times.

(3) Challenge: adversary $\mathscr{A}$ randomly selects two keywords from the keyword space and sends them to the simulation challenger. The simulation challenger executes the trapdoor generation algorithm and then randomly selects a trapdoor and sends it to $\mathscr{A}$.

(4) Guess: After adversary $\mathscr{A}$ inquires n times for the different keywords, it analyzes and guesses. If the $\mathscr{A}$ can guess the trapdoor, then adversary $\mathscr{A}$ wins the game.

2.7. Proof of Bilinear Diffie–Hellman Hypothesis for Difficult Problems. If there is an adversary $\mathscr{A}$ who can solve the solution with an advantage $\varepsilon(\lambda)$ in polynomial time, then the adversary $\mathscr{A}$ can solve the BDH difficult problem with an advantage $\varepsilon(\lambda)$ in polynomial time. Define the two-linear Diffie–Hellman hypothesis that the difficult problem specification is proved as follows:

(1) Initialization: given the group G_1, G_2 and the mapping e: $G_1 \times G_1 \longrightarrow G_2$. Simulate challenger $\mathscr{B}$ randomly generates $a, b, c \in Z_p^*$ and sets g, $x = g^a$, $y = g^b$, $z = g^c$.

(2) Phase 1: adversary $\mathscr{A}$ runs the encryption algorithm multiple times.

(3) Challenge: the simulate challenger $\mathscr{B}$ randomly selects the plaintext m, requires that m is not queried in stage 1, generates the ciphertext C_m, and transmits the ciphertext to the adversary $\mathscr{A}$.

(4) Guess: the adversary $\mathscr{A}$ analyzes and decrypts the ciphertext C_m. If the adversary $\mathscr{A}$ can decrypt the ciphertext C_m and get the correct plaintext m, then the adversary $\mathscr{A}$ wins the game.

(5) Proof: if adversary $\mathscr{A}$ can decrypt the ciphertext, adversary $\mathscr{A}$ can also solve the difficult problem of bilinear Diffie–Hellman assumption.

3. The Proposed

The program mainly includes the following essential roles: patients, hospitals, doctors, cloud storage servers, edge server, and alliance chain. The description of symbols in the text is shown in Table 1.

3.1. Initialization. The key generation center according to the security parameter λ generates the public parameter $\text{par} = \{p, g, G_1, G_2, e, H_1, H_2\}$, where G_1 and G_2 are the cyclic group of prime order p, the generator of group G_1 is g, e satisfies $G_1 \times G_1 \longrightarrow G_2$, and H_1: $\{0,1\}^* \longrightarrow G_1$ and H_2: $G_2 \longrightarrow \{0,1\}^{\log p}$ are two hash functions.

3.2. Key Generation. The patient $\mathscr{P}$ randomly selects $\alpha \in Z_p^*$ and calculates $h = g^\alpha$, so the keys of $\mathscr{P}$ are $(sk_{\mathscr{P}} = \alpha, pk_{\mathscr{P}} = g^\alpha)$. Similarly, the doctors $\mathscr{D}_1$ and $\mathscr{D}_2$ randomly select β and γ and calculate $d = g^\beta$ and $f = g^\gamma$, so

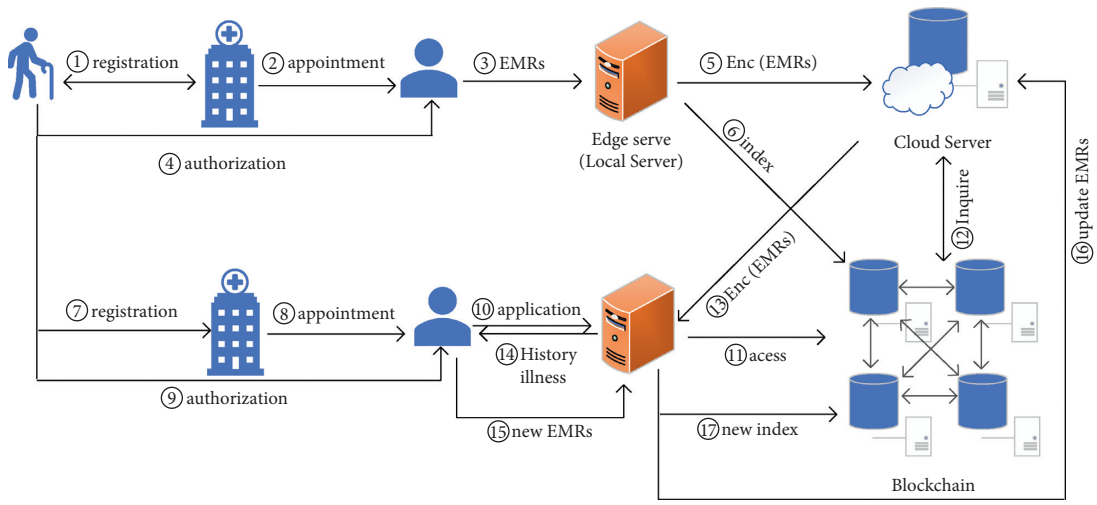


Figure 1: Cross-domain sharing scheme of EMRs.

Table 1: Symbol description.

Symbols	Roles
$\mathscr{P}$	Patient
$\mathscr{H}$	Hospital
$\mathscr{D}$	Doctor
$\mathrm{ID}_{\mathscr{P}}$	The patient's identification
$\mathrm{ID}_{\mathscr{D}}$	The doctor's identification
$\mathscr{CS}$	Cloud storage server
$\mathscr{BC}$	Alliance blockchain
$\mathscr{ES}$	Edge server (local server of hospital)
τ	Treatment key
Aux	Other auxiliary information
Dep	Department
App	Appointment information
Gua	Authorization guarantee
C	Ciphertext
T_{Access}	Access time
k	Access key
a	Repeat mark
t	Repeat time
A	Access
I	Index
T	Trapdoor

the keys of $\mathscr{D}_1$ and $\mathscr{D}_2$ are $(sk_{\mathscr{D}} = \beta, pk_{\mathscr{D}} = g^{\beta})$ and $(sk_{\mathscr{D}_2} = \gamma, pk_{\mathscr{D}_2} = g^{\gamma})$.

3.3. Registration. The patient $\mathscr{P}$ registers with the hospital $\mathscr{H}_1$, and the $\mathscr{H}_1$ stores the patient's identification $\mathrm{ID}_{\mathscr{P}}$, randomly selects the treatment key τ, and sends the encrypted $\mathrm{Enc}(pk_{\mathscr{P}}, \tau)$ to $\mathscr{P}$. The hospital $\mathscr{H}_1$ makes an appointment with the attending doctor $\mathscr{D}_1$ for the patient $\mathscr{P}$ and encrypts the appointment information $\mathrm{App} = \mathrm{ID}_{\mathscr{D}_1} \| \mathrm{Dep} \| \mathrm{Aux}$ with the τ and sends it to $\mathscr{P}$. The patient uses τ to decrypt the App and obtains the doctor's $\mathrm{ID}_{\mathscr{D}_1}$, department Dep, and other auxiliary information Aux. At the same time, the hospital $\mathscr{H}_1$ sends τ to the attending doctor $\mathscr{D}_1$.

3.4. Authorization. The patient $\mathscr{P}$ authorizes the doctor $\mathscr{D}_1$ to generate EMRs. $\mathscr{P}$ generates an authorization guarantee $\mathrm{Gua}_1 = \mathrm{ID}_{\rho} \| \mathrm{ID}_{D_1} \| T_{\mathrm{Access}_1} \| k_1 \| \tau$, $k_1 \in Z_p^*$, while signing it with the personal private key $\sigma_{\mathrm{Gua}_1} = \mathrm{sig}(sk_{\rho}, \mathrm{Gua}_1)$ and encrypting it with the doctor's public key $C_1 = \mathrm{Enc}(pk_{D_1}, Gua_1 \| \sigma_{Gua_1})$, and then sends C_1 to $\mathscr{D}_1$. The doctor $\mathscr{D}_1$ decrypts C_1 with the personal private key $sk_{\mathscr{D}_1}$ to obtain the Gua_1 and the signature σ_{Gua_1}, and then the doctor verifies the correctness of the authorization with the patient's public key $pk_{\mathscr{P}}$.

3.5. Generation and Storage of Electronic Medical Records. When the verification is passed, the doctor generates EMRs m_1 for $\mathscr{P}$ and sends them to edge server. The edge server calculates the ciphertext $C_{m_1} = H_1(k_1)m_1$ and then randomly selects $u \in Z_p^*$ and calculates $I_1 = g^u$, $I_2 = H_2(r)$, $I = \{I_1, I_2\}$, where $r = e(H_1(\mathrm{ID}_{\mathscr{P}}), pk_{\mathscr{D}}^u)$. Finally, it uploads $A_{\mathscr{CS}} = \{C_1, \mathrm{ID}_{\mathscr{D}_1}, I, C_{m_1}\}$ to the cloud server and uploads $A_{\mathscr{BC}} = \{H_1(\mathrm{ID}_{\mathscr{P}}), I_1, I_2, N\}$ to the blockchain, here N is the file number returned by the cloud server.

3.6. Access. When $\mathscr{P}$ registers and sees a doctor $\mathscr{D}_2$ in the hospital $\mathscr{H}_2$, $\mathscr{P}$ first authorizes D_2 to access his EMRs through the authorization guarantee $Gua_2 = \mathrm{ID}_{\rho} \| \mathrm{ID}_{D_2} \| \tau \| k_1 \| T_{\mathrm{access}}$ and encrypts it as $C_2 = \mathrm{Enc}(pk_{D_2}, Gua_2 \| \sigma_{Gua_2})$, where $\sigma_{Gua_2} = \mathrm{sig}(sk_{\rho}, Gua_2)$. The doctor D_2 sends $C_2 = \mathrm{Enc}(pk_{\mathscr{D}_2}, Gua_2 \| \sigma_{Gua_2})$ to edge server, the edge server decrypts C_2 with the personal private key $sk_{\mathscr{D}_2}$ to obtain the Gua_2 and the signature σ_{Gua_2} and then verifies the correctness of the authorization with the patient's public key $pk_{\mathscr{P}}$.

When the verification is passed, the edge server calculates $T = H_1(\mathrm{ID}'_{\mathscr{P}})^{\beta} \in G_1$ and sends $A = \{Gua_2, \mathrm{ID}_{\mathscr{D}_2}, T\}$ to the blockchain nodes. The blockchain nodes execute matching algorithms through $H_2(e(T, I_1)) = I_2$ and return

the corresponding file number N. The $\mathscr{CS}$ finds the corresponding ciphertext C_{m_1} through the file number N and returns it to edge server. The edge server sends it to the doctor $\mathscr{D}_2$. $\mathscr{D}_2$ views the patient's history EMRs C_{m_1} by the access key k_1 within the limited access time T_{Access_1}.

3.7. Update. When the doctor $\mathscr{D}_2$ obtains the patient's EMRs with the access key k_1, he first understands the patient's medical history through historical EMRs and generates a new EMRs $m\prime$ on this basis and sends them to edge server. Then, the edge server checks whether the new EMRs have duplicate data by comparing them with the historical EMRs. If there are duplicates, the edge server adds a mark a and a date t based on the historical EMRs and then encrypts the updated EMRs to ciphertext $\text{Enc}(m_2)$ with k_2 and adds the newly EMRs to the patient's personal EMRs system in order to complete the update of the EMRs.

When $\mathscr{P}$ registers and sees a doctor $\mathscr{D}_n$ in the hospital $\mathscr{H}_n$, repeat the above process.

4. Analysis

4.1. Correctness

Theorem 1. *In the search phase, the blockchain nodes need to verify the identity of the visitor and secondly verify whether the trapdoor submitted by the edge server has corresponding index and other information, that is, needs to verify whether the equation $H_2(e(T_{\mathscr{CS}}, I_1)) = I_2$ is established. If the equation holds, the corresponding index is returned for the doctor; otherwise, the visit is denied.*

Proof. According to the above, we know

$$\begin{aligned} e(T, I_1) &= e\left(H_1(\text{ID}_{\mathscr{P}}')^{\beta}, g^{u}\right), \\ &= e\left(H_1(\text{ID}_{\mathscr{P}}'), g^{u\beta}\right), \\ &= e\left(H_1(\text{ID}_{\mathscr{P}}'), pk_{\mathscr{D}}^{u}\right). \end{aligned} \tag{1}$$

If

$$\text{ID}_{\mathscr{P}}' = \text{ID}_{\mathscr{P}}, \tag{2}$$

then

$$e\left(H_1(\text{ID}_{\mathscr{P}}'), pk_{\mathscr{D}}^{u}\right) = e\left(H_1(\text{ID}_{\mathscr{P}}), pk_{\mathscr{D}}^{u}\right) = r. \tag{3}$$

So,

$$H_2\left(e(T, I_1)\right) = H_2(r) = I_2. \tag{4}$$

Through the proof, we can find that the verification equation is established, the ciphertext retrieval verification is successful, and the result is correct. So, it can retrieve the index information corresponding to the patient's history EMRs, and the correctness of the scheme is verified. □

4.2. Security. The scheme satisfies the difficult problem of the BDH assumption; the proof is as follows.

Theorem 2. *Assuming that the BDH problem is difficult, the scheme is indistinguishable under adaptive chosen ciphertext attacks (IND-CCA2).*

Suppose H_1: $(0,1)^* \longrightarrow G_1$ and H_2: $\{0,1\}^* \longrightarrow Z_p^*$ are two random oracles; $\mathscr{A}$ is the adversary of the superior $\varepsilon(k)$ attack scheme. At any time, A can ask H_1 or H_2 and ask at most q_{H_1} and q_{H_2} times, respectively. Constructing the simulator $\mathscr{B}$ can solve the BDH problem with at least the advantage of $2\varepsilon(k)/eq_{H_2}$ and the running time of $O(\text{time}(A))$.

Proof. Suppose the simulator $\mathscr{B}$ has known g, g^x, g^y, g^z $(x, y, z \in Z_p^*)$ and simulate the challenger, with $\mathscr{A}$ as the adversary, and the goal is to calculate $D = e(g,g)^{xyz} \in G_2$.

For simplicity, suppose (1) $\mathscr{A}$ will not initiate the same query to $H_1(\text{ID}_{\mathscr{P}})$ twice, and (2) if $\mathscr{A}$ requests a trapdoor for keyword $\text{ID}_{\mathscr{P}}$, it has already asked $H_1(\text{ID}_{\mathscr{P}})$ before.

(1) System establishment: the simulator $\mathscr{B}$ builds the system, generates the safety parameter λ, runs the algorithm setup (1^{λ}), obtains the safety parameter $\text{par} = \{p, g, G_1, G_2, e, H_1, H_2\}$, and generates the keys $K_{\mathscr{C}} = (sk_{\mathscr{C}}, pk_{\mathscr{C}})$ and keeps the private key $sk_{\mathscr{C}}$. The simulator chooses $(x, y, z \in Z_p^*)$, setup g, $u_1 = g^x$, $u_2 = g^y$, $u_3 = g^z \in G_1$. The simulator challenger $\mathscr{B}$ returns the parameters Par and the public key $pk_{\mathscr{C}}$ to adversary $\mathscr{A}$, and $\mathscr{A}$ asks the simulator $\mathscr{B}$ with random oracles.

(2) H_1 and H_2 query: $\mathscr{B}$ randomly chooses $l \in \{1, \cdots, q_{H_1}\}$. l is the guess value of $\mathscr{B}$, and the l-th query to H_1 corresponds to the final attack result of $\mathscr{A}$. At any time, $\mathscr{A}$ can ask H_1 or H_2 and ask at most q_{H_1} and q_{H_2} times, respectively.

(1) Inquire H_1: $\mathscr{B}$ creates an H_1^{list}, initially empty, and the element is $\langle w_i, h_i, a_i \rangle$. When $\mathscr{A}$ initiates the i-th query (set the query value as w_i), $\mathscr{B}$ responds as follows:
If w_i is already in the list H_1^{list}, $\mathscr{B}$ takes out the 3-tuple $\langle w_i, h_i, a_i \rangle$ and responds with $H_1(w_i) = h_i \in G_1$. Otherwise, $\mathscr{B}$ chooses a random $a_i \in Z_p$ and calculates as follows: if $i = l$, $\mathscr{B}$ calculates $h_i = y \cdot g^{a_i} \in G_1$; otherwise, $\mathscr{B}$ calculates $h_i = g^{a_i} \in G_1$. Then, $\mathscr{B}$ adds $\langle w_i, h_i, a_i \rangle$ to H_1^{list} and responds to $\mathscr{A}$ with h_i.

(2) Inquire H_2: similarly, $\mathscr{B}$ creates a list H_2^{list} (initially empty) with element type $\langle r_i, v_i \rangle$, $\mathscr{A}$ can query H_2^{list} at any time, and $\mathscr{B}$ responds as follows:
If s_i is already in H_2^{list}, answer with $H_2(r_i) = v_i$; otherwise, choose $v_i \in \{0,1\}^n$ randomly, answer with $H_2(r_i) = v_i$, and add $\langle r_i, v_i \rangle$ to H_2^{list}.

(3) Trapdoor query (at most q_{H_1} times): when $\mathscr{A}$ requests the trapdoor $T_{\mathscr{CS}}$ corresponding to the keyword w_i, let i satisfy $w = w_i$, and w_i represents the query value of the i-th query to H_1. $\mathscr{B}$ answers the query as follows:

If $i \neq l$, then there is a 3-tuple $\langle w_i, h_i, a_i \rangle$ in H_1^{list}, calculate and return $T_i = u_1^{a_i}$. If $i = l$, then interrupt.

(4) Challenge: $\mathscr{A}$ initiates a challenge. Suppose the keywords of $\mathscr{A}$'s challenge are w_0 and w_1, and $\mathscr{B}$ randomly selects $J \in \{0,1\}^{\log p}$ and responds with $C = [u_3, J]$.

Note that this response implicitly defines $H_2(e(H_1(w_b), {u_1}^z)) = J$. In other words, $J = H_2(e(H_1(w_b), {u_1}^z)) = H_2(e(yg^{a_b}, g^{ac})) = H_2(e(g^{a_b}, g^{az})^{az(y+a_b)})$. According to this definition, C is a valid trapdoor for the keyword w_b.

(5) Trapdoor query: $\mathscr{A}$ can continue to do trapdoor queries for the keyword w_i; the only restriction is that $w_i \neq w_0, w_1$, and $\mathscr{B}$ responds as before.

(6) Guess: $\mathscr{A}$ outputs the guess $b' \in (0,1)$, and $\mathscr{B}$ randomly selects $\langle r_i, v_i \rangle$ from H_2^{list} and outputs $r/e(u_1, u_3)^{a_b}$ as his guess of $e(g,g)^{xyz}$, where a_b is the value used in the challenge phase. This is because H_2^{list} contains a pair of $\langle r_i, v_i \rangle$, where $r = e(H_1(w_b), u_1^z) = e(g,g)^{az(y+a_b)}$. If $\mathscr{B}$ chooses this pair from H_2^{list}, then $r/e(u_1, u_3)^{a_b} = e(g,g)^{az(y+a_b)}$. The advantage of $\mathscr{B}$ choosing the correct result is $2\varepsilon(\lambda)/eq_{H_2}$, so the probability that $\mathscr{B}$ breaks the security of the proposed scheme is $\Pr|A(r/e(u_1, u_3)^{a_b}) = e(g,g)^{az}(y + a_b)| \geq 2\varepsilon(\lambda)/eq_{H_2}$. □

4.3. Performance. By comparing Table 2, we can find that all the above schemes are based on blockchain and realized access control and privacy protection functions. But none of the literatures [11, 20, 22, 23] can implement data deduplication. In addition, reference [11] did not use searchable encryption technology to realize ciphertext search, and reference [20] did not realize data sharing. Therefore, the function of this scheme is better.

Nowadays, there are many researches on EMRs, but it still faces many problems to be solved urgently. For example, we are familiar with privacy protection, access control, and data-sharing issues. With the development of science and technology, more problems have been exposed between the increasing demand of people and the actual status of EMRs. For example, there are no systematic EMRs for patients, and the storage of patients' EMRs is relatively scattered and unsystematic, which makes patients unable to understand personal health systematically. In addition, given the huge data storage and limited storage space of EMRs, deduplication is particularly important. Deduplication can effectively reduce storage consumption and improve storage efficiency. Therefore, it is also one of the urgent problems to be solved in EMRs. In response to the above problems, this article provides some solutions, as shown in the following.

According to Table 3, the plan allows the doctor to update the patient's previous EMRs, so the EMRs system can store the latest medical record in time which ensures the timeliness of the data and realizes integrity and systematic of the patient's EMRs data. Secondly, the deletion of duplicate data effectively improves storage efficiency and reduces storage overhead.

4.4. Simulation. The operating system used in the simulation experiment in this article is Windows 10, Intel CPU i7-9750H, and MyEclipse 2015 CI. From the initialization, key generation, encryption, decryption, indexing, and trapdoor generation stages, the execution efficiency of the scheme is investigated. The initialization phase is the configuration of system parameters. The key generation stage is mainly used to generate participants' personal keys. The encryption and decryption use symmetric encryption algorithms. Indexes and trapdoors are used for file query and retrieval. The program selected documents [22, 23] for comparison, and the selected documents were all EMRs sharing schemes based on the blockchain. The comparison results of each stage are shown in Figure 2.

It can be seen from Figure 2 that the execution efficiency of this article is relatively higher than that of documents [22, 23], and documents [22] need to be improved in terms of efficiency. In the index generation stage, the cost of this article is slightly higher than literature [23], while other stages are lower than the comparative literature. This is because literature [23] does not require bilinear operation in the index generation stage, while the solution of this paper needs to perform the bilinear operation, which makes the efficiency relatively lower than literature [23]. In the encryption and decryption stages, literatures [22, 23] require complex operations with the high cost of bilinear pairing and modular idempotence. While this scheme only needs one hash and one inverse operation, computational efficiency is relatively high. In the trapdoor generation stage, the solution in this paper only needs to perform power operation and hash operation, which is more efficient than the comparative literature.

In addition, to further verify the program's performance, the program uses keywords as variables to compare the execution efficiency of the index, trapdoor generation, and search phrases. Figure 3 is the execution time of the index generation phase, Figure 4 is the execution time of the trapdoor generation phase, and Figure 5 is the execution time of the retrieval phase.

It can be seen from Figures 3–5 that with the increase of keywords, the running time of the trapdoor, indexing and retrieval phases in this article, and the comparative literature show an increasing trend. Literature [23] has a higher running time cost with the increase of keywords in the three stages. The running time cost of this article and the literature [22] is relatively consistent, and its execution efficiency is relatively low. Compared with literature [22], the keyword ciphertext matching of this scheme belongs to exact matching, while literature [22] belongs to fuzzy matching. So, the keyword matching result of this scheme is more accurate than literature [22].

Table 2: Function comparison of different schemes.

Features	Literatures				
	Literature [11]	Literature [20]	Literature [22]	Literature [23]	This article
Blockchain	√	√	√	√	√
Access control	√	√	√	√	√
Privacy protection	√	√	√	√	√
SE	×	√	√	√	√
Data sharing	√	×	√	√	√
Deduplication	×	×	×	×	√

Table 3: Problems and the solutions of existing EMRs.

Types	Problems	Solutions
Systematisms of personal EMRs	Lack of systematic EMRs for the individuals	By updating EMRs, establishing systematic personal EMRs for patients
Privacy leaks	The personal EMRs information of patients is directly shared without encryption or is intercepted or forged by malicious attackers, etc. There is a risk of privacy leakage, and the privacy of patients cannot be guaranteed	Using SE technology to search ciphertexts to avoid privacy leakage, and using blockchain technology to ensure the immutability and integrity of data
Store	For the same patient, the same EMRs from the different hospital is repeatedly stored, which makes the storage space consumption high	Delete newly added duplicate data, only mark duplicate EMRs without repetitive storage, and add new cases to the original EMRs
Data sharing	The hospitals are relatively independent and have poor interaction. The EMRs of the same patient cannot be shared between hospitals in real time, and there is a problem of data islands	Establish an alliance chain between the hospitals to realize real-time EMRs data sharing
Access control permissions	Patients are unaware of personal case sharing and have no access control authority to personal medical records. The hospital can view and share patient data at any time without the patient knowing	Only the doctors authorized by the patient can view and update the patient's personal EMRs

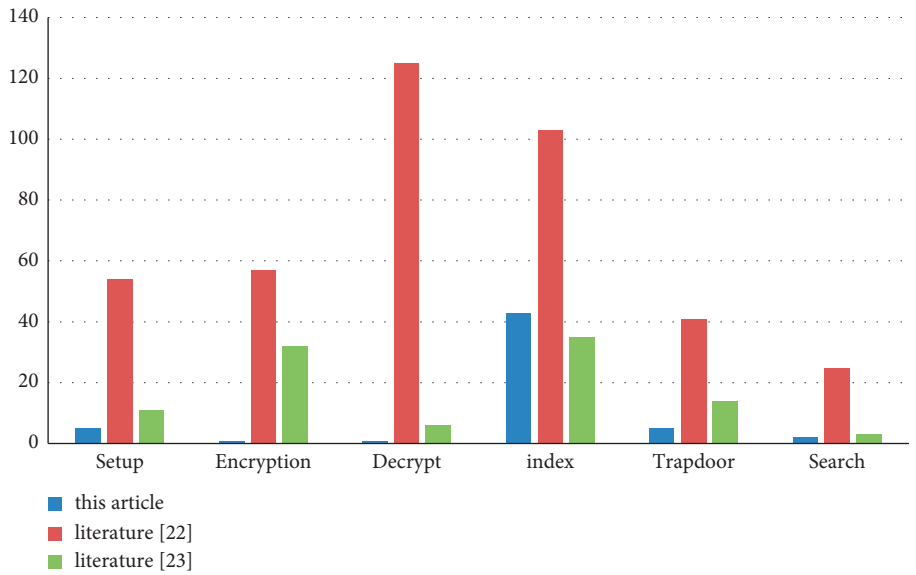


Figure 2: Comparison of the running time of each stage.

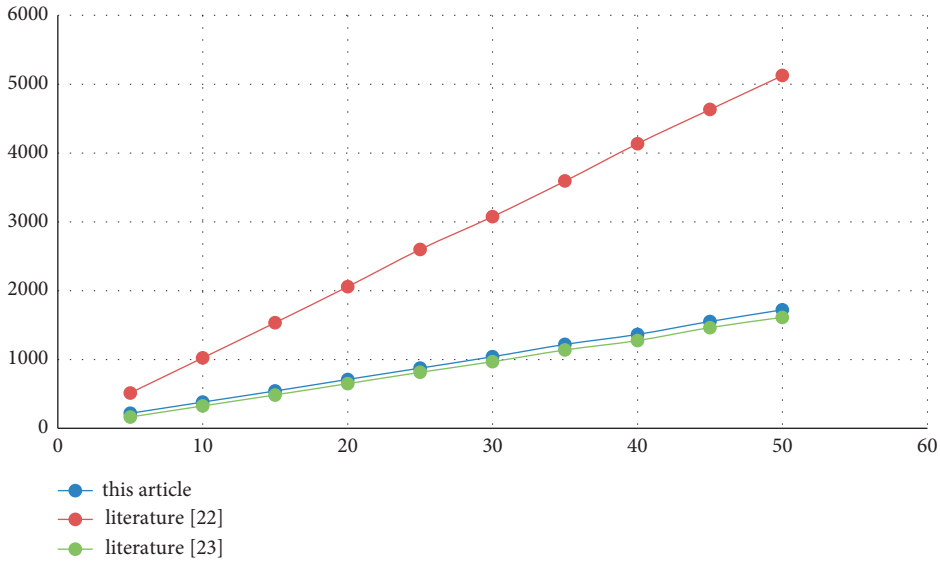


Figure 3: Comparison of index generation time.

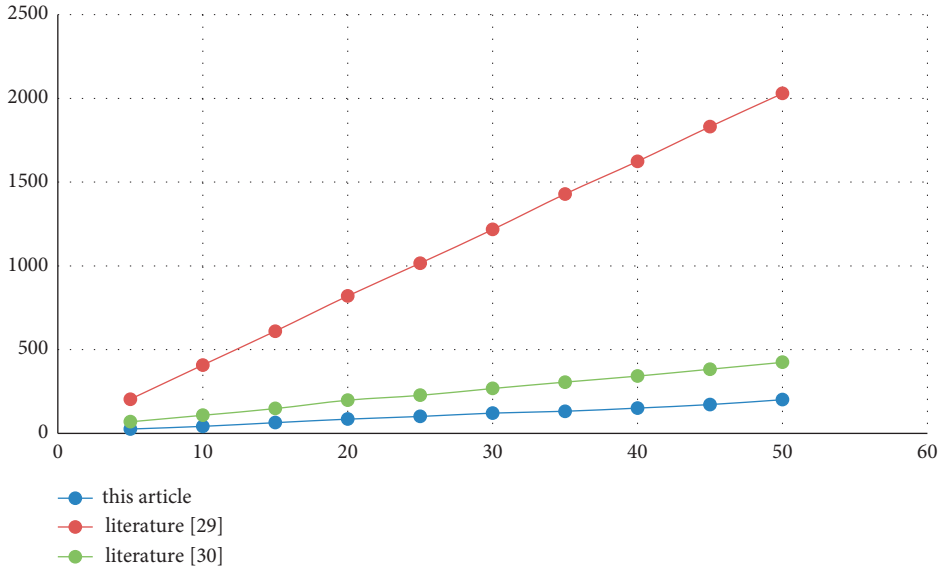


Figure 4: Comparison of trapdoor generation time.

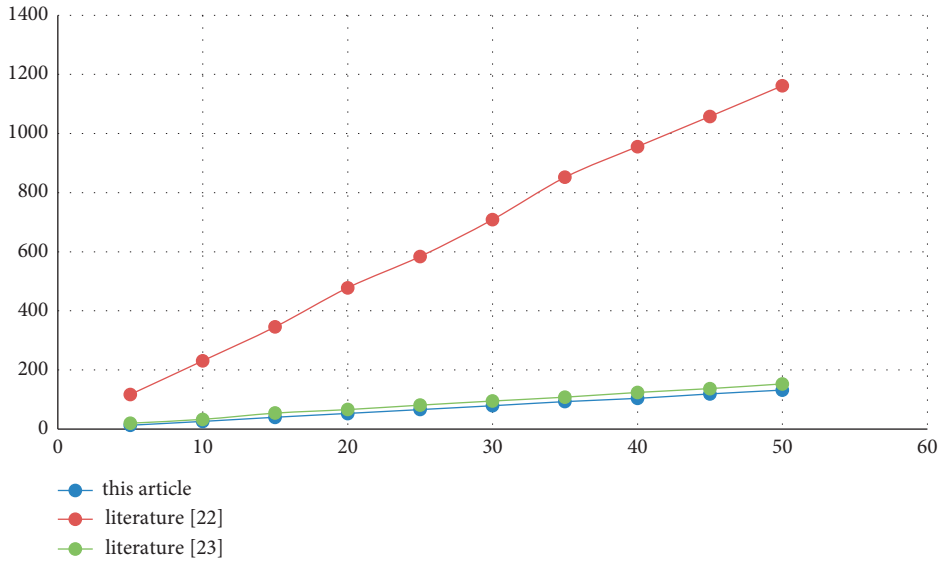


Figure 5: Comparison of retrieval time.

5. Conclusions

This article proposes a cross-domain sharing of EMRs among different hospitals based on blockchain and edge computing, which solves the difficulty of EMRs data sharing among hospitals and the problem of isolated and duplicated storage. Through patient authorization, cross-domain secure sharing of EMRs is realized and making the patient's personal EMRs more systematic and complete. The use of blockchain technology ensures that the data cannot be tampered with, and the use of searchable encryption ensures the security of EMRs and personal privacy. Edge servers offload the computing tasks of cloud services and improve computing efficiency. By analysis, it is found that the security of the scheme is proved based on the BDH assumption. Performance analysis and simulation experiments show that the computational complexity is relatively low and has high execution efficiency.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

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

Acknowledgments

This research was supported by the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (SKLNST-2020-1-09), Henan Key Research Projects of Universities (20A520043 and 21B520022), Natural Science Foundation of Henan Province (202300410510), and National Key Research and Development Program of China (2020YFB1005404).

References

[1] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned Internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5213–5222, 2021.

[2] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for Internet of Things," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.

[3] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for Internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[4] A. Shahnaz, U. Qamar, and A. Khalid, "Using blockchain for electronic health records," *IEEE Access*, vol. 7, Article ID 147795, 2019.

[5] P. B. Jensen, L. J. Jensen, and S. Brunak, "Mining electronic health records: towards better research applications and clinical care," *Nature Reviews Genetics*, vol. 13, no. 6, pp. 395–405, 2012.

[6] A. Hoerbst and E. Ammenwerth, "Electronic health records. A systematic review on quality requirements," *Methods of Information in Medicine*, vol. 49, no. 4, pp. 320–36, 2010.

[7] Y. Yuan and F. Y. Wang, "Blockchain: the state of the art and future trends," *Acta Automatica Sinica*, vol. 42, no. 4, pp. 481–494, 2016.

[8] Y. Yuan, X. C. Ni, S. Zeng, and F. Y. Wang, "Blockchain consensus algorithms: the state of the artand future trends," *Acta Automatica Sinica*, vol. 44, no. 11, pp. 2011–2022, 2018.

[9] X. Han, Y. Yuan, and F. Y. Wang, "Security problems on blockchain: the state of the art and future trends," *Acta Automatica Sinica*, vol. 45, no. 1, pp. 206–225, 2019.

[10] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control," *Journal of Medical Systems*, vol. 40, no. 10, pp. 218–226, 2016.

[11] Q. Xia, E. Sifah, A. Smahi, S. Amofa, and X. Zhang, "B. B. D. S.: BBDS: blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, pp. 44–60, 2017.

[12] C. Zhang, Q. Li, Z. H. Chen, Z. R. Li, and Z. Zhang, "Medical chain: alliance medical blockchain system," *Acta Automatica Sinica*, vol. 45, no. 8, pp. 1495–1510, 2019.

[13] Y. Chen, S. Ding, Z. Xu, H. Zheng, and S. Yang, "Blockchain-based medical records secure storage and medical service framework," *Journal of Medical Systems*, vol. 43, no. 1, pp. 5–14, 2019.

[14] W. Hao and Y. Song, "Secure cloud-based EHR system using attribute-based cryptosystem and blockchain," *Journal of Medical Systems*, vol. 18, no. 2, pp. 152–161, 2018.

[15] J. W. Li, C. F. Jia, Z. L. Liu, J. Li, and M. Li, "Survey on the SE," *Journal of Software*, vol. 26, no. 1, pp. 109–128, 2015.

[16] Z. R. Shen, W. Xue, and J. W. Shu, "Survey on the research and development of SE schemes," *Journal of Software*, vol. 25, no. 4, pp. 880–895, 2014.

[17] Y. L. Wang and X. F. Chen, "Research on searchable symmetric encryption," *Journal of Electronics and Information Technology*, vol. 54, no. 10, pp. 2374–2385, 2020.

[18] X. L. Dong, J. Zhou, and Z. F. Cao, "Research advances on secure SE," *Journal of Computer Research and Development*, vol. 54, no. 10, pp. 2107–2120, 2017.

[19] L. Chen, W. K. Lee, C. C. Chang, K. K. R. Choo, and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing," *Future Generation Computer Systems*, vol. 95, no. 2, pp. 420–429, 2019.

[20] A. Zhang and X. Lin, "Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain," *Journal of Medical Systems*, vol. 42, no. 8, pp. 140–158, 2018.

[21] Y. Wang, A. Zhang, P. Zhang, and H. Wang, "Cloud-Assisted EHR sharing with security and privacy preservation via consortium blockchain," *IEEE Access*, vol. 7, no. 2, Article ID 136719, 2019.

[22] S. F. Niu, L. X. Chen, W. T. Li, C. F. Wang, and X. N. Du, "Data sharing scheme based on blockchain," *Acta Automatica Sinica*, vol. 1-11, 2020.

[23] L. Zhang, Z. Y. Zhang, and Y. Yuan, "A controllable sharing model for electronic health records based on blockchain," *Acta Automatica Sinica*, vol. 1-14, 2020.

[24] B. Yang, *Modern Cryptography*, 4th edition, Tsinghua University Press, Beijing, China, 2017.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 1951141, 10 pages
https://doi.org/10.1155/2021/1951141

Research Article

A Game-Based Scheme for Resource Purchasing and Pricing in MEC for Internet of Things

Yajing Leng,[1] **Ming Wang**,[1] **Bowen Ma**,[1] **Ying Chen**,[2] **and Jiwei Huang**[1]

[1]*Beijing Key Laboratory of Petroleum Data Mining, China University of Petroleum-Beijing, Beijing 102249, China*
[2]*Computer School, Beijing Information Science and Technology University, Beijing 100101, China*

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

Received 6 September 2021; Revised 11 October 2021; Accepted 21 October 2021; Published 12 November 2021

Academic Editor: Xuyun Zhang

Mobile edge computing (MEC) is emerging as a promising paradigm to support the applications of Internet of Things (IoT). The edge servers bring computing resources to the edge of the network, so as to meet the delay requirements of the IoT devices' service requests. At the same time, the edge servers can gain profit by leasing computing resources to IoT users and realize the allocation of computing resources. How to determine a reasonable resource leasing price for the edge servers and how to determine the number of resource purchased by users with different needs is a challenging problem. In order to solve the problem, this paper proposes a game-based scheme for resource purchasing and pricing aiming at maximizing user utility and server profit. The interaction between users and the edge servers is modeled based on Stackelberg game theory. The properties of incentive compatibility and envy freeness are theoretically proved, and the existence of Stackelberg equilibrium is also proved. A game-based user resource purchasing algorithm called GURP and a game-based server resource pricing algorithm called GSRP are proposed. It is theoretically proven that solutions of the proposed algorithms satisfy the individual rationality property. Finally, simulation experiments are carried out, and the experimental results show that the GURP algorithm and the GSRP algorithm can quickly converge to the optimal solutions. Comparison experiments with the benchmark algorithms are also carried out, and the experimental results show that the GURP algorithm and the GSRP algorithm can maximize user utility and server profit.

1. Introduction

With the rapid development of Internet of Things (IoT) technology, various IoT devices such as smart phones and vehicles have been connected to the Internet [1, 2]. Service requests generated by IoT devices usually have strict requirements for computing resources and real-time processing [3]. Because IoT devices usually do not have enough computing resources [4], they usually offload service requests to the cloud for computing [5]. Generally speaking, large data processing centers or cloud servers are usually built in remote areas away from users. Therefore, when the service requests are offloaded to the cloud for computing, it will result in a lot of transmission costs and service delay. This is intolerable for IoT services that require high real-time performance.

To solve this problem, mobile edge computing (MEC) is proposed. MEC provides users with short-range cloud computing services by deploying edge servers [6]. In MEC, users can offload service requests to the network edge for calculation [7]. The edge servers are close to users and have rich computing resources. Compared with the public cloud, the edge cloud is closer to the IoT devices, which can meet the requirements of IoT applications for low latency [8]. Because the user service request does not need to be transmitted to the remote cloud for calculation through the Internet, the transmission delay is reduced. In recent years, with the development of the IoT, a huge number of service requests have been offloaded to edge servers for computing [9]. Therefore, more and more edge cloud service providers came into being [10].

Although MEC can help provide resources for IoT applications, it faces unprecedented challenges. With the development of the IoT market, more and more different types of users will access the IoT networks [11]. Different users

have different purchasing needs for resources. Compared with the public cloud, there are some restrictions on the computing resources on the edge servers, which cannot meet the resource needs of all users. Therefore, how to reasonably allocate resources is a main challenge faced by MEC.

Reasonable pricing of resources can be used to solve the above problems. The servers price the provided computing resources and publish it to the users. Users choose appropriate resources to purchase according to the resource price of the servers and process the service request on the servers, so as to realize the reasonable allocation of resources. Therefore, the current resource pricing scheme in MEC needs to balance and meet the needs of different types of users.

In this work, we focus on resource purchasing under the condition of maximizing user utility and server profit. Its operation mechanism is as follows: the servers publish the resource leasing price, and then the users determine the number of resource purchasing. The servers obtain the resulting profit and repeatedly modify the leasing price in game. When the game equilibrium is reached, both the pricing of the servers and the resource purchasing of the users will be optimal.

Our contributions are summarized as follows:

(i) We consider the scenario of an MEC system with multiple IoT device users and an edge server. Each user can purchase computing resources from the edge server and offload the service requests to the edge server for computing. We study the problem of resource purchasing and resource pricing from the perspective of users and servers and establish both the user utility function and the server profit function. The goal is to optimize both the user utility and the server profit together.

(ii) We establish a Stackelberg game model to represent the interaction process of resource purchasing and resource pricing between multiple users and the server. The existence of Stackelberg equilibrium point is theoretically proved. It is also proved that the properties of incentive compatibility and envy freeness are satisfied. Then, we propose a game-based user resource purchasing algorithm (GURP) and a game-based server resource pricing algorithm (GSRP) which can obtain the optimal solution of Stackelberg equilibrium. We propose the theorem that the individual rationality property is satisfied.

(iii) In order to verify the performance of our GURP and GSRP algorithms, we carry out simulation experiments. Experimental results show that the algorithms can eventually converge to the optimal solution. In addition, in terms of resource pricing and resource purchasing, two groups of comparison experiments with the benchmark algorithms are carried out. The results show that the GURP and GSRP algorithms can obtain the maximum user utility and server profit.

The remainder of this paper is organized as follows. We present the system model and relevant problem formulation in Section 2. We construct Stackelberg game model to analyze the interaction between users and servers and propose the GURP and GSRP algorithms in Section 3. We evaluate the performance of our GURP and GSRP algorithms in Section 4. The related works are reviewed in Section 5. The conclusion is given in Section 6.

2. System Model and Problem Formulation

2.1. System Model. An MEC system for the IoT considered in this paper consists of one edge server, denoted by S, and a set of users, denoted by U. Users can lease and purchase computing resources on the edge server and offload service requests to the edge server for computing. This can overcome the problem of insufficient local computing power of users. The edge server provides computing resource leasing services to users within their signal coverage in order to obtain profit. In this paper, we assume that the resources on the edge server can meet the needs of all users in its coverage.

We consider a game-based scene for resource purchasing and pricing in MEC shown in Figure 1. As mentioned earlier, at different times, users accessing the IoT have different needs and satisfactions with resources [12]. If the edge server always adopts a single resource pricing, it will have an impact on resource allocation and market economy.

Therefore, from the perspective of users and server, based on game theory, this paper determines the resource purchasing and resource pricing scheme that can optimize user utility and server profit.

2.1.1. User Utility. There are totally N users who propose the service requests, denoted by $U = \{u_1, u_2, u_3, \ldots, u_N\}$. We assume that each user u_i proposes a service request. The service request of user u_i is specified as a tuple (C_i, T_i^m). C_i represents the calculated size of u_i service request. T_i^m indicates the longest service request completion time acceptable to u_i.

We consider that all user's service requests must be transmitted before starting computing. Thus, the transmission time of the service request from user i to server is

$$T_i^s = \frac{C_i}{b}, \tag{1}$$

where b is the transfer rate.

According to the source price p published by the edge server, user i determines its resource purchasing strategy, which is denoted by a_i. The computing time of the service request from user i is

$$T_i^c = \frac{C_i \beta}{a_i f}, \tag{2}$$

where β represents the cycles per bit for computing one sample data of user and f represents the CPU frequency of a single resource in the edge server.

We define T_i as the actual completion time of the user's service request:

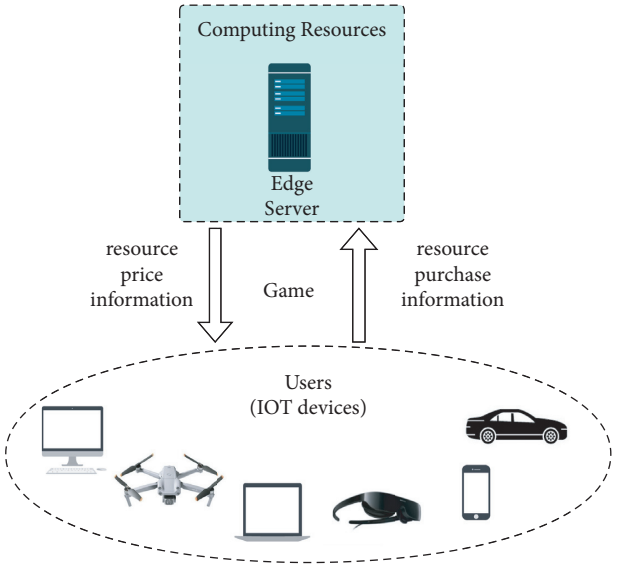


Figure 1: A game-based scene for resource purchasing and pricing in MEC.

$$T_i = T_i^s + T_i^c = \frac{C_i}{b} + \frac{C_i\beta}{a_i f}. \tag{3}$$

The user utility of u_i is defined as

$$V_i = \alpha_i \log\left(\frac{T_i^m}{T_i}\right) - \mathrm{pa}_i = \alpha_i \log\left(\frac{T_i^m}{C_i/b + C_i\beta/a_i f}\right) - \mathrm{pa}_i. \tag{4}$$

where α_i indicates user's u_i satisfaction with renting server resources. The higher the user satisfaction is, the more the users tend to purchase more server resources for service request calculation [13]. pa_i represents the cost of user purchases edge server's resource.

2.1.2. Edge Server Profit. In addition to leasing computing resources to users, the edge server also needs to maintain computing resources. The edge server profit function can be defined as

$$M_s = p\sum_{i=1}^{n} a_i - q\sum_{i=1}^{n} a_i, \tag{5}$$

where q denotes the maintenance cost of the server to a single computing resource.

For the server, it only needs to maintain the resources leased to users. Other computing resources not leased to users will not incur maintenance costs.

The main notations and their definitions used in the following discussion are given in Table 1.

Table 1: Summary of key notations.

Notation	Definition
u_i	User i
α_i	u_i satisfaction with renting server resources
T_i^m	Longest service request completion time acceptable to u_i
T_i	Actual completion time of the u_i service request
a_i	u_i purchases the number of resources from the server
β	Cycles per bit for computing one sample data of user
C_i	Calculated size of u_i service request
p	Price of a single server resource
f	CPU frequency of an edge server's single resource
b	Service request transfer rate
q	Maintenance cost of a server to a single resource
V_i	User u_i utility
M_s	Edge server S profit

2.2. Problem Formulation. We formulate the scheme for resource purchasing and pricing as a Stackelberg game. We divide the whole game process into two stages. In the first stage, the edge server determines its own resource pricing scheme. In the second stage, each user determines its resource purchasing strategy to maximize its own user utility. Therefore, in the process of this game, the edge server is a leader and users are followers. The strategy of the edge server is the source price p and the strategy of user is the number of resource purchasing, which is denoted by a_i. For the arbitrary pricing p of the edge server, user i will determine an optimal resource purchasing strategy to optimize its user utility, i.e.,

$$\begin{aligned} &\max V_i, \\ &s.t.\ V_i \geq 0, \\ &a_i \geq 0. \end{aligned} \tag{6}$$

The edge server will also update the pricing information according to users' resource purchasing strategies to pursue maximum profit, i.e.,

$$\begin{aligned} &\max M_s, \\ &s.t.\ M_s \geq 0. \end{aligned} \tag{7}$$

3. Game for Purchasing and Pricing Scheme

3.1. User Utility Optimization. User i needs to determine appropriate resource purchasing strategy according to the resource price of the edge server to maximize its own user utility. The problem is defined as follows:

$$\begin{aligned} Q_1 &= \max\{V_i\}, \\ V_i &= \alpha_i \log\left(\frac{T_i^m}{C_i/b + C_i\beta/a_i f}\right) - \mathrm{pa}_i. \end{aligned} \tag{8}$$

The first derivative of V_i with regard to a_i is given by

$$\frac{\mathrm{d}V_i}{\mathrm{d}a_i} = \frac{\alpha_i \beta b}{\ln 2\left(a_i^2 f + a_i \beta b\right)} - p. \tag{9}$$

The second derivative of V_i with regard to a_i is given by

$$\frac{\mathrm{d}^2 V_i}{\mathrm{d}a_i^2} = -\frac{1}{\ln 2}\frac{\alpha_i \beta b\left(2a_i f + \beta b\right)}{a_i^4 f^2 + a_i^2 \beta^2 b^2 + 2a_i^3 f \beta b} < 0. \tag{10}$$

Because the second derivative of V_i with regard to a_i is always negative, the function of V_i is a convex function. Q_1 can be regarded as a convex optimization problem, and its optimal solution is

$$\frac{\mathrm{d}V_i}{\mathrm{d}a_i} = 0, \tag{11}$$

i.e.,

$$a_i^* = \begin{cases} \dfrac{-\beta b + \sqrt{\beta^2 b^2 + 4 f \alpha_i \beta b / p \ln 2}}{2f}, & V_i \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

When the resource unit price p is fixed, in order to obtain the maximum utility, the user purchases the number of resources as shown in (12). When the user utility value $V_i \geq 0$, the user chooses to purchase server resources and offload the service request to the server for calculation. Otherwise, the user will calculate the service request locally.

Theorem 1 (incentive compatibility). *Users can truly report resource purchasing strategies. Users cannot obtain higher user utility by reporting false strategies.*

Proof. As proved earlier, user u_i determines the resource purchasing strategy a_i^* according to the resource unit price p formulated by the edge server. a_i^* is the unique maximizer of the user utility in equation (4). Then, its utility function satisfies

$$V_i\left(a_i^*\right) \geq V_i\left(a_i\right). \tag{13}$$

Therefore, user will not obtain better user utility through false reporting strategy. The user has no incentive to misreport its strategy. So, there exists incentive compatibility. □

Theorem 2 (envy freeness). *The user always prefers its own purchased number of resources to that of others.*

Proof. In the system model proposed in this paper, users are independent of *each other. All users can determine their resource purchasing strategies according to the price of edge server, so as to obtain the optimal user utility. The utility function of users only depends on their own resource purchasing strategies and resource unit price formulated by the edge server. Each user's resource purchasing strategy is optimal for itself. Therefore, users will not envy the strategies of other users.* □

3.2. Edge Server Profit Maximization. An edge server makes profit by leasing its computing resources. The server achieves the goal of maximum profit by adjusting its resource unit price. The problem is defined as follows:

$$\begin{aligned} Q_2 &= \max\{M_s\}, \\ M_s &= p\sum_{i=1}^{n} a_i - q\sum_{i=1}^{n} a_i, \end{aligned} \tag{14}$$

where the value of a_i is determined by equation (12).

The first derivative of M_s with regard to p is given by

$$\frac{\mathrm{d}M_s}{\mathrm{d}p} = \sum_{i=1}^{n} \frac{\sqrt{b}\left(2fq\alpha_i + b \ln 2\, \beta p^2 + 2fp\alpha_i\right)}{2f\sqrt{\ln 2}\sqrt{\beta} p^{3/2}\sqrt{\ln 2\, b\beta p + 4f\alpha_i}} - n\frac{b}{2f}. \tag{15}$$

The second derivative of M_s with regard to p is given by

$$\frac{\mathrm{d}^2 M_s}{\mathrm{d}p^2} = \sum_{i=1}^{n} -\frac{\sqrt{\beta b}\left(\left(2\beta b\alpha_i p \ln 2 + 6f\alpha_i^2\right)q + 2f\alpha_i^2 p\right)}{\sqrt{\ln 2}\sqrt{p}\sqrt{\beta b p \ln 2 + 4f\alpha_i}\left(\beta b \ln 2 p^3 + 4f\alpha_i p^2\right)} < 0. \tag{16}$$

Because the second derivative of M_s with respect to p is always negative, the function of M_s is a convex function. $p \longrightarrow 0, M_s < 0$; $p \longrightarrow \infty, M_s = 0$. Thus, Q_2 can be regarded as a convex optimization problem, and it has a unique optimal solution p^*. The optimal solution p^* is related to the satisfaction of leasing resources of each user (α_i).

3.3. Stackelberg Equilibrium. For users and the edge server, in the game model, the existence of Stackelberg equilibrium can be proved by the existence of optimal solutions for problems Q_1 and Q_2. This not only ensures that the edge server can get the optimal profit but also ensures that users can get the optimal utility.

Theorem 3. *For the edge server, there is an optimal resource price p^*, which makes the server profit optimal. u_i has an optimal resource purchasing strategy a_i^*, which makes the user utility optimal. Then, it can be explained that the game model has a Stackelberg equilibrium, i.e.,*

$$\begin{aligned} M_s\left(p^*\right) &\geq M_s\left(p\right), \\ V_i\left(a_i^*\right) &\geq V_i\left(a_i\right). \end{aligned} \tag{17}$$

Proof. It can be obtained from formula (16) *that the edge server profit M_s is a convex function with regard to resource price p.* Therefore, the edge server can get an optimal resource pricing strategy, so as to maximize the server profit. For a certain resource price, according to (12), user can make an optimal source purchasing strategy to maximize personal utility. Therefore, there is Stackelberg equilibrium in the game model. □

3.4. Algorithm Design

3.4.1. Game-Based User Resource Purchasing Algorithm. We propose the game-based user resource purchasing (GURP) algorithm as shown in Algorithm 1. For each user, in each game with the server, they will first accept the

resource price p information published by the server. The user determines the resource purchasing strategy that maximizes the user's utility according to formula (12). Due to individual rationality, each user will judge whether its utility value is greater than 0. If the utility value is less than 0, the user will give up purchasing computing resources. On the contrary, the user determines the resource purchasing strategy and reports the strategy information to the server.

The specific process is as follows. In line 2, we initially set the user utility values of all users to 0. In lines 4–14, each user sets resource purchasing strategy according to (12). Finally, user will make a resource purchasing strategy to maximize its user utility and report the resource purchasing strategy to the edge server.

3.4.2. Game-Based Server Resource Pricing Algorithm. We propose the game-based server resource pricing (GSRP) algorithm as shown in Algorithm 2. For the server, at the beginning of the game, set a small resource pricing information and publish the information to the user. Then, the server accepts the purchase information of users and calculates its own profit. Next, the server sets an appropriate resource price update step. It continuously updates the resource pricing information and publishes it to users and accepts the user's purchase information and calculates the profit. In the iterative process of game between the server and the users, the server's pricing strategy will converge to the price that maximizes the profit of server.

The specific process is as follows. In lines 1–5, we initialize the variable values in the Algorithm 2. In lines 6–12, the edge server will constantly update the resource price to maximize the server profit and record the optimal resource price. Finally, the edge server will get the resource price which can maximize the profit.

Both Algorithm 1 and 2 have high computational efficiency. For GURP, as shown in Algorithm 1, for line 4, because of n users participating, it needs to cycle n times for calculation. Therefore, the computational complexity of Algorithm 1 is $\mathcal{O}(N)$. For GSRP, as shown in Algorithm 2, for lines 6–12, the number of iterations for the convergence of server profit is limited. We use M to represent the number of iterations, so its computational complexity is $\mathcal{O}(M)$. Then, from line 10, the computational complexity is $\mathcal{O}(N)$. Therefore, the computational complexity of Algorithm 2 is $\mathcal{O}(MN)$.

Theorem 4. *Both GURP and GSRP mechanisms satisfy individual rationality.*

Proof. Individual rationality means that no one will suffer from participating in the sale mechanism. For resource purchasers (users), in GURP, as shown in Algorithm 1, for line 6, users will purchase resources on the premise that their user utility is greater than 0. For resource seller (edge server), in GSRP, as shown in Algorithm 2, for line 6, the resource price set by the edge server must make its profit greater than 0. Therefore, both GURP and GSRP mechanisms satisfy individual rationality. □

4. Performance Evaluation

4.1. Setup. We conduct simulation experiments and use simulation data to verify the game algorithm proposed in this paper. In the experimental scenario, there are six users and an edge server. We set the maintenance cost of a server to a single resource q as 1 [14] and set the CPU frequency of an edge server's single resource f as 4 GHz [15]. The satisfaction of u_i is set to 50–100 [16].

To be specific, the main parameters involved in this experiment are shown in Table 2.

4.2. Parametric Analysis. The first set of experiments is to investigate the change of edge server profit in the game. From the result shown in Figure 2, we can know that the profit of the server will converge to the equilibrium point of the game with the increase of the number of games. The convergence rate is related to the update step of p. When step Δp is small, the server profit will have to go through multiple rounds of iteration to reach the convergence point. When step Δp is big, the server profit value may miss the convergence point.

The game iteration between users and the server takes some time and energy. Too many iterations may cause users to give up resource purchasing because they cannot stand the consumption of time and energy. Too few iterations may cause the server to miss the optimal pricing strategy and damage the profit of the service provider. Therefore, in an actual scenario, the update step size of the server resource pricing strategy will have an impact on the final profit of the server. The server needs to formulate a reasonable resource unit price update step.

Figure 3 shows the change of user's utility value with the number of iterations. We study and analyze the change of utility value of six users with different satisfaction and different service request sizes. The user's satisfaction relationship is $\alpha_1 < \alpha_2 < \alpha_3 < \alpha_4 < \alpha_5 < \alpha_6$. At the beginning, the server makes the price of resources very low, so users will have high user utility. With the progress of the game, the server will gradually formulate optimal resource price, so user's utility will gradually decrease and reach the equilibrium point of the game. The higher the user's satisfaction with the leased resources is, the more the user tends to purchase more edge computing resources to obtain greater user utility. The higher the satisfaction of users, the higher the user's utility after reaching the equilibrium point of the game.

We specifically study the impact of satisfaction on user's utility and server's profit. From the result shown in Figure 4, for users, users with high satisfaction tend to buy more computing resources at the edge server, so as to obtain greater user's utility. For the server, simultaneously, providing more resources to the group of users with high satisfaction can bring in higher profits.

Finally, Figure 5 illustrates the impact of service request transfer rate and CPU frequency on server's profit. It is shown that as the service request transfer rate and CPU frequency increase, the server's profit becomes higher. The server provides high transmission rate channel and high

Input: p
Output: *optimal* a_i^* *for each* u_i
(1) ***for*** *each* u_i *in* U ***do***
(2) $V_i = 0$;
(3) ***end***
(4) ***for*** *each* u_i *in* U ***do***
(5) $a_i = -\beta b + \sqrt{\beta^2 b^2 + 4 f \alpha_i \beta b / p \ln 2}/2f$
(6) ***if*** $V_i > 0$ ***then***
(7) $a_i^* = -\beta b + \sqrt{\beta^2 b^2 + 4 f \alpha_i \beta b / p \ln 2}/2f$;
(8) $V_i = \alpha_i \log(T_i^m / C_i / b + C_i \beta / a_i^* f) - p a_i^*$;
(9) *else*
(10) $a_i^* = 0$;
(11) $V_i = 0$;
(12) *end*
(13) *end*
(14) *return* a_i^* *for each* u_i;

Algorithm 1: Game-based user resource purchase algorithm (GURP).

Input: n, $\sum_{i=1}^{n} a_i$
Output: *optimal price* p^*
(1) $M_s^* = 0$;
(2) $p^* = 0$;
(3) q = *maintenance cost;*
(4) p = *initial price;*
(5) $M_s = p \sum_{i=1}^{n} a_i - q \sum_{i=1}^{n} a_i$;
(6) ***while*** $M_s > M_s^*$ ***do***
(7) $M_s^* = M_s$;
(8) $p^* = p$;
(9) $p = p + \Delta p$; *(update p)*
(10) *GURP (p);*
(11) $M_s = p \sum_{i=1}^{n} a_i - q \sum_{i=1}^{n} a_i$;
(12) *end*
(13) *return* p^*;

Algorithm 2: Game-based server resource pricing algorithm (GSRP).

Table 2: Experiment setup.

Parameters	Value
α_i (u_i satisfaction)	[50, 100]
C_i (calculated size of u_i service request)	[60, 80] bit
f (CPU frequency of an edge server's single resource)	4 GHz
β (cycles per bit for computing one sample data of user)	20 cycles/bit
b (transfer rate)	1×10^8 bit/s

computing power resources, which can attract users to purchase more computing resources, so as to improve the profit of the server.

4.3. Comparison Experiment. In this part, aiming at the problem of resource pricing and resource purchasing, we compare and analyze many different resource pricing and resource purchasing algorithms and further evaluate the performance of the game algorithm (GURP and GSRP) proposed in this paper.

4.3.1. Resource Pricing. The server leases resources to users, determines the resource leasing price, and obtains profit by charging users a fee. For server's resource pricing, we compare three pricing strategies:

(i) Random resource pricing: the server randomly makes a resource leasing price.

(ii) Historical optimal resource pricing: the server queries the historical optimal resource pricing and takes it as the current resource pricing scheme.

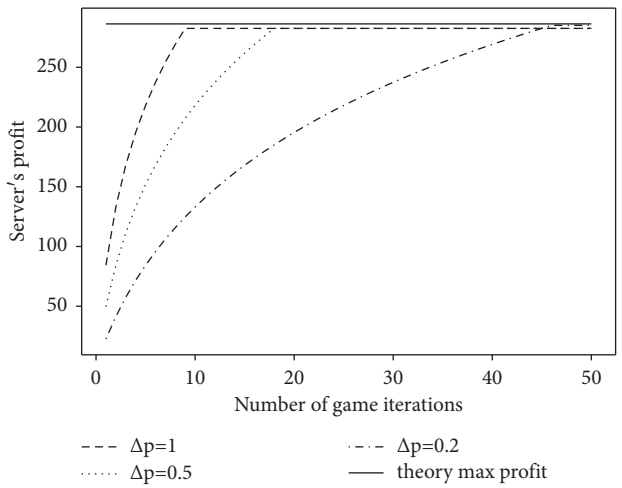


Figure 2: The profit of the server in game process.

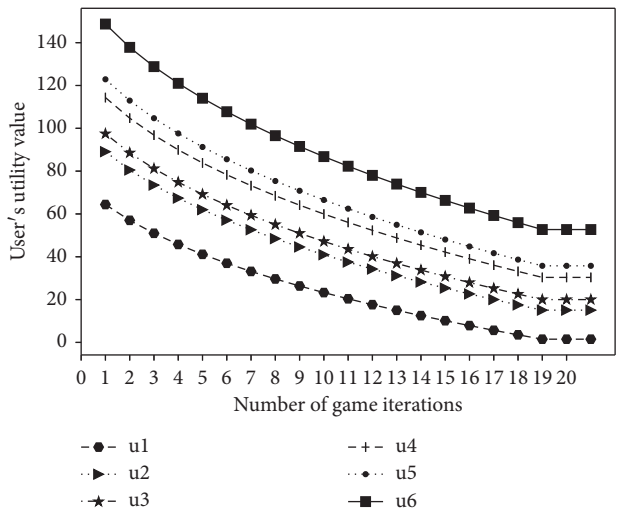


Figure 3: The utility values of users in game process.

(iii) Game-based server resource pricing (GSRP): in the process of game with users, the server continuously updates the price until the optimal resource pricing scheme is given.

Figure 6 shows the server's profit with different resource pricing strategies. The experimental result shows that the profit of the server is different for the user groups with different satisfaction. With the increase of user satisfaction, the profit of server also increases. The game-based pricing algorithm proposed in this paper is superior to the other two kinds of pricing algorithms in maximizing server profit. The GSRP algorithm can find the most suitable resource price for the current user group and obtain the maximum profit in the process of game with users.

4.3.2. Resource Purchasing. In order to maximize user utility, users purchase appropriate edge computing resources to calculate service requests. For user's resource purchasing, we compare three resource purchasing strategies:

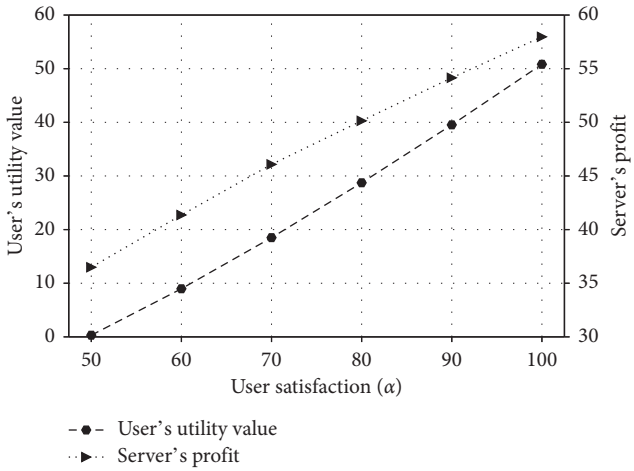


Figure 4: Impact of user's satisfaction.

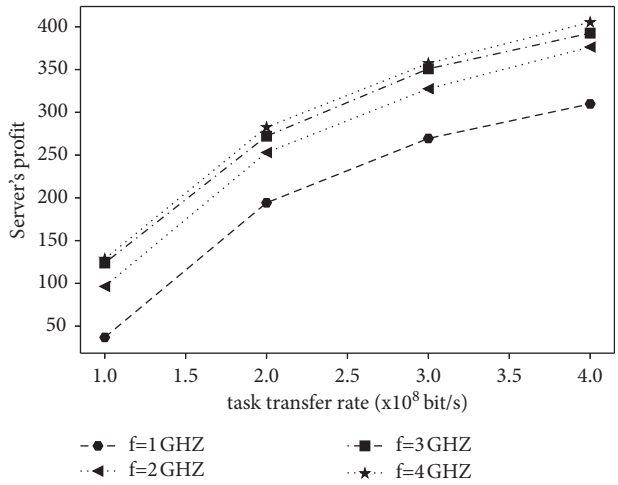


Figure 5: Impact of calculation frequency and transfer rate.

(i) Random resource purchasing: user randomly purchases a certain number of resources on the edge server.

(ii) Fixed resource purchasing: user purchases a fixed number of resources on the edge server according to the size of its service request. User with larger service request will purchase more edge server resources for calculation.

(iii) Game-based user resource purchasing (GURP): in the process of game with server, user determines the resource purchasing strategy to maximize its user's utility according to the resource pricing of the edge server.

Figure 7 shows the utility values of users with different resource purchasing strategies. The experimental result shows that the user's utility is different for the users with different satisfaction. The more satisfied the users are, the more inclined they are to participate in the resource purchasing market, so as to obtain greater utility. The game-based resource purchasing algorithm proposed in this paper is superior to the other two kinds of resource purchasing

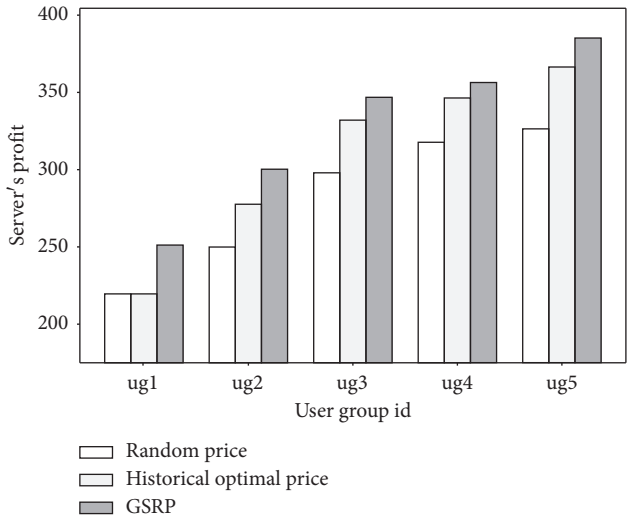


Figure 6: The profit of the server with different pricing strategies.

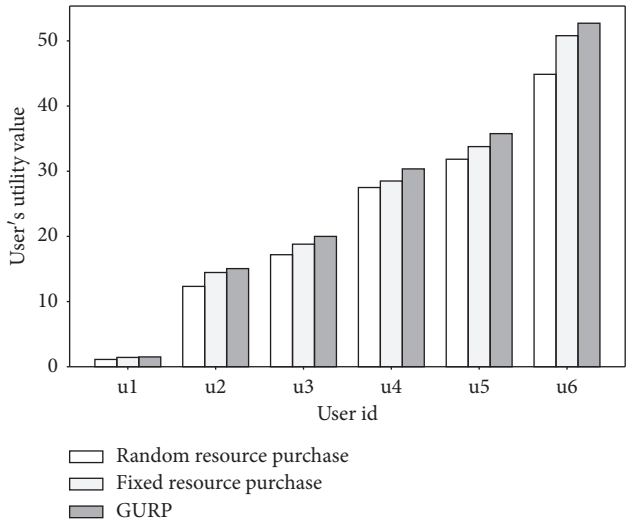


Figure 7: The utility values of users with different resource purchasing strategies.

algorithms in maximizing user's utility. The GURP algorithm can determine the best resource purchasing strategy according to the real-time resource price and obtain the optimal user utility.

5. Related Works

In recent years, with the development of the IoT, many research studies focus on its resource allocation and resource pricing problems [17].

In [18], the authors adopt the machine learning method for the first time and obtain a market model by using big data. Through the model, servers can get an optimal resource pricing scheme. Siew et al. [19] mainly studied the resource allocation in MEC. From the perspective of maximizing servers' profits, the authors designed two dynamic resource pricing mechanisms for resource allocation. In [20], from the perspective of users, users can independently select server resources by judging the acceptable service request price and time cost to realize resource allocation. In [21], the authors introduced a Petri net and proposed a resource allocation strategy based on pricing time. In [22], the authors studied a fog calculation scene. By judging the priority of users requests, the servers formulate relevant pricing schemes and resource allocation schemes.

Due to the resource leasing and resource purchasing behavior between servers and users, some studies focus on the economic problems in MEC. In [23], in order to solve the resource allocation problem in MEC, the authors proposed two dynamic pricing double auction algorithms. From the perspective of maximizing social welfare, the authors in [24] introduced a broker between users and servers to manage market purchasing and pricing behavior and proposed an iterative bilateral auction scheme. In [25], Stackelberg game was applied to the task offloading for mobile blockchain. The authors proposed a dual auction game mechanism to obtain the optimal resource price and equipment resource demand.

The Stackelberg game theory is widely used to solve the problem of resource management. The authors in [16] studied a scenario of Internet of vehicles, and the real-time pricing problem of computing resources in Internet of vehicles was solved by using Stackelberg game. In [26], the authors studied the allocation of computing resources in multiple edge clouds and ToT devices and proposed a computing offload mechanism based on two-stage Stackelberg game. Zhang et al. [27] proposed a distributed algorithm for resource allocation based on Stackelberg game.

However, few studies pay attention to the problem that users actively determine the number of resources to purchase according to the real-time resource pricing of servers in MEC. This paper studies the resource purchasing and resource pricing scheme in MEC from the perspective of Stackelberg game theory. The Stackelberg game theory is used to model the interaction between edge servers and users, and the existence of Stackelberg equilibrium is proved. In addition, the resource purchasing and resource pricing algorithms based on game theory are proposed.

6. Conclusion

In this paper, we investigate a game-based scheme for resource purchasing and pricing in MEC for IoT. Based on Stackelberg game, the server and users can update their resource pricing strategy and their resource purchasing strategies continuously during the game. The models of optimal user utility and server profit are given. We propose a game-based scheme for resource purchasing and pricing, and the existence of Stackelberg equilibrium is proved. Finally, the algorithm is evaluated by simulation experiments. The experiment results demonstrate that user utility value and edge server profit obtained by this algorithm are better than other basic resource purchasing and resource pricing algorithms. For our future work, we will consider unloading part of the user's service request to the edge server for calculation. Service requests will be calculated in parallel on local and edge servers. Users can maximize their user utility

values by determining the size of offloading service requests and the number of resource purchasing.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This study was supported by the National Natural Science Foundation of China (nos. 61972414, 61902029, and 61973161), Beijing Nova Program (no. Z201100006820082), Beijing Natural Science Foundation (no. 4202066), Fundamental Research Funds for Central Universities (no. 2462018YJRC040), Excellent Talents Projects of Beijing (no. 9111923401), and Scientific Research Project of Beijing Municipal Education Commission (no. KM202011232015).

References

[1] Y. Liu, D. Li, S. Wan et al., "A long short-term memory-based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, 2021.

[2] J. Mabrouki, M. Azrour, G. Fattah, D. Dhiba, and S. E. Hajjaji, "Intelligent monitoring system for biogas detection based on the internet of things: mohammedia, Morocco city landfill case," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 10–17, 2021.

[3] S. Chen, L. Zhang, Y. Tang et al., "Indoor temperature monitoring using wireless sensor networks: a SMAC application in smart cities," *Sustainable Cities and Society*, vol. 61, Article ID 102333, 2020.

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

[5] W. Zhang, X. Chen, and J. Jiang, "A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems," *Tsinghua Science and Technology*, vol. 26, no. 1, pp. 95–111, 2020.

[6] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "Energy efficient dynamic offloading in mobile edge computing for Internet of Things," *IEEE Transactions on Cloud Computing*, vol. 9, 2019.

[7] R. Bi, Q. Liu, J. Ren, and G. Tan, "Utility aware offloading for mobile-edge computing," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 239–250, 2020.

[8] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "TOFFEE: task offloading and frequency scaling for energy efficiency of mobile devices in mobile edge computing," *IEEE Transactions on Cloud Computing*, vol. 2019, Article ID 292369, 2019.

[9] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4925–4934, 2020.

[10] D. Kim, J. Son, D. Seo, Y. Kim, H. Kim, and J. T. Seo, "A novel transparent and auditable fog-assisted cloud storage with compensation mechanism," *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 28–43, 2019.

[11] Z. Tong, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[12] Y. N. Malek, M. Najib, M. Bakhouya, and M. Essaaidi, "Multivariate deep learning approach for electric vehicle speed forecasting," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 56–64, 2021.

[13] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Transactions on Network Science and Engineering*, vol. 2020, Article ID 2969489, 2020.

[14] X. Yu and L. Tang, "Competition and cooperation between edge and remote clouds: a stackelberg game approach," in *Proceedings of the 2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pp. 1919–1923, IEEE, Chengdu, China, December 2018.

[15] T. Mahn, H. Al-Shatri, and A. Klein, "Distributed algorithm for energy efficient joint cloud and edge computing with splittable tasks," in *Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–7, IEEE, Marrakesh, Morocco, April 2019.

[16] C. Tang, C. Zhu, H. Wu, X. Wei, Q. Li, and J. J. Rodrigues, "A game theoretical pricing scheme for vehicles in vehicular edge computing," in *Proceedings of the 2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 17–22, IEEE, Tokyo, Japan, December 2020.

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

[18] D. Niyato, M. A. Alsheikh, P. Wang, D. I. Kim, and Z. Han, "Market model and optimal pricing scheme of big data and internet of things (IoT)," in *Proceedings of the 2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Kuala Lumpur, Malaysia, May 2016.

[19] M. Siew, D. Cai, L. Li, and T. Q. S. Quek, "Dynamic pricing for resource-quota sharing in multi-access edge computing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2901–2912, 2020.

[20] L. Ni, J. Zhang, and J. Yu, "Priced timed Petri nets based resource allocation strategy for fog computing," in *Proceedings of the 2016 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI)*, pp. 39–44, IEEE, Beijing, China, October 2016.

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

[22] A. Sutagundar and S. B. Shahapur, "Development of fog based dynamic resource allocation and pricing model in IoT," in *Proceedings of the 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 349–354, IEEE, Bangalore, India, August 2018.

[23] W. Sun, J. Liu, Y. Yue, and H. Zhang, "Double auction-based resource allocation for mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4692–4701, 2018.

[24] Z. Li, Z. Yang, and S. Xie, "Computing resourcike trading for edge-cloud-assisted internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3661–3669, 2019.

[25] S. Guo, Y. Dai, S. Guo, X. Qiu, and F. Qi, "Blockchain meets edge computing: stackelberg game and double auction based task offloading for mobile blockchain," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5549–5561, 2020.

[26] F. Li, H. Yao, J. Du, C. Jiang, and Y. Qian, "Stackelberg game-based computation offloading in social and cognitive industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5444–5455, 2019.

[27] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: a joint optimization approach combining Stackelberg game and matching," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1204–1215, 2017.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5732300, 12 pages
https://doi.org/10.1155/2021/5732300

Research Article

Design of the Wireless Network Hierarchy System of Intelligent City Industrial Data Management Based on SDN Network Architecture

Wenken Tan[1] **and Jianmin Hu**[2]

[1]*College of Architecture and Urban Planning, Tongji University, Shanghai 200092, China*
[2]*Shanghai Tongqian Construction Planning and Design Co. Ltd., Shanghai 200433, China*

Correspondence should be addressed to Jianmin Hu; 49405890@qq.com

Received 13 August 2021; Accepted 4 October 2021; Published 10 November 2021

Academic Editor: Xuyun Zhang

With the rapid development of the industrial Internet of Things and the comprehensive popularization of mobile intelligent devices, the construction of smart city and economic development of wireless network demand are increasingly high. SDN has the advantages of control separation, programmable interface, and centralized control logic. Therefore, integrating this technical concept into the smart city data management WLAN network not only can effectively solve the problems existing in the previous wireless network operation but also provide more functions according to different user needs. In this case, the traditional WLAN network is of low cost and is simple to operate, but it cannot guarantee network compatibility and performance. From a practical perspective, further network compatibility and security are a key part of industrial IoT applications. This paper designs the network architecture of smart city industrial IoT based on SDN, summarizes the access control requirements and research status of industrial IoT, and puts forward the access control requirements and objectives of industrial IoT based on SDN. The characteristics of the industrial Internet of Things are regularly associated with data resources. In the framework of SDN industrial Internet of Things, gateway protocol is simplified and topology discovery algorithm is designed. The access control policy is configured on the gateway. The access control rule can be dynamically adjusted in real time. An SDN-based intelligent city industrial Internet of Things access control function test platform was built, and the system was simulated. The proposed method is compared with other methods in terms of extension protocol and channel allocation algorithm. Experimental results verify the feasibility of the proposed scheme. Finally, on the basis of performance analysis, the practical significance of the design of a smart city wireless network hierarchical data management system based on SDN industrial Internet of Things architecture is expounded.

1. Introduction

According to the analysis of "Made in China 2025" proposed under the background of the new era, the development of the industrial Internet of Things has received the attention of the whole society, especially the information and communication technology and intelligent manufacturing have now become the core content of China's urban construction and development. From the perspective of the Internet of Things practice, the Internet of Things can complete big data integration services more efficiently through the use of work. Among them, the Industrial Internet of Things, as the basis of Internet content, will integrate cloud computing, big data, and sensors into the entire process. The emergence of the Industrial Internet of Things uses advanced technology to gradually transform traditional industries into intelligent industries, thereby improving product quality and production efficiency. At the same time, during the development of the Industry 4.0 era, the biggest feature is the network physical system, also known as CPS. The whole physical process needs to be controlled by a computer, and relevant data and commands are transmitted to the CPS controller. With the comprehensive promotion of subsequent Ethernet technology, it can bring clear real-time

attributes and can transmit real-time or non-real-time traffic in the same media, but because of their incompatibility, it is impossible to run different technologies in the same physical media in the development of practice. In view of this situation, researchers have strengthened the research on time-sensitive network (TSN), especially as the key content of which the smart city data management algorithm has become the focus of scientific research and exploration, which will play a positive role in the future industrial communication and automation technology innovation. Abosata et al. [1] further verified the importance of TSN research in Industry 4.0 in their research on the development of China's industrial Internet of Things. Wiers et al. [2] also put forward new conclusions about time-sensitive network technology and its application in an industrial network, especially it has been widely used in many fields of industrial network and attracted more scholars to participate in the research. Nowadays, with the deepening of practical research tasks, effective countermeasures have been obtained to solve some TSN data processing problems [3]. For example, Cerrato et al. [4] put forward new views on the South Industrial Control Security Gateway on the basis of ensuring the security of sensitive data in the industrial control system. Pinheiro et al. [5] proposed the TT traffic data processing method based on the operation characteristics of TSN and completed various TT communication data processing work accordingly. At the same time, Xu et al. [6] proposed the tabu search heuristic algorithm to clarify the TT frame data processing so as to minimize the WCD of the RC frame. In addition, Smirnov et al. [7] calculated and analyzed the constraint conditions required for offline data processing according to the clear characteristics of time-sensitive traffic in TSN and the general configuration of behavioral function data so as to avoid the delay phenomenon of key communication flow from port to port. Chen et al. [8]. Mapped the actual system data processing problems to the job-shop data processing problems that did not need to wait under the condition of ensuring time-sensitive traffic to minimize network delay and then used the tabu search algorithm to deal with the problems. As the core content of urban economic construction and development, Internet of Things technology is gradually integrated into the innovation work of various industries and fields but also shows a very broad development prospect and brings more terminal equipment management problems. Combined with the above design and analysis of the smart city data management wireless networking hierarchical system based on SDN network architecture, it can be seen that the future technology research and development of the Internet of Things must pay attention to wireless network access control, and combined with CPS, the TSN data processing algorithm is deeply explored [9].

2. Methods

Similar to the industrial Internet of Things, it can be applied to the development of various industries and can provide more opportunities and challenges for the realization of automation and intelligent industrialization. Considering the integration of SDN network architecture and wireless network sharing system design in smart city construction, it can be seen that when the virtual WLAN design simplifies network management operation, it does not put forward effective solutions for resource allocation, so it is difficult to improve the overall performance in the practical development. Due to the open network architecture and industrial Internet of Things sharing data information in big data era, network information security issues are easy to affect information operations, so real-time networks have advantages in R&D application. First is large data transmission capacity. The second is to give priority to functional design. At this time, this problem can be solved by controlling the real-time communication network. This provides the basic guarantee for the actual data transmission [10] as shown in Figure 1.

2.1. Architecture Analysis. According to the analysis of Figure 2, the overall design framework is mainly divided into three layers: the first layer is the application layer; the second is the network control layer; and the third is the network infrastructure layer. Take the network infrastructure layer as an example, all the wireless access points (APs) contained in it need to build virtualized APs, also known as VAPs, which belong to virtual machines of physical APs and can be used to virtualize network resources. At the same time, all APs have the ability to virtual multiple VAPs, and all VAPs exist independently not only to provide system users with required network services but also to provide relay services for various APs. In addition, any switch with a wire terminal in the network system can run the OpenFlow protocol service [11].

2.2. Functional Design. According to the analysis of Figure 3, the structure of SD-AP contains three layers: (1) VAP: this level refers to the abstract materialization of SD-AP, which is mainly used to deal with channel detection, user management, state detection and other work. (2) Wireless network card driver: this level is mainly used to support the transmission of WLAN protocol signals and collect the information of the state of the wireless channel. (3) The south interface agent: it is mainly used to expand the protocol interpretation, to execute the commands proposed by the controller, and on the basis of encapsulating the network events, the relevant contents are transmitted to the controller through the southward protocol.

(1) *User Management*. System users should connect with the network through the SD-AP wireless port and conduct comprehensive control over the operation of the wireless network. This effort is user-centric, not port-centric. In this study, in order to better control all associated users, a lightweight virtual access point (LVAP), which represents the user's proprietary state set, is proposed to represent the specific state of the user's connection with the network. In essence, LVAP can be regarded as the interface provided by SD-AP for all users, mainly used

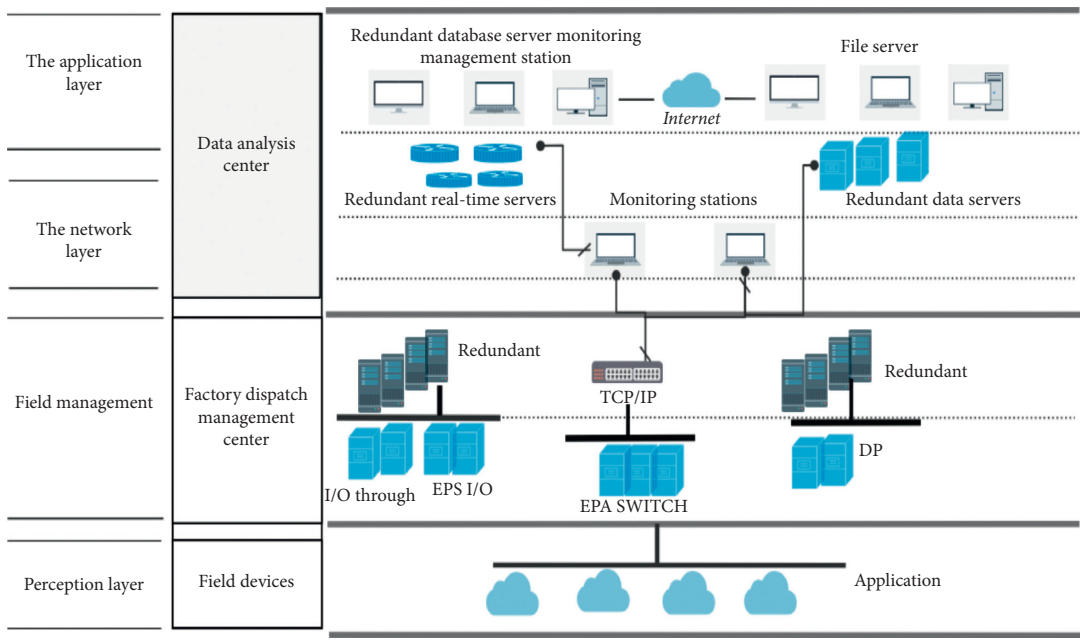


Figure 1: Industrial Internet of Things architecture.

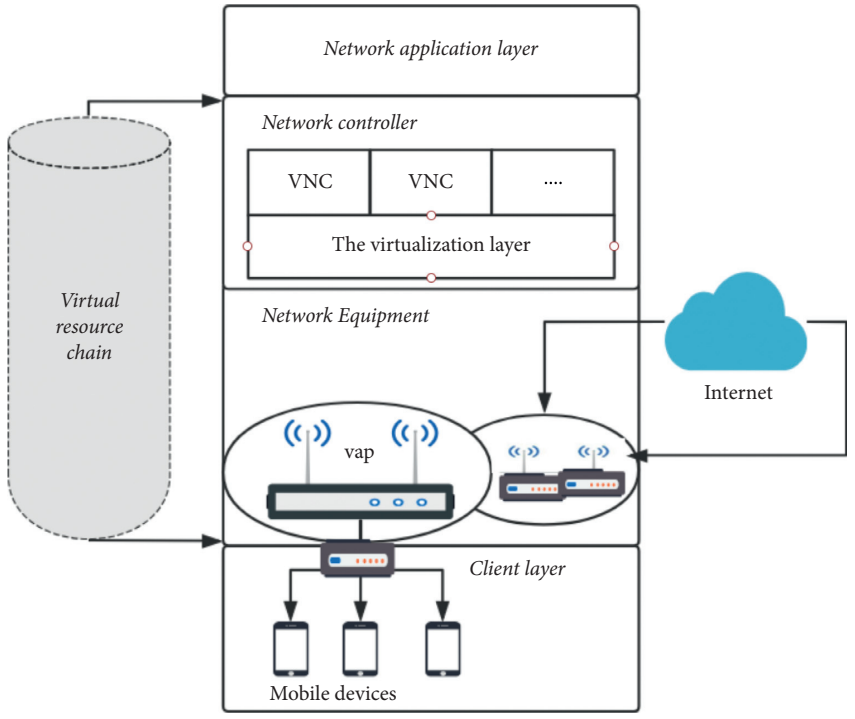


Figure 2: Network architecture diagram based on SDN.

to control the specific situation of users. When all users connect to the network for the first time, this module will form the LVAP, and all LVAPs will have a unique basic service set identifier (BSSID), which needs to be formed from the MAC address of the system user. At the same time, because LVAP reserves the authentication information of all users, such as MAC address, IP address, and so on, in the process of communication with SD-AP, it is equivalent to communicating with LVAP. In other words, as long as LVAPs are formed in SD-AP, users can be effectively linked to the network. In this module, users not only can manage the scheduling problems transformed into LVAP after abstract processing but also can put forward appropriate security countermeasures according to different user needs. [12].

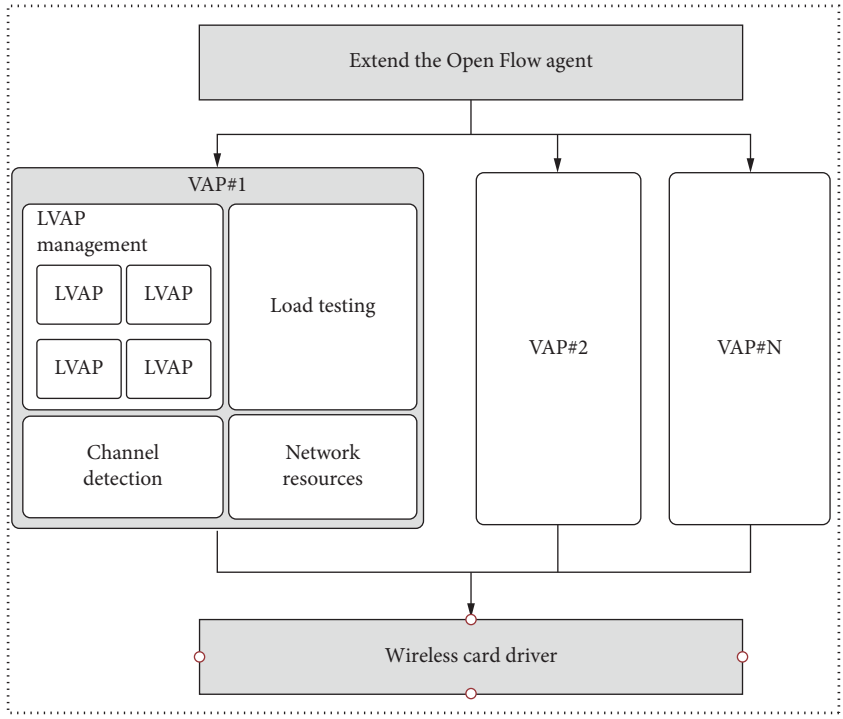


Figure 3: The SD-AP structural analysis diagram.

(2) *Channel Detection*. For all users, the most important concern is how to obtain a QoE or how to provide a corresponding QoS. And this mapping to the physical layer is the transmission quality of wireless communication. From the practical point of view, the construction of the channel monitoring module is to better guarantee the operation management quality of the application layer, avoid the interference of various factors during the application period, and thus improve the effectiveness of network resource allocation and the experience of receiving services. Generally speaking, the channel monitoring module is mainly used to control layer transmission. There is a relationship between the user and the signal-to-interference ratio, the overall strength of the received signal, the duration, and the average packet loss rate associating all the parameters between the SD-AP. These contents are uploaded to the network controller after encapsulation and processing, which can facilitate the network management personnel to implement resource scheduling according to the specific operation state [13].

(3) *Resource Description*. The network is distributed equally among all users, and the smallest unit of resources is called a resource block. And the user will obtain the network resources according to the acquisition of resource blocks. In the wireless network system, because the access mechanism of random access has been fully promoted. So user access to resources is competitive. However, in a large-scale network, factors such as adjacent channels and transmitting power will affect user services, and each SD-AP will involve computing storage resources, real-time performance and other related issues, which need to be dealt with directly in SD-AP. In other words, on the basis of collecting and analyzing network, SD-AP resources can implement global optimization processing to the whole network. It can be seen that the purpose of this module design is to obtain the resources in SD-AP and bring the specific view of network resources to the control layer so as to help the system application carry out the resource scheduling work in an orderly manner [14].

(4) *Status Monitoring*. In order to better provide users with quality services, the network will put forward corresponding optimization mechanisms according to the system operation, such as interference management and load balancing, so as to ensure that users can obtain a better service experience inside the system. In the process of popularizing these optimization countermeasures, it is necessary to combine the current network state analysis results to conduct in-depth exploration. Therefore, it is necessary to obtain the state information of SD-AP in the control layer, so as to bring an effective basis for the actual application layer. In practice, this module is mainly used to obtain the information related to computing storage and network resource application contained in SD-AP, package and process it into state frame, and transmit the relevant information to the controller under the condition that the controller makes a request. Because all SD-APs can involve multiple VAPs, in order to ensure the stable operation of all VAPs, it is necessary to balance all SD-AP

resources occupied by all VAPs in the control layer and report the application of resources in real time.

2.3. Network Controller. According to the analysis of Figure 3 above, this module is mainly divided into two parts: (1) the network virtualization module and (2) the network controller set (VNCs). The former is mainly used to virtualize the physical networks, while the latter is mainly used to control the virtualization resources within the network. This paper studies thinking from the user perspective, mapping the transmit power included in SD-AP into the letter drying ratio (SINR) of all users, then the mapping between the resource and the user signal quality is obtained, and the following formula is obtained:

$$\mathrm{SINR}_{ij} = \frac{g_{ij}P_j}{\sum_{k\in Ai\cap k\neq j} g_{ik}P_k + N_0}, \tag{1}$$

where g_{ij} represents the link gain case existing by the SD-AP user i, P_j represents the emission power of the SD-AP, N_0 represents additive Gaussian white noise, and Ai represents the cover range, involving all SD-AP sets contained by the user i.

According to the network controller, the wireless network system design must include two forms: one is resource management (RVNC) and the other is state management (SVNC). For the former, its work content is to classify the internal resources according to the resource view described by the network virtual module so as to facilitate the subsequent resource scheduling. The latter needs to collect various states of the network, such as resource occupation and network operation, so as to lay a foundation guarantee for the orderly allocation of subsequent resources [15].

The biggest advantage of the southbound interface expansion protocol of OpenFlow is centralized processing control and convenient fine-grained flow-based flow control. Therefore, the number of associated user terminals for all VAPs can be accurately calculated in a specific topological view. The average channel utilization probability formula for VAP_j is as follows:

$$CU_j = \frac{\sum_{i=1}^{N} t_i}{T} = \frac{\sum_{i=1}^{N} l_i/r_{ij}}{T}, \tag{2}$$

Formula (2) expresses the transmission speed of real-time network. In the formula, T represents the unit test time, t_i represents the user I channel time, Ii represents the user I's unit test time range, and r_{ij} represents the user I's VAPj probability of data transmission speed. Unit test includes general data collection and transmission times; the specific calculation formula is as follows:

$$r_{ij} = f(G, P), \tag{3}$$

where P represents the transmitted power, G represents the factor of link gain, and F represents the mapping function

SDN integrates existing virtual WLAN and time-sensitive networks (TSN) to obtain the TSSDN architecture, as shown in Figure 4. It can isolate the time-triggered flow from the spatial and temporal perspectives and thus provide a basic guarantee for data clarity.

Combined with the analysis of the OpenFlow architecture diagram shown in Figure 5, it is assumed that the source of the time-triggered flow will deliver unicast packets to the specified area at a fixed bit rate, and the time period is set to be an integer multiple of the minimum transmission period that can be supported. This time-triggered mode is more suitable for sensors with fixed sampling cycles or actuators that transmit commands at specified time intervals so that time-triggered traffic can transmit data in high-priority UDP packets and is stronger than other priority traffic. Because the end system needs to synchronize accurately using the precise time protocol (PTP) and all event-triggered traffic has the same priority, an additional scheduling mechanism needs to be proposed to optimize the time-triggered traffic when conflicts occur.

According to the Figure 5 architecture, derived by the above formula, the trigger time can be determined. $G \equiv (V, E)$.

Consider the trigger flow of time as a tuple $ts_i \equiv (s_i, d_i)$. And, V represents a collection of nodes, and derive $E \equiv \{(i, j) \mid i, j \in V\}$, where i and j represent network link connections as a set of network connections. At the same time, $V \equiv (S \cup H)$, where S and H are collections of switches and hosts $s_i, d_i \in H$, where s_i and d_i represent the source and final destination region of the stream, respectively. Then you design the input variables for the related problem, where the set of streams triggered by scheduled events is as follows:

$$\mathrm{TS} : \mathrm{TS} \equiv \{ts_i\}. \tag{4}$$

The specific map to the network link is as follows:

$$\mathrm{SL} : \mathrm{SL} \equiv \left\{f_{i,j}\right\}, \forall i \in \mathrm{TS}, \forall j \in E. \tag{5}$$

Suppose the stream i is transferred to the destination area via the link j, otherwise, it will become 0.

The specific map of the flow to the time slot is as follows:

$$\mathrm{ST} : \mathrm{ST} \equiv \left\{t_{i,k}\right\}, \quad \forall i \in \mathrm{TS}, \forall k \in T. \tag{6}$$

The corresponding variables are as follows:

$$\mathrm{SLT} : \mathrm{SLT} \equiv \left\{y_{i,j,k}\right\}, \quad \forall i \in \mathrm{TS}, \forall j \in E, \forall k \in T. \tag{7}$$

The stream i is obtained assuming that the link j is transferred to the destination area and is assigned to the time slot k, $y_{i,j,k} = 1$. Otherwise, it will become 0. This gives the best scheduling countermeasures to define the delivery schedule of the event trigger flow, as follows:

$$\begin{aligned} &\text{subjectto: } u_n = \text{minimize} \sum_{\forall i\in TS} \sum_{\forall j\in E} f_{i,j}, \\ &\sum_{\forall k\in T} t_{i,k} = 1 \forall i \in TS. \end{aligned} \tag{8}$$

Among them, the shortest path formula for the optimization objective is as follows:

$$\sum_{\forall j\in \mathrm{in}(\mathrm{src}(i))} f_{i,j} = 0 \quad \sum_{\forall j\in \mathrm{out}(\mathrm{src}(i))} f_{i,j} = 1. \tag{9}$$

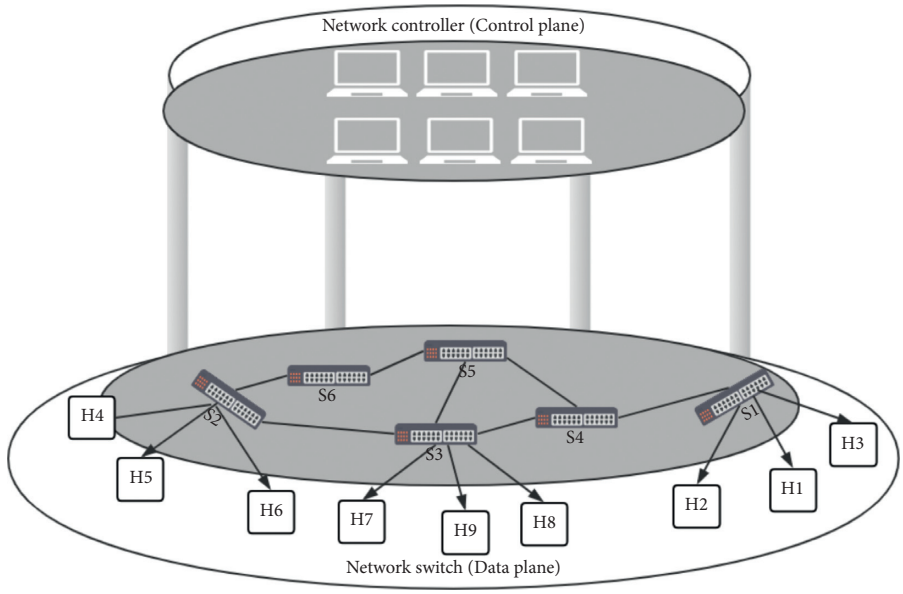


Figure 4: Network architecture diagram of TSSDN.

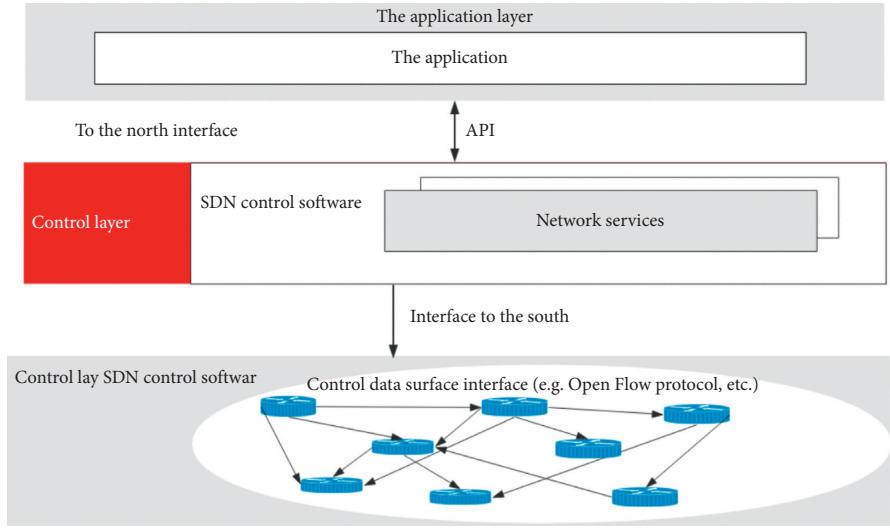


Figure 5: Network architecture diagram of OpenFlow.

The calculation formula for the time-slot constraints is as follows:

$$\sum_{\forall j \in m(\mathrm{dst}(i))} f_{i,j} = 1 \quad \sum_{\forall j \in \mathrm{out}(\mathrm{dst}(i))} f_{i,j} = 0. \tag{10}$$

The numerical formula for variable consistency is as follows:

$$y_{i,j,k} = f_{i,j} \times t_{i,k} \forall i \in TS, \quad \forall j \in E, \forall k \in T, \tag{11}$$

where in (src (i)) represents the input of the source host, out (src (i)) represents the source host output, in (dst (i)) represents the input of the destination host, and out (dst (i)) represents the output of the destination host.

2.4. Transmission Scheduling. In studying the scheduling of time-triggered flows in the industrial Internet of Things, the transmission delay of link packets can be seen as a binary multibackpack phenomenon and is studied using the minimum ant system (MMAS) algorithm. Assuming you want n items into different weights of m backpacks and all the items in the backpack have different qualities, then you need to consider the backpack in which they should be placed. The multiple pack problem is more complex than the 0–1 pack problem and can be described as a backpack with m capacity is $c_1, c_2, \ldots, c_m$. And there are n items whose value is P_i; in the case of putting the i-th item into the j-th item, the corresponding weight is W_{ij} and conforms to $1 \le i \le n$ and $1 \le j \le m$. This problem involves calculating two cases: one

where a single item goes into the backpack and the other is not to put a certain item into the backpack at all; in the backpack, capacity does not meet the constraints on the basis of maximizing the total value of the items into the backpack.

The problem of binary multiple knapsacks can be expressed as follows:

$$\text{maximizeP}(\overline{x}) = \sum_{i=1}^{n} x_i p_i, i \in \{1, ..., n\}. \tag{12}$$

The formula for maximizing the total value of items in the backpack of the optimization target is as follows:

$$\sum_{i=1}^{n} w_{ij} x_i \le c_j j \in \{1, \ldots, m\}. \tag{13}$$

The capacity limit formula of each backpack is as follows:

$$x_i = 1\text{or}0. \tag{14}$$

Among them, $x_i = 1$. Item I is selected into the backpack, and 0 is not selected into the backpack. The formula for selecting a game for a bunch of items is as follows:

$$\overline{x} = (x_1, x_2, \ldots, x_n). \tag{15}$$

This paper studies the use of a directed graph $G(V, E)$ to represent the network, where V represents the set of nodes and conforms to $V \equiv (S \cup H)$. This condition. The KTH bandwidth provided by link (I, j) is assumed to be consistent $x_{ij}^k = 1\text{or}0$. A stream can go through a link, so it is 1, or vice versa. In addition, combining with the conservation constraint analysis of flow, we can get

$$\begin{cases} \sum_{j:(i,j)\in E} x_{ij}^k - \sum_{j:(j,i)\in E} x_{ij}^k = 0, & k \in TS,\ i \ne s_i, d_i, \\ \sum_{j:(i,j)\in E} x_{ij}^k - \sum_{j:(j,i)\in E} x_{ij}^k = 1, & k \in TS,\ i = s_i, \\ \sum_{j:(i,j)\in E} x_{ij}^k - \sum_{j:(j,i)\in E} x_{ij}^k = -1, & k \in TS,\ i = d_i. \end{cases} \tag{16}$$

It is proved that the traffic of the KTH stream of the network entering the transmission node should be consistent with the traffic proposed by this node.

At the same time, the link capacity should be restricted and meet the constraint conditions, as follows:

$$\sum_{k\in TS} d_k x_{ij}^k \le c_{ij}, (i, j) \in E. \tag{17}$$

In addition, the formula of service self-similar traffic is used to study the average packet delay of the flow through the i-th link. The calculation formula is as follows:

$$\tau_{ij} = \frac{\rho_{ij}^{1/2(1-H)}}{d_k x_{ij}^k \left(1 - \rho_{ij}\right)^{1/(1-H)}}, \tag{18}$$

where ρ_{ij} conforms to the $\rho_{ij} = d_k x_{ij}^k / c_{ij}$. This condition, and H is a constant.

According to the binary backpack problem, the number of the link through which the event-triggered flow passes is determined so as to avoid queuing phenomena during the transmission of packets and to minimize the actual delay phenomenon so as to achieve optimization processing goal. The specific mathematical model is as follows:

$$\begin{aligned} &\text{Subjectto: } u(n) = \text{Mininize} \sum_{(i,j)\in E} x_{ij}^k \tau_{ij}, k \in \text{TS}, \\ &\quad x_{ij}^k = 1\text{or}0, \\ &\quad \begin{cases} \sum_{j(i,j)\in E} x_{ij}^k - \sum_{j(j,i)\in E} x_{ji}^k = 0, & k \in \text{TS},\ i \ne s_i, d_i, \\ \sum_{j(i,j)\in E} x_{ij}^k - \sum_{j(j,i)\in E} x_{ji}^k = 1, & k \in \text{TS},\ i = s_i, \\ \sum_{j(i,j)\in E} x_{ij}^k - \sum_{j(j,i)\in E} x_{ji}^k = -1, & k \in \text{TS},\ i = d_i, \end{cases} \\ &\quad \sum_{k\in \text{TS}} d_k x_{ij}^k \le c_{ij}, (i, j) \in E. \end{aligned} \tag{19}$$

From the perspective of ant colony optimization algorithm, it is assumed that the number of nodes is n; the total number of ants is m; and the distance between node I and node j needs to be expressed as $d_{ij}(i, j = 1, 2, ..., n)$. Then, the residual information intensity at time t on the link between the two can be expressed as $\tau_{ij}(t)$. Ant k looks for the next path according to the residual pheromone intensity; then the probability that it moves from node I at time t to node j is $p_{ij}^k(t)$. Then, we can get

$$p_{ij}^k(t) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}(t)\right]^{\beta}}{\sum_{s\subset J_i(i)} \left[\tau_{is}(t)\right]^{\alpha}\left[\eta_{is}(t)\right]^{\beta}}, & j \in J_k(i), \\ 0, & \text{other} \end{cases} \tag{20}$$

where $J_k(i)(i = 1, 2, ..., n) - \text{tab}u_k$ represents the node set that ant k can walk, and the taboos can be expressed as $\text{tab}u_k$; Nodes that pass through are recorded in the taboos table. After the ant passes through all nodes and returns to the initial point, n cities are recorded $\text{tab}u_k$. And the path is the feasible solution. $\eta_{ij}(t)$ represents the degree of expectation of ant K from node I to I and represents the heuristic factor, usually $1/d_{ij}$. The corresponding importance needs to be utilized α, β. s represents the neighboring nodes of the current time t. After the completion of all ants in each link, the global update formula is as follows:

$$\begin{aligned} \tau_{ij}(t + 1) &= (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}, \\ \Delta\tau_{ij} &= \sum_{k=1}^{m} \Delta_{\tau_{ij}}^k, \end{aligned} \tag{21}$$

where $\rho(0 < \rho < 1)$ represents the volatility coefficient, $1 - \rho$ represents the permanent coefficient of pheromone, using the information increment in the ant week model $\Delta\tau_{ij}$. After the above derivation, as shown in the figure, we show the basic model of the Ant (ACO) algorithm:

$$\Delta\tau_{ij} = \begin{cases} \dfrac{Q}{L_k}, & \text{ant } k \text{ passes a link between } I \text{ and } j, \\ 0, & \text{other}, \end{cases} \tag{22}$$

where Q represents a constant, which refers to the intensity of pheromone, and L_k represents the length of the path traveled by the KTH ant.

Through optimization and improvement analysis of TSN scheduling algorithm combined with ant colony system, the improved form of the initial pheromone is shown as follows:

$$\tau_0 = \frac{1}{A + \text{count}\left(U_{J_i(i)}\right)}, \tag{23}$$

where A represents the parameter, U represents the complement of the node set, and count $(U_{J_i(i)})$ represents the number of currently impassable nodes in the vicinity of the node.

In order to improve the level of actual random search, the adaptive pseudo-random ratio should be used to obtain the next node j. The specific formula is as follows:

$$j = \begin{cases} \arg\max\left\{\tau_{ij}\left[\eta_{ij}\right]^{\beta}\right\}, & q \le q_0, \\ s, & q > q_0, \end{cases} \tag{24}$$

where q_0 will usually get a constant in the range of 0 to 1 and q can also be selected at random. The probability of getting the next node needs to be determined by q. Assuming that the number goes down gradually, the corresponding probability goes up. When the amount of information and heuristic factors between nodes are strengthened, the results are as follows:

$$\tau_{ij}(t+1) = (1-\rho)^{\sqrt{m+c/m}}\tau_{ij}(t) + \Delta\tau_{ij}^{bs}, \tag{25}$$

in $\tau < \tau_{\max}$. At this time, it is derived as

$$\tau_{ij}(t+1) = (1-\rho)^{\sqrt{m-c/m}}\tau_{ij}(t) + \Delta\tau_{ij}^{bs}, \tag{26}$$

where $\rho \in (0,1)$ represents the volatility coefficient of pheromone, while t represents the specific parameter. The corresponding algorithm flow chart is shown in Algorithm 1.

3. Results

3.1. Performance Analysis of Transmission Scheduling. First, the application performance of the proposed improved algorithm shows that the impact of the number of event-triggered flows in the industrial IoT should be studied. The improved algorithm MMAS uses the network topology of the switch and the host to detect the working hours of industrial data processing at different times; the algorithm is compared with 10 traditional simulated annealing algorithms (TSA) and 11 traditional genetic algorithms (TGA). The final results are shown in Figures 6 and 7.

According to the analysis of the picture presentation results, the working time needed to outline the algorithm is less than the other two algorithms [16].

The actual number is controlled between 13 and 86, and the 60 event trigger streams are used for smart city transmission data processing, and finally, the working hours are carefully detected under different network topologies. Based on the linear optimization of topology size, the improved algorithm proposed in this paper is significantly lower than the results of other algorithms [17].

This paper designs two standard forms, one with 4 switches and 12 hosts, 6 switches with 24 hosts, and 60 event trigger streams to process and transmit data at any time. Compared with the above three algorithms, 30 operations are performed, and the results are shown in Table 1. At this time, it is shown to improve the convergence of the algorithm under different network scales. This shows that the target value and average value of the improved algorithm are better than the other two algorithms, and the number of optimal solutions in the 30-run process is more than that of the other two algorithms [18].

Combined with the performance comparison results shown in Table 1, we study the target convergence results of three algorithms at two sizes, as shown in Figure 8 [18].

Compared with the convergence at two scales, the improved algorithm is superior and the optimal solution can be obtained quickly. It proves that the MMAS algorithm can show positive advantages in the intelligent urban data management based on SDN network architecture, can effectively deal with the TSN data processing problems, appear no falling problems similar to the traditional algorithm, and thus improve the computational efficiency and quality of the actual optimal solution. [19].

3.2. Controller Performance Analysis. Considering the jitter change of the network time continuation, the observation and analysis show that when the ants look for the initial route, the number of pheromones in the link is very small, so the initial jitter change is very large. On the basis of increasing iterations, the jitter level of the design algorithm began to reach stability. On the basis of increasing the number of iterations, it can be lower than 22% from the perspective of parameters, so the jitter level of the design algorithm begins to stabilize. In addition, when studying the convergence of the target value obtained by triggering the flow algorithm at different times, 6 switches and 24 hosts are selected to build a network scenario, and the control quantity value is between 12 and 68. It is concluded that the actual number of iterations of the simulation results reaches 50 times, and the final results are shown in Table 2.

It is proved that the algorithm selected in this paper has better performance. In order to prove that the algorithm selected in this paper has better performance, the influence on the simulation is studied in combination with the number of network connections. Based on the continued increase in the number of network links, the objective function value will be increased, and the results show that the increase in the improved algorithm is significantly lower than the original algorithm. Therefore, the improved algorithm can meet the needs of system design. Through observation and analysis, it can be seen that the working time of the two

Input: $G\ (V, E)$
Initialize: τ_0, $[\tau_{ij\min}, \tau_{ij\max}]$, α, β, ρ, MacGen, MaxN
Output: $u(n)$
Procedure SET_TABU_INFORMATION
For $\forall k \in M$
Build $J_k(i)(i = 1, 2, \ldots, n) - \text{tab}u_k$
End for
End procedure
Procedure CONSTRUCT_ROUTES
For $i, j \in V$
For $k \in M$ do
Select the next node according to the following formula

$$j = \begin{cases} \arg\max\left\{\tau_{ij}[\eta_{ij}]^{\beta}\right\}, & q \le q_n \\ s, & q > q_n \end{cases}$$

End for
End for
For $\forall k \in M$ do
Continue to pathfinding
End for
End procedure
Procedure UPDATE_PHEROMONES
To calculate L_{hen}

$$\text{update } \tau_{ij}(t+1) = \begin{cases} (1-\rho)^{\sqrt{m+c/m}}\tau_{ij}(t) + \Delta\tau_{ij}^{bc}, \tau > \tau_{\max} \\ (1-\rho)^{\sqrt{m-c/m}}\tau_{ij}(t) + \Delta\tau_{ij}^{bc}, \tau < \tau_{\max} \end{cases}$$

End procedure
Procedure MAIN
For $\forall (i, j) \in E$ do
$t_{ij} \longleftarrow \tau_0$
$\eta_{ij} \longleftarrow 1/d_{ij}$
End for
While the termination condition is not met do
SET_TABU_INFORMATION
CONSTRUCT_ROUTES
UPDATE_PHEROMONES
End while
End procedure

Algorithm 1: MMAS algorithm.

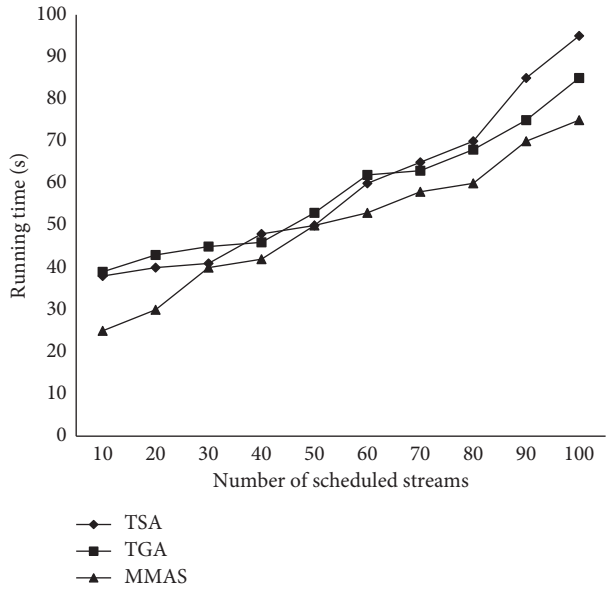


Figure 6: Research algorithm running time based on the number of trigger flows.

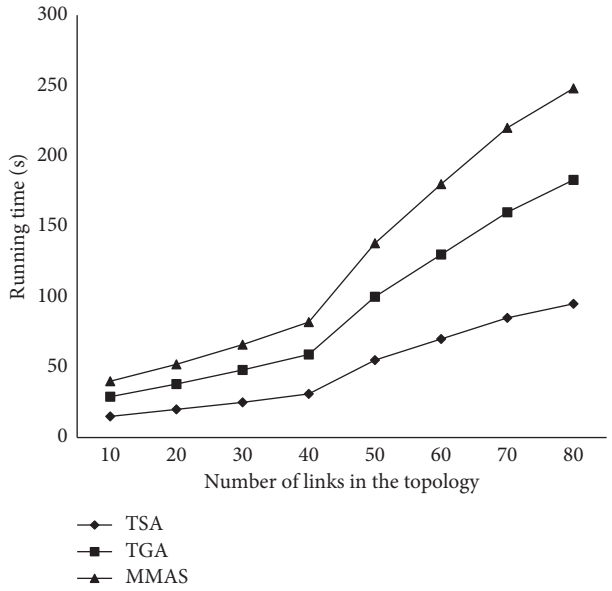


Figure 7: Run time according to the network topology research algorithm.

Table 1: Performance comparison results of the two network sizes.

Network size	Algorithm	Number of optimal solutions	Optimal solution	Average
Size 1	TSA	12	33	35.6
Size 1	TGA	13	32	34.2
Size 2	MMAS	19	29	29
Size 1	TSA	15	136	151.3
Size 2	TGA	11	145	160.7
Size 2	MMAS	18	126	135.2

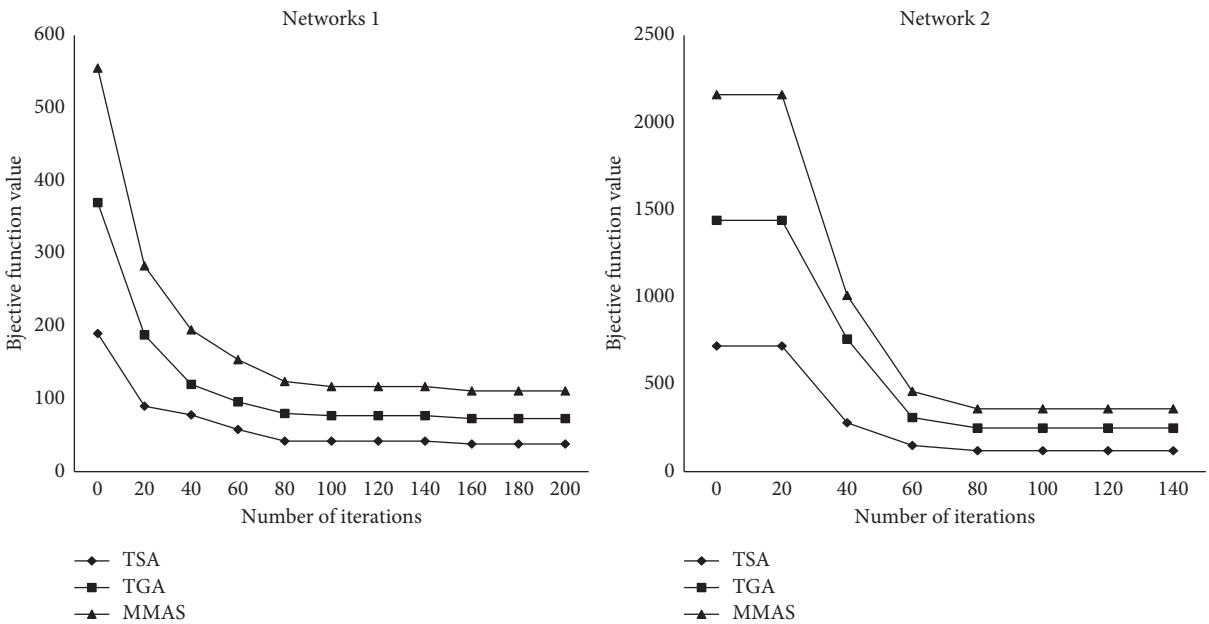


Figure 8: The algorithm target convergence situation.

Table 2: Comparison results of the algorithm performance.

Purpose of the flow	Optimal solution obtained	Average solution	Average number of iterations
12	126.2	133.4	21.23
	118.5	124.6	16.45
26	132.1	136.1	22.21
	121.3	125.7	16.36
40	136.4	140.2	18.77
	126.5	128.6	15.12
68	137.1	139.8	16.15
	125.9	128.2	14.2

algorithms will increase along with the continuous linear increase of the network topology. For smaller network structures, the incremental ant colony algorithm takes a longer time than the original one. However, from the perspective of large-scale network structure, the incremental ant colony algorithm has a shorter running time. Therefore, the incremental ant colony algorithm can effectively deal with large-scale network scheduling problems in the SDN network architecture.

4. Conclusion

To sum up, with the in-depth research on industrial Internet of Things technology concept in recent years, domestic and foreign researchers gradually realize the positive role of ant colony algorithm in the overall technical research and start to use the soft-considered network (SDN) framework to provide basic guarantee for data processing transmitted by a smart city. In this study, the scheduling problems in smart city data management design are static scheduling. It changes the possibility that some unpredictable disturbances may interfere with our schedule due to the complexity of the intelligent industry environment and uses the minimum ant system (MMAS) algorithm to design the system framework. For better improvement, dynamic scheduling is selected instead. At the same time, the data processing problem is discussed deeply. In this way, we not only can find effective optimization countermeasures but also further improve the urban intelligent construction system to meet the needs of urban residents and industrial development [20, 21].

Data Availability

This article did not collect relevant data but designed the algorithm and added parameters for simulation. The data have been reflected in the table.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the present study.

References

[1] N. Abosata, S. Al-Rubaye, G. Inalhan, and C. Emmanouilidis, "Internet of things for system integrity: a comprehensive survey on security, attacks and countermeasures for industrial applications," *Sensors*, vol. 21, no. 11, p. 3654, 2021.

[2] R. W. Wiers, S. L. Ames, W. Hofmann, M. Krank, and A. W. Stacy, "Impulsivity, impulsive and reflective processes and the development of alcohol use and misuse in adolescents and young adults," *Frontiers in Psychology*, vol. 1, no. 6, p. 144, 2010.

[3] X. Xu, D. Zhu, X. Yang, S. Wang, L. Qi, and W. Dou, "Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–17, 2021.

[4] I. Cerrato, F. Risso, R. Bonafiglia, K. Pentikousis, G. Pongrácz, and H. Woesner, "COMPOSER: a compact open-source service platform," *Computer Networks*, vol. 139, pp. 151–174, 2018.

[5] A. J. Pinheiro, E. B. Gondim, and D. R. Campelo, "An efficient architecture for dynamic middlebox policy enforcement in SDN networks," *Computer Networks*, vol. 122, pp. 153–162, 2017.

[6] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[7] F. Smirnov, M. Glas, and F. Reimann, "Formal timing analysis of non-scheduled traffic in automotive scheduled TSN networks," in *Proceedings of the Conference on Design. European Design and Automation Association*, pp. 1643–1646, Lausanne, Switzerland, September 2017.

[8] M.-H. Chen, Y.-C. Tien, Y.-T. Huang, I.-H. Chung, and C.-F. Chou, "A low-latency two-tier measurement and control platform for commodity SDN," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 98–104, 2016.

[9] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for IoV in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[10] R. De Donno, A. Ghidoni, G. Noventa, and S. Rebay, "Shape optimization of the ERCOFTAC centrifugal pump impeller using open-source software," *Optimization and Engineering*, vol. 20, no. 3, pp. 929–953, 2019.

[11] X. Luo, J. Liu, D. Zhang, and X. Chang, "A large-scale web QoS prediction scheme for the industrial internet of things based on a kernel machine learning algorithm," *Computer Networks*, vol. 101, pp. 81–89, 2016.

[12] B. Rahmani, "Industrial internet of things: design and stabilization of nonlinear automation systems," *Journal of Intelligent and Robotic Systems*, vol. 86, no. 3-4, 2017.

[13] T. Miyashita, T. Suzuki, T. Soumiya, and A. Yamada, "SDN solution for wide area networks," *Fujitsu Scientific & Technical Journal*, vol. 52, no. 2, pp. 28–34, 2016.

[14] Y. Han, T. Vachuska, A. Al-Shabibi, and J. Li, "ONVisor: towards a scalable and flexible SDN-based network

virtualization platform on ONOS," *International Journal of Network Management*, vol. 28, no. 2, pp. 1–20, 2018.

[15] J. Lian and M. Gao, "Design and Implementation of network information service platform based on microservice architecture," *Journal of Physics: Conference Series*, vol. 1678, p. 012094, 2020.

[16] F. Dongqing and X. Zhu, "Design and implementation of 6LOWPAN smart city data acquisition system," *Computer Engineering*, vol. 11, pp. 286–291, 2017.

[17] Y. Zhang, P. Tan, H. Ma, and M. Don, "Improving the seismic performance of staircases in building structures with a novel isolator," *Computer Modeling in Engineering and Sciences*, vol. 124, no. 2, pp. 415–431, 2020.

[18] L. Zhang, L. He, and J. Huang, "Research on secure device routing based on SDN network," *Computer Engineering and Applications*, vol. 54, no. 4, pp. 103–109, 2018.

[19] P. Gao, F. Zhang, and D. Zhang, "High deterministic traffic control method for cloud architecture networks based on SDN," *Computer Engineering*, vol. 44, no. 12, pp. 80–84+90, 2018.

[20] Y. Jin, Y. Liu, and X. Wang, "Multi-path traffic scheduling algorithm for data center network based on SDN," *Computer Science*, vol. 6, pp. 90–99, 2019.

[21] F. Li, Z. Yu, and C. Qin, "Constructive texture steganography based on compression mapping of secret messages," *Computer Modeling in Engineering and Sciences*, vol. 124, no. 1, pp. 393–410, 2020.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 1737850, 11 pages
https://doi.org/10.1155/2021/1737850

Research Article

Coauthorship Network Mining for Scholar Communication and Collaboration Path Recommendation

Weiting Zhao,[1] **Zheng Zou**,[1] **Zidong Wei**,[1] **Wenwen Gong**,[2] **Chao Yan**,[1] **and Ashish Kr Luhach**[3]

[1]*School of Computer Science, Qufu Normal University, Jining, China*
[2]*College of Information and Electrical Engineering, China Agricultural University, Beijing, China*
[3]*Department of Electrical and Communications Engineering, The PNG University of Technology, Lae, Papua New Guinea*

Correspondence should be addressed to Ashish Kr Luhach; ashish.kumar@pnguot.ac.pg

Received 20 August 2021; Revised 9 October 2021; Accepted 16 October 2021; Published 5 November 2021

Academic Editor: Hao Wang

With the increasing penetration of interdisciplinary subjects, it is more difficult for researchers to complete a paper individually, showing that the division of labor can improve the level and efficiency of scientific research. Thus, collaboration among multiple scholars has become a trend in academic research. However, because the numbers of scholars and papers are increasing and cooperation between scholars has become more frequent in recent years, it is an increasingly challenging task to discover useful knowledge resources for researchers. Against the background of big data, how to help scholars quickly find interested target collaborators, encourage them to participate more actively in academic communication, and create high-quality achievements in scientific research has become a significant problem. Considering this challenge, this article proposes a framework of coauthorship strength, author contribution, and search (CCS , taking the first letter of the keyword), which is based on the coauthorship feature of Google Academics. In CSS, we combined the search algorithm to select the optimal connection path to help scholars find interested target scholars efficiently and to better solve practical application problems. Finally, our proposal is evaluated by a set of experiments based on a real-world dataset. Experimental results of our approach show better search outcomes compared to other competitive approaches.

1. Introduction

With the development of Internet information technology, academic communication and cooperation are no longer restricted by geographical location. Scholars from different research institutions and different countries can conveniently engage in academic communication. At the same time, with the increasing penetration of interdisciplinary subjects, it is more difficult for researchers to complete a paper individually, and the division of labor can improve the level and efficiency of scientific research. Thus, collaborative research between scholars has become the trend in academic investigation [1]. In academia, coauthors jointly publish papers, a practice that can be regarded as reliably representing scientific cooperation. Generally, papers coauthored by multiple institutions have a higher number of citations than papers published by one research institute [2]. The reason for this is that, through formal or informal personal interactions, researchers share knowledge, exchange ideas, and jointly ensure the accuracy of research results, providing a scientific advantage.

However, considering the example, suppose that scholar A has cooperated with many scholars, excluding scholar B. Scholar A inadvertently reads the scientific research achievements of scholar B and wants to exchange ideas and enter discussions with scholar B. At this time, intermediary scholars can make connections. However, the choice of intermediary scholars has also changed. Generally, the greater the coauthorship intensity is, the closer the communication is, and the easier it is to establish connections with other scholars. Therefore, establishing a coauthor network and selecting proper intermediary scholars to help other researchers connect with each other will be useful and meaningful. However, currently, research faces the two following challenges:

(1) The current scholarly contributions mainly use single indicators, for example, the number of coauthors, which lacks a comprehensive method to calculate coauthorship strength. This may lead to an assessment of the results of the collaboration that are not sufficiently accurate.

(2) Most of the existing studies did not combine related search algorithms for experimental verification, which leads to a lack of application in actual scenarios. This drawback is prone to utilize theoretical methods of calculating coauthor strength that may not be suitable for practical applications.

Considering the above challenges, we propose a coauthorship strength, author contribution, and search framework (CCS) based on the Google Academic Platform. This new approach not only considers multiple coauthorship indicators to comprehensively calculate coauthorship strength but also uses the Dijkstra search algorithm to apply it to the actual coauthorship scene. Therefore, it can use a more comprehensive coauthorship strength calculation method to choose more suitable intermediaries, which can establish connections between scholars and solve the existing research shortcomings.

In summary, our scientific contributions in this study are fourfold:

(1) We obtain the real dataset of the Google Academic Platform using crawler technology to build a coauthorship network.

(2) We take into consideration the number of coauthors, the number of times they have collaborated, the number of citations of the paper, and scholarly contributions of the same paper to ensure an accurate measurement of coauthorship.

(3) We combined a search algorithm to select the optimal connection path between scholars in the form of intermediaries to help scholars find interested target scholars efficiently and to be more effective at solving practical application problems.

(4) A wide range of experiments are enacted according to the real dataset from the Google Academic Platform. Compared with other solutions, the reported experimental results show that our solution has better performance.

The remainder of the paper is organized as follows. The recent literature is investigated in Section 2 to review the current research status in the field. In Section 3, we introduce the coauthorship calculation method and search framework in detail. Evaluations of the experiment are presented in Section 4. In Section 5, we summarize the article and indicate prospective future work.

2. Related Work

In 2001, Newman et al. [3] first studied coauthorship networks. They pointed out that the coauthorship strength between two authors is not constant because the number of authors is inversely proportional to the strength of coauthorship, so a method for calculating the edge weight of coauthorship networks based on the number of coauthors is proposed. However, this method does not take into account the influence of the papers; thus, Hrisch et al. [4] proposed the h index based on the frequency of citations and the number of documents. Scholars from the Chinese Academy of Sciences have performed a more in-depth study of scientific research cooperation [5]. They analyzed four existing weighting models and found that none distinguished the different coauthorship strength between authors based on signature order. Therefore, they suggested introducing factors such as interpersonal relationships, author discipline, and organization affiliation into the calculation of coauthorship strength. Han et al. [6] believed that the cooperation of two authors could be regarded as one author supporting the scientific work of another and proposed a method that uses author cooperation support analysis to calculate the strength of cooperation in a coauthorship network. Empirical research on real large-scale datasets shows that support measures are meaningful. As the data scale continues to grow, the ranking of academic entities is becoming an increasingly compelling task. Therefore, Amjad et al. [7] considered the number of papers, the number of citations, and whether the scholar was the first author; they also proposed a scholarly ranking algorithm based on mutual influence and citation exclusivity. Practice has shown that the proposed method has produced substantial results.

However, existing research on coauthorships only proposes a theoretical method to calculate coauthorship strength, which has not been applied to actual scenarios. According to the Google Academic Platform, scholars can effectively find interested scholars through the existing coauthorship intermediaries capacity, yet there is no complete and effective practical plan. Based on this situation, we propose a coauthorship strength, author contribution, and search framework (CCS) based on the Google Academic Platform. It not only considers multiple coauthorship indicators to comprehensively calculate coauthorship strength but also uses a search algorithm to solve the application problem in the actual coauthorship scene.

3. Motivation

Figure 1 gives a concrete example to illustrate the motivation of this article. Suppose that scholar A has cooperated with many scholars, with the exception of scholar B. Scholar A inadvertently reads the scientific research achievements of scholar B and wants to exchange ideas and engage with scholar B.

In this situation, scholar A is defined as the source scholar, scholar B is defined as the target scholar, and scholar A finds scholar B of interest. We find that there is not just one social relationship between these scholars, meaning that there are many possible contact schemes. For example, scholar A can regard scholar C as an intermediary and connect with scholar B indirectly. Another option for scholar A is through scholar D. Both schemes are

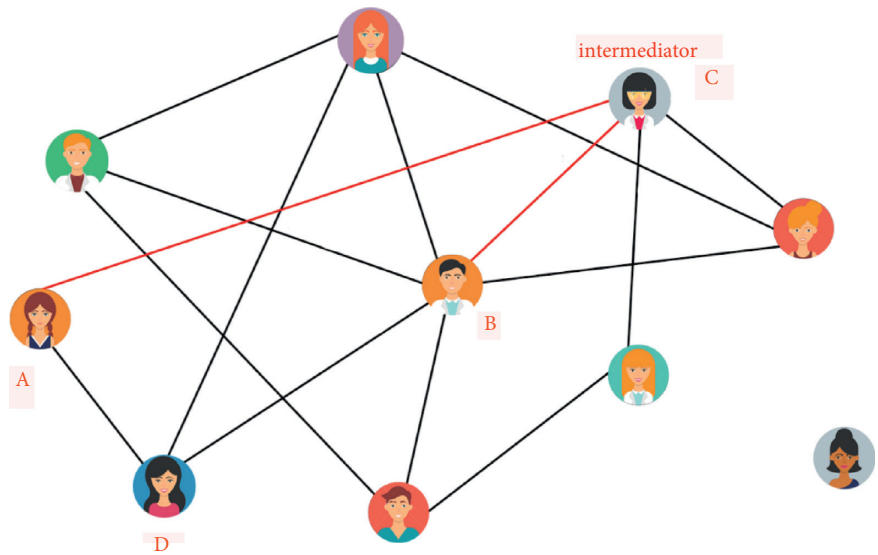


Figure 1: Coauthorship relationship based on the Google Academic Platform.

theoretically feasible, but because the coauthorship strength between scholars is often different, a popular understanding is that two people with a stronger relationship may contact each other more frequently. Choosing scholars with stronger coauthorship relationships as intermediaries will greatly help source scholars get in touch with target scholars and, at the same time, reduce the cost of time and other costs.

To solve this problem, we propose a coauthorship strength, author contribution, and search framework based on the Google Academic Platform, namely, CCS. Our idea is shown in Figure 2, which consists of the four following steps.

Step 1. Crawler technology obtains data. Python was combined with network packet capture and HTML packet capture to obtain real scholarly data on the Google Academic Platform.

Step 2. Construct coauthorship network. The nodes represent scholars, and the edges represent coauthorship.

Step 3. Calculate the coauthorship strength. Based on the calculation of the number of coauthors, the number of times they have cooperated, the number of citations, and the author contributions of different contributors, the coauthorship relationship is comprehensively measured.

Step 4. Use the shortest path algorithm to search. The Dijkstra and the Bellman–Ford algorithms are used to search for intermediaries and establish the connection paths between the source scholar and the target scholar.

4. Coauthorship Strength and Search Framework

4.1. Coauthorship Strength Calculation Model. Scientific research is the purposeful creation of knowledge or the arrangement of knowledge based on existing knowledge. With the increase in scholarly connections, the phenomenon of academic cooperation is becoming increasingly popular. Beaver et al. [8] proposed that the spirit of encouraging cooperation should be materialized, and the contributions of different authors should be distinguished in academic evaluations. Therefore, we establish a coauthorship strength model from different perspectives, such as the number of coauthors, the number of times they have cooperated, the number of citations of the coauthored paper, and scholarly contributions, to measure the coauthorship relationship accurately.

4.1.1. Coauthorship Index Calculation. There are many ways to calculate the edge rights in coauthorship networks. For example, according to the number of times scholars have collaborated, this method assumes that the coauthorship strength between coauthors in the same paper is equal; this may not be the same in practice, however, so this method has limitations. Therefore, Bormer et al. [5] proposed a new method to calculate coauthorship network edge weights, which not only considers the number of coauthors and the number of times they have cooperated but also takes into account the coauthorship effect, which is expressed by the number of citations. The formula is as follows:

$$w_{ij} = \frac{\left(1 + c_p\right)}{n_p\left(n_p - 1\right)}, \tag{1}$$

where n_p represents the number of coauthors of document p, c_p represents the total number of citations of document p, and w_{ij} represents the edge weight between the two nodes of author i and author j.

4.1.2. The Solution of Contribution Degree. Li et al. [9] evaluated the core authors in intelligence research and assessed the authors' contribution rate ranking method, which is mainly used to assign the weight of each author in the coauthored literature in descending order; nevertheless, the author who ranks first plays a leading role. Tang et al.

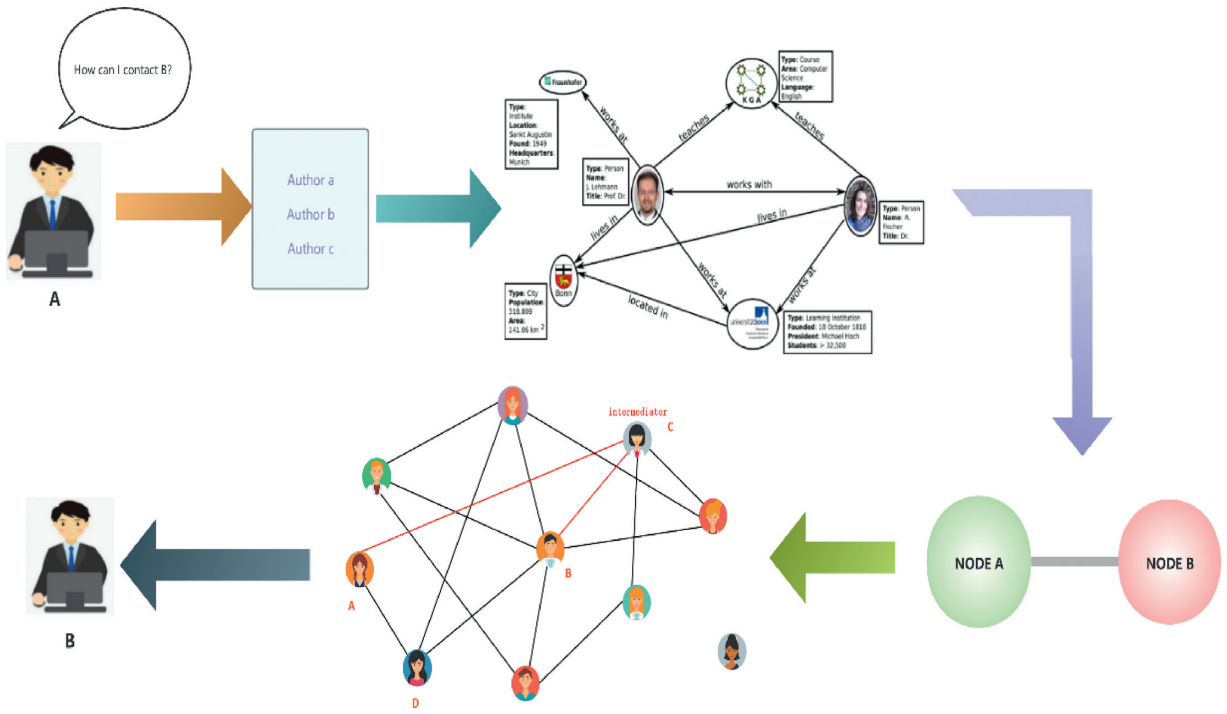


Figure 2: A coauthorship strength, author contribution, and search framework based on the Google Academic Platform.

[10] proposed the author contribution rate grade distribution method that uses an inverse proportional function to express the author contribution value and rank, as in the following formula:

$$W_i = \frac{1}{i\sum_{i=1}^{N} 1/i}, \tag{2}$$

where N represents the number of coauthors and i represents the rank of the authors.

Ling et al. [11] proposed a method for calculating inventor contribution based on the order of patent signatures, combined with the analysis of the topological characteristics of the patent inventor cooperation network, from the influence of inventors qualitatively and quantitatively, to measure personal innovation ability and domain cooperation ability, as in the two following formulas:

$$P_i = \frac{N - i + 1}{N}, \tag{3}$$

$$W_i = \frac{P_i}{\sum_{i=1}^{N} P_i}, \tag{4}$$

where N represents the number of coauthors, i represents the rank of the author, and P_i represents the contribution of the i-th inventor.

Sukhwan et al. [12] proposed a citation-based author contribution measurement method that is independent of the order of the authors in a publication and captures the importance of the first and last authors. They posit that the number of citations by researchers is the degree of recognition in the academic field and an indispensable basis for the measurement of research quality, as in the following formula:

$$w_k = \frac{c_k}{\sum_{k=1}^{N} c_k}, \tag{5}$$

where $k\,i$ is the position of the author in the signature, N is the total number of authors, and c_k is the number of citations of the k-th author.

The authors' contribution calculated by this method is relative to their citations count, not their position in a particular byline. This allows the calculation of signatures that are deliberately sorted alphabetically.

Zuckerman et al. [13] believed that, based on a long-term tradition, there must be a corresponding author at the end of the signature. As the most important author in the authors list, half of the total credits can be given to the last author, as in the following formula:

$$w_k = \begin{cases} 0.5, & k = A, \\ \dfrac{1}{2(A-1)}, & k = 1, \ldots, A-1, \end{cases} \tag{6}$$

where k is the position of the author in the signature and A is the total number of authors.

The Global Nature Index [9] proposed the $1/N$ evaluation model, which assumes that the contribution of each author in the same article is the same. However, the author signature order reflects the different scholarly contributions to a certain extent. Therefore, Du et al. [14] revised the estimation formula as follows:

$$P_{ij} = \begin{cases} 1 - \sum_{i=2}^{N} P_{ij}, \\ \\ \dfrac{1}{\text{Order}_j\ (i) +\ (N-1)}, \end{cases} \tag{7}$$

where P_{ij} is the contribution value of author i with $\text{Order}_{(i)}$ to paper j and N represents the total number of authors.

It is generally believed that scholars in different signature orders have made different contributions to the scientific research results. The academic community is accustomed to placing the names of scholars with the greatest contributions first. Therefore, to distinguish the degree of knowledge exchange between scholars in the same coauthored paper, this article also considers adding the authors' contribution to more accurately measure scholarly coauthorship strength. The formula proposed in this article is as follows:

$$W_{ij} = \begin{cases} P_{ij} - \sum_{i=2}^{N} W_{ij}, \\ \\ P_{ij} * \left(\dfrac{1 + c_p}{n_p (n_p - 1)} \right), \end{cases} \tag{8}$$

where n_p represents the number of coauthors of document p, c_p represents the total number of citations of document p, and w_{ij} represents the edge weight between the two nodes of author i and author j. P_{ij} is the contribution value of author i to paper j, and N represents the total number of authors.

4.2. Optimal Path Search Algorithm. The shortest path problem aims to solve how a search can minimize the sum of the weights of the edges. The Dijkstra algorithm, heuristic search algorithm, and Bellman–Ford algorithm commonly use shortest path algorithms [15]. The coauthorship strength, author contribution, and search framework proposed in this article selects scholars with stronger coauthorship strength as the intermediaries, which is more likely to increase the probability that source scholars successfully find target scholars. Since the edge weight of the coauthorship network is defined as the inverse of coauthorship strength, we need to select the edge with the greater coauthorship strength, that is, the smaller the weight that should be selected during each search, in which case the shortest path search algorithm is applicable.

4.2.1. Dijkstra Search. The Dijkstra search algorithm was first proposed by the Dutch computer scientist E. W. Dijkstra [16]. The algorithm searches for the node closest to the starting point each time, determines the length of the path from the starting node to the node, and then checks whether the shortest path length from the vertex to the end node has decreased [17]. In concrete, the pseudocode of this method is specified formally in Algorithm 1.

The Dijkstra algorithm is the most basic and most widely used algorithm for finding the shortest path. When finding the shortest path from a certain node (source point) in the network to the rest of the nodes, the classic Dijkstra algorithm divides the nodes in the network into three parts: unmarked nodes, temporarily marked nodes, and shortest path nodes (permanently marked nodes). At the beginning of the algorithm, the source point is initialized as the shortest path node, and the rest are unmarked nodes. During the execution of the algorithm, each time from the shortest path node to the neighboring node, the neighboring node of the nonshortest path node is modified to a temporarily labeled node, as well as the right to judge. After the value is updated, the node with the smallest weight from all the temporary marked nodes is extracted and then modified as the shortest path node and used as the next expansion source, and then the previous steps are repeated. When all nodes have expanded sources, the algorithm ends. More details can be found in Algorithm 1.

4.2.2. Bellman–Ford Search. The Bellman–Ford algorithm was invented by American mathematicians Chad Bellman and Lester Ford Jr [18]. The algorithm repeatedly judges each edge in the graph by an iterative method so that the estimated value of the shortest path from the starting node to other vertices gradually approximates its shortest distance [17]. The steps of the Bellman–Ford algorithm are as follows:

(1) Initialize the shortest distance from all points to the starting point to infinity and the distance from the starting point to itself to be zero

(2) Traverse each edge in the edge set array E and perform relaxation operations

(3) Check in turn whether the two vertices of each edge in the edge set array E converge

To solve the shortest path problem between two given nodes in the graph, the Dijkstra search algorithm and the Bellman–Ford search algorithm are better solutions. They are widely used in various fields, such as routing algorithms in computer networks, intelligent robot path-finding problems, and navigation of traffic routes [19]. They are also effective in searching for the optimal connection path between the two scholars by coauthorship strength proposed in this article.

In this article, we have made improvements when applying the Dijkstra algorithm and the Bellman–Ford search algorithm. After using the number of coauthors, the number of citations of the paper, and the contributions of scholars to find the coauthoring strength, we take the reciprocal of the coauthoring strength as the weight of the coauthored edges. Considering that, in real life, scholars with greater coauthoring strength are selected as intermediaries, the closer the coauthoring relationship, the higher the success rate of successfully introducing scholars who do not know each other. However, when applying the shortest path length algorithm to search, each time the edge with the smallest weight connected to the current node is selected; therefore, we take the reciprocal of the joint strength as the weight.

```
Inputs:
  s: start of path, e: end of path
  G: undirected weighted graph composed of all nodes
Output:
path: shortest path from node start to node end
dist: shortest path length from node start to node end
      for v_i ∈ G − {s} do
          dist[s, v_i] = w(s, v_i)
            if dist[s, v_j] + w_{j,i} < dist[s, v_i] then
            dist[s, v_i] = w_j + dist[s, v_j]
      if v_j == e then return dist[s, e], path
```

Algorithm 1: Dijkstra algorithm.

```
Inputs:
s: start of path
e: end of path
G: undirected weighted graph composed of all nodes
Output:
path: shortest path from node start to node end
dist: shortest path length from node start to node end
for i = 1 to |G.V| − 1
        for each edge(u, v) ∈∈G.E
        RELAX(u, v, w)
for each edge(u, v)∈∈G.E
        if v.d > u.d + w(u, v)
           return FALSE
return TRUE
```

Algorithm 2: Bellman–Ford algorithm.

5. Experiments

5.1. Experimental Configuration. A dataset from crawler technology is used for feasibility validation purposes. We crawled 5,201 papers and 11,191 authors' data from the Google Academic Platform. The numbers of coauthors and paper citations can be obtained directly from the personal pages and the number of cooperation times can be calculated from the dataset. For experimental comparison, four competition methods were implemented and tested; the running configuration included hardware settings (2.40 GHz CPU, 16 GB RAM) and software settings (Windows 10 and Python 3.7).

The four comparison solutions for comparison are the following:

(1) Reverse [10]: a method that uses an inverse proportional function to express author contributions

(2) Forward [11]: a calculation method based on the forward sequence of signatures.

(3) Cite [12]: a citation-based author contribution method that emphasizes the first and last authors

(4) Last [13]: a method that considers the last author to be the most important author

Evaluation metrics include the following:

(1) The shortest path length: smaller is better for measuring the path length (the sum of weights) when finding the target scholar

(2) Computational memory: smaller is better for measuring the memory usage of the algorithm when finding the target scholar

(3) Computation time: smaller is better for measuring the time consumed of the algorithm when finding the target scholar

5.2. Results Comparison

5.2.1. The Shortest Path Length (SPL). In this test profile, we compare 5 related methods to find SPL from the source scholar to the target scholar. In the experimental setting, the x-axis represents the number of nodes (the number of scholars), and the y-axis represents SPL when using different coauthor strength calculation methods. The experiment's comparison is reported in Figure 3.

Figure 3 shows the CCS that we proposed has the smallest shortest path length compared to the other methods. Since the weight of the edge is the reciprocal of

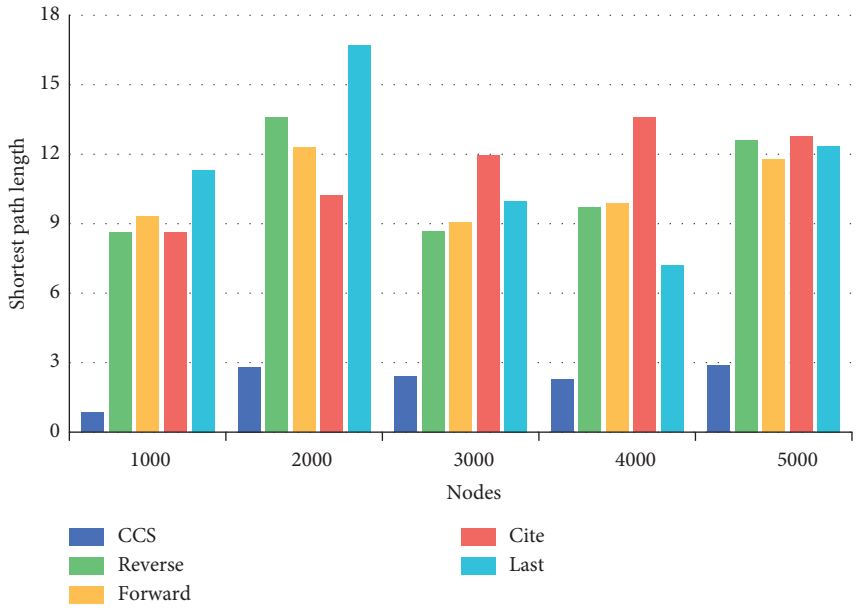


Figure 3: Comparison of the shortest path length with the number of nodes.

coauthorship strength, using the shortest path search algorithm to select a node with a smaller weight actually represents choosing a scholar with a closer coauthorship strength as an intermediary.

As the number of nodes increases from 1,000 to 5,000, SPL of CCS that we proposed is always stable within 3, and the magnitude of the change is small. Meanwhile, the four methods of comparison are basically above 6, especially when the number of nodes is 2000; Last reached 15 during the search. According to this dataset, CCS had the best results on the smallest shortest path length, Reverse and Forward had a similar effect, and Last needed the longest shortest path length.

This means that CCSs are more inclined to look for scholars who have greater coauthorship strength because the greater the coauthorship strength, the smaller the weight and the shorter SPL. Through intermediary scholars who have greater coauthorship intensity, the contact probability of source scholars and target scholars will increase. Therefore, the CCS proposed in this article performs best in terms of the smallest shortest path length.

5.2.2. Computation Memory (CM). In this test profile, we measure and compare the memory usage of five related methods when applying the Dijkstra search algorithm to search for target scholars, where the number of text inputs ranges from 1,000 to 5,000. Because different coauthorship strength calculation methods obtain different weights, there is a gap in memory usage when searching from the source scholar to the target scholar. Concrete comparison results are demonstrated in Figure 4.

Concretely, Reverse, Forward, Cite, and Last do not perform well in terms of memory usage. Reverse and Forward do not consider the number of coauthors or the number of papers cited. Cites are based on citations, ignoring the position of scholars in the order of specific signatures. Last only considers the order of signatures, assuming that the last scholar is the most important. In other words, these four algorithms cannot guarantee a comprehensive measurement of the strength of coauthorship.

In contrast, the CCS has better memory performance than the former four coauthorship strength algorithms. As the number of nodes continues to increase, memory usage is small. When the number of nodes is 1,000, the memory occupies 57 MB, and when the number of nodes is 5,000, the increase is approximately 4919 MB; thus, the effect is better.

5.2.3. Computation Time (CT). In this experiment, we test the efficiency and compare the time taken of five related methods. In the experimental setting, the x-axis represents the number of nodes (the number of scholars), the y-axis represents the time taken when using different coauthorship strength calculation methods, and the number of nodes = {1,000, 2,000, 3,000, 4,000, 5,000}. The consumed computational time of the five algorithms is presented in Figure 5.

Experimental data show that the time costs of the five algorithms all increase with the growth of nodes, as more author data often require additional search time. Furthermore, CCS runs more quickly than the former four coauthorship strength algorithms. Reverse and Forward methods require more computational time when applying the Dijkstra search algorithm to search for target scholars. Among these, Last consumes the most time, which has been validated by the data reported in Figure 5.

It should be noted that when the number of nodes is less than 4,000, the time taken by the five methods increases relatively smoothly. However, when the number of nodes is more than 4,000, the time taken will increase rapidly. CCS occupies the least time close to 30 s when the number of nodes is 1,000, while the most time taken is close to 120 s

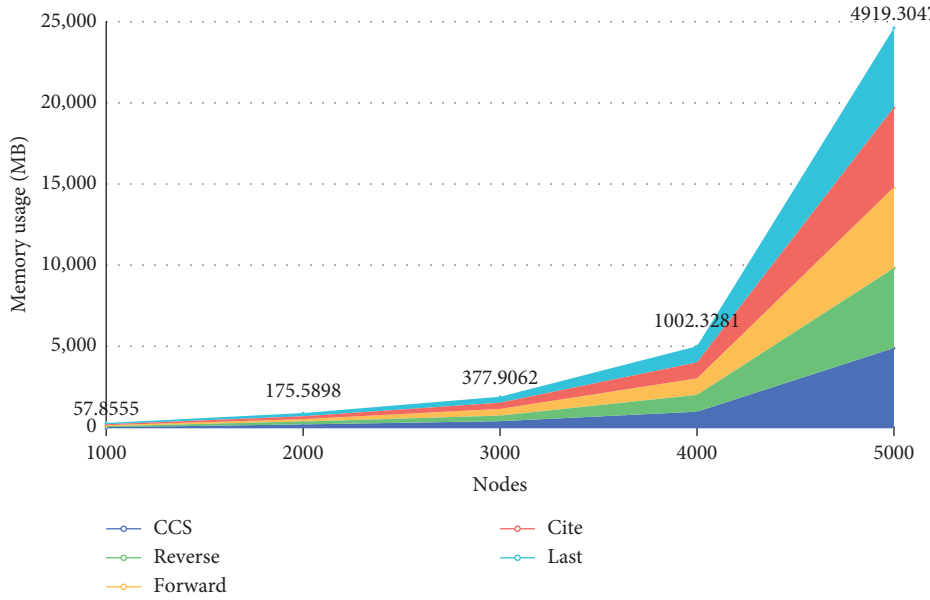


Figure 4: Comparison of the memory usage with the number of nodes.

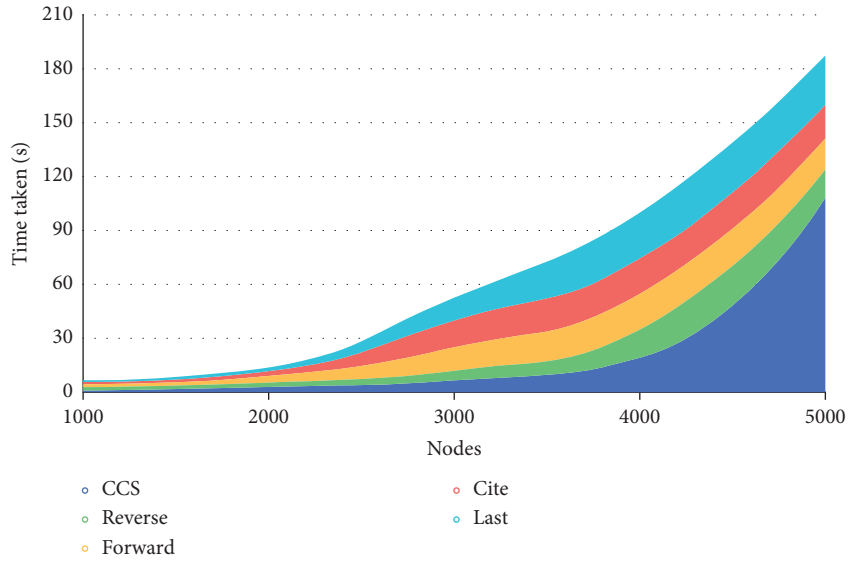


Figure 5: Comparison of the time taken with the number of nodes.

when the number of nodes reaches 5,000. From Figures 4 and 5, a comprehensive conclusion that our CCS has good performance in the areas of memory usage and time taken could be drawn.

In addition, we also verified the feasibility by calculating the number of intermediary scholars needed to find the target scholars shown in Figure 6. It can be seen that, for this dataset, applying the CCS that we have proposed to calculate the coauthorship strength, the number of intermediaries required for searching is one or two more intermediary scholars compared to those required by the other four methods. However, combined with the comprehensive analysis of the shortest path length, time, and memory consumption, the CCS is more inclined to find intermediary scholars with greater coauthor strength, which makes the probability of contacting the target scholar higher.

5.3. Further Discussions. In this test, we also used the Dijkstra algorithm and Bellman–Ford algorithm to conduct search experiments. The experimental comparison is reported in Figure 7.

Figure 7 shows that as the number of nodes in the dataset increases, that is, as the coauthorship network of scholars continues to increase, when using the Dijkstra and Bellman–Ford algorithms to search for intermediary scholars to find target scholars of interest, the required time and memory usage continue to increase, which is in line with

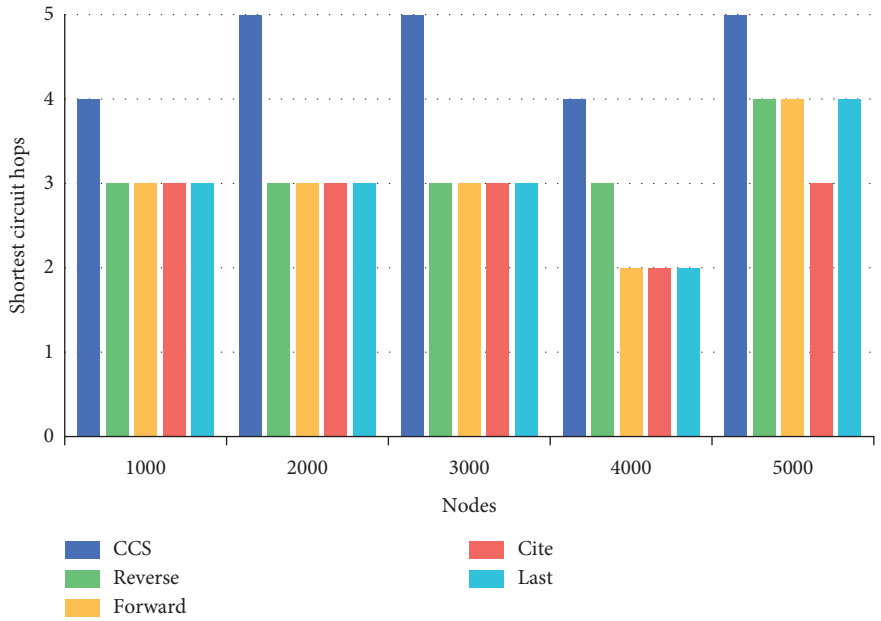


Figure 6: Comparison of the intermediate nodes with the number of nodes.

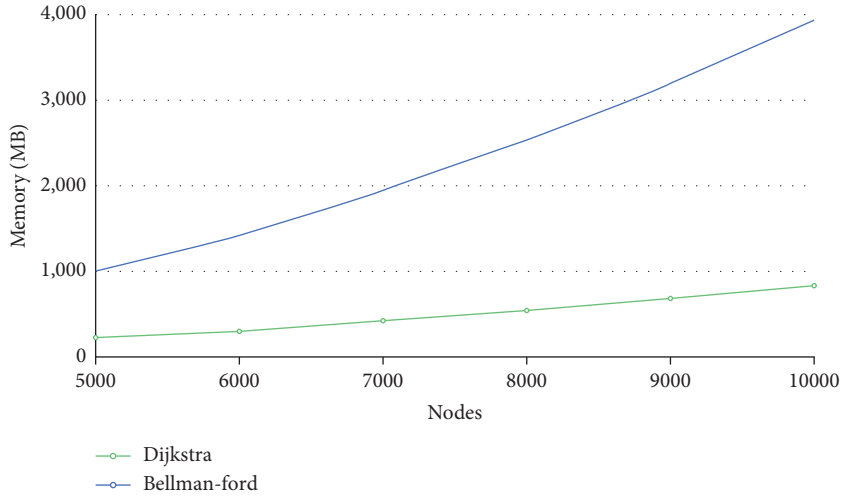


Figure 7: Memory usage of the Dijkstra algorithm and Bellman–Ford algorithm with the number of nodes.

reality. It can also be observed that memory usage of Bellman–Ford algorithm is always higher than that of the Dijkstra algorithm and increases at a very rapid rate; when the number of nodes is 5000, the memory required by the algorithm is 1000 MB, which is much larger than that of the Dijkstra algorithm. Compared with the Dijkstra search algorithm, the Bellman–Ford search algorithm, based on the coauthorship strength calculation method that we have proposed, is more effective in terms of memory usage.

6. Conclusion

With the development of Internet information technology, academic communication and cooperation are no longer restricted by geographical location. Nevertheless, it is an increasingly challenging task to discover useful knowledge resources. How to help scholars quickly find interested target collaborators, encourage them to participate more actively, and create higher-quality achievements has become a significant problem.

Considering this challenge, we propose a coauthorship strength, author contribution, and search framework in this article, based on the Google Academic Platform, which obtains real scholarly data from Google Scholar through crawlers and establishes a scholarly coauthored network. In this way, we take into consideration multiple indicators to ensure accurate measurement of coauthorship. Moreover, we combined it with a search algorithm to better solve practical application problems.

Finally, we validate the advantages of the CCS framework that we proposed through a set of experiments using real-world data from the Google Academic Platform. As a result, we find that the coauthorship model proposed in this article is more likely to choose scholars with stronger coauthorship intermediaries. In practice, intermediary scholars who are more closely connected can improve the probability that source scholars can quickly find target scholars.

In the future, we will introduce more academic indicators, such as user trust [20–22] and time context [23–28]. In addition, computational cost or time cost is a key concern when the dataset to be processed is big [29–39]. Therefore, we will further optimize our proposed method to accommodate the big data applicable scenarios.

Data Availability

The dataset can be accessed at https://www.aminer.cn/data/?nav=openData.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] T. Wan, S. Yue, and W. Liao, "Privacy-preserving incentive mechanism for mobile crowdsensing," *Security and Communication Networks*, vol. 2021, Article ID 4804758, 17 pages, 2021.

[2] H. Liu, J. Guo, J. Li, F. Mei, and H. He, "An A~ algorithm based on random walk," *Journal of Civel Aviation University of China*, vol. 35, no. 6, pp. 61–64, 2017.

[3] D. Zou, "Evolution analysis of scientific research co authorship network in the field of computer science [J/OL]," *Knowledge Management Forum*, vol. 1, no. 2, pp. 130–135, 2016.

[4] R. Xie, X. Li, X. Han, and S. Shi, "Author influence evaluation index construction based on weighted citation frequency and signature order," *Information Science*, vol. 36, no. 8, pp. 90–93+111, 2018.

[5] L. Zhu and J. Yu, "Research on the weighted model of Co-author relationship network," *Library and Information Service*, vol. 54, no. 12, pp. 69–73, 2010.

[6] Y. Han, B. Zhou, J. Pei, and Y Jia, "Understanding importance of collaborations in Co-authorship networks: a supportiveness analysis approach," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, Sparks, Nevada, May 2009.

[7] T. Amjad, A. Daud, D. Che, and A. Akram, "MuICE: mutual influence and citation exclusivity author rank," *Information Processing & Management*, vol. 52, no. 3, pp. 374–386, 2016.

[8] D. B. Beaver and R. Rosen, "Studies in scientific collaboration: Part Il--Professionalization and the natural history of modern scientific co- authorship," *Scientometrics*, no. 1, pp. 231–245, 1979.

[9] L. Li and Z. Zhang, "Research on the impact evaluation method of core authors in information research," *Journal of Information*, vol. 29, no. 10, pp. 80–83, 2010.

[10] Q. Tang and Y. Wang, "Core author evaluation and collaborative network research based on field contribution value," *Information Theory and Practice*, vol. 38, no. 1, pp. 85–89, 2015.

[11] Y. Ling, "Research on patent inventor influence evaluation based on contribution degree and cooperation network analysis," *Journal of Agricultural Library and Information Science*, vol. 30, no. 9, pp. 27–32, 2018.

[12] S. Jung and W. C. Yoon, "Citation-based author contribution measure for byline-independency," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, pp. 6086–6088, IEEE, Los Angeles, CA, USA, December 2019.

[13] H. A. Zuckerman, "Patterns of name ordering among authors of scientific papers: a study of social symbolism and its ambiguity," *American Journal of Sociology*, vol. 74, no. 3, pp. 276–291, 1968.

[14] https://www.novopro.cn/articles/2015%2007221217.html1811.

[15] https://wiki.mbalib.com/wiki/Dijkstra%E7%AE%97%E6%B3%95.

[16] S. Idwan and W. Etaiwi, "Dijkstra algorithm heuristic approach for large graph," *Journal of Applied Sciences*, vol. 11, no. 12, pp. 2255–2259, 2011.

[17] W. Han, "Fixed order. An improvement of Bellman-Ford algorithm," *Journal of Harbin Institute of Technology*, vol. 46, no. 11, pp. 58–62, 2014.

[18] Y. Cao and J. Ma, "Optimization of traditional Chinese medicine delivery route based on improved bellman-ford algorithm," *Journal of Hebei North University (Natural Science Edition)*, vol. 36, no. 3, pp. 18–21, 2020.

[19] W. Zhao, Z. Gong, W. Wang, and S. fan, "Comparative analysis of several classical shortest path algorithms," *Journal of Chifeng University (Natural Science Edition)*, vol. 34, no. 12, pp. 47–49, 2018.

[20] F. Wang, H. Zhu, G. Srivastava, S. Li, M. R. Khosravi, and L. Qi, "Robust collaborative filtering recommendation with user-item-trust records," *IEEE Transactions on Computational Social Systems*, pp. 1–11, 2021.

[21] W. Zhang, Z. Li, and X. Chen, "Quality-Aware user recruitment based on federated learning in mobile crowd sensing," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 869–877, 2021.

[22] H. Kou, H. Liu, Y. Duan et al., "Building trust/distrust relationships on signed social service network through privacy-aware link prediction process," *Applied Soft Computing*, vol. 100, Article ID 106942, 2021.

[23] X. Yang, X. Jia, M. Yuan, and D.-M. Yan, "Real-time facial pose estimation and tracking by coarse-to-fine iterative optimization," *Tsinghua Science and Technology*, vol. 25, no. 5, pp. 690–700, 2020.

[24] L. Qi, R. Wang, C. Hu, S. Li, Q. He, and X. Xu, "Time-aware distributed service recommendation with privacy-preservation," *Information Sciences*, vol. 480, pp. 354–364, 2019.

[25] P. Nitu, J. Coelho, and P. Madiraju, "Improvising personalized travel recommendation system with recency effects," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 139–154, 2021.

[26] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for IoV in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[27] Y. Jin, W. Guo, and Y. Zhang, "A time-aware dynamic service quality prediction approach for services," *Tsinghua Science and Technology*, vol. 25, no. 02, pp. 227–238, 2020.

[28] M. S. Mahmud, J. Z. Huang, S. Salloum, T. Z. Emara, and K. Sadatdiynov, "A survey of data partitioning and sampling methods to support big data analysis," *Big Data Mining and Analytics*, vol. 3, no. 2, pp. 85–101, 2020.

[29] X. Xu, Q. Huang, H. Zhu et al., "Secure service offloading for internet of vehicles in SDN-enabled mobile edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3720–3729, 2021.

[30] R. Bi, Q. Liu, J. Ren, and G. Tan, "Utility aware offloading for mobile-edge computing," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 239–250, 2021.

[31] Z. Tong, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[32] X. Xu, Q. Huang, Y. Zhang, S. Li, L. Qi, and W. Dou, "An LSH-based offloading method for IoMT services in integrated cloud-edge environment," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 3s, pp. 1–19, 2021.

[33] J. Guo, H. Liang, S. Ai, C. Lu, H. Hua, and J. Cao, "Improved approximate minimum degree ordering method and its application for electrical power network analysis and computation," *Tsinghua Science and Technology*, vol. 26, no. 4, pp. 464–474, 2021.

[34] Y. Bie and Y. Yang, "A multitask multiview neural network for end-to-end aspect-based sentiment analysis," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 195–207, 2021.

[35] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, and M. Z. A. Bhuiyan, "Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5213–5222, 2021.

[36] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. E. Hajjaji, "IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32, 2021.

[37] X. Xu, Q. Huang, X. Yin, M. Abbasi, M. R. Khosravi, and L. Qi, "Intelligent offloading for collaborative smart city services in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7919–7927, 2020.

[38] J. Cai, Z. Huang, L. Liao, J. Luo, and W.-X. Liu, "APPM: adaptive parallel processing mechanism for service function chains," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1540–1555, 2021.

[39] J. Luo, J. Li, L. Jiao, and J. Cai, "On the effective parallelization and near-optimal deployment of service function chains," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1238–1255, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6571700, 10 pages
https://doi.org/10.1155/2021/6571700

Research Article

pKAS: A Secure Password-Based Key Agreement Scheme for the Edge Cloud

Ping Liu,[1] **Syed Hamad Shirazi**,[2] **Wei Liu**,[1] **and Yong Xie**[1]

[1]*Department of Computer Technology and Application, Qinghai University, Xining, China*
[2]*Department of Information Technology, Hazara University, Baffa, Pakistan*

Correspondence should be addressed to Yong Xie; mark.y.xie@qq.com

Received 5 September 2021; Accepted 5 October 2021; Published 18 October 2021

Academic Editor: Xiaolong Xu

For the simplicity and feasibility, password-based authentication and key agreement scheme has gradually become a popular way to protect network security. In order to achieve mutual authentication between users and edge cloud servers during data collection, password-based key agreement scheme has attracted much attention from researchers and users. However, security and simplicity are a contradiction, which is one of the biggest difficulties in designing a password-based key agreement scheme. Aimed to provide secure and efficient key agreement schemes for data collecting in edge cloud, we propose an efficient and secure key agreement in this paper. Our proposed scheme is proved by rigorous security proof, and the proposed scheme can be protected from various attacks. By comparing with other similar password-based key agreement schemes, our proposed scheme has lower computational and communication costs and has higher security.

1. Introduction

With the dawn of the Internet of everything, Internet of things (IoT) has become to obtain the leading strategic position in research and development in the world. Even though various countries in the world pay attention to the development of the IoT, the influx of diverse traffic and the need of diversified application scenario has not only put forward new challenge for the centralized cloud computing architecture nowadays but also drove the emergence of the cloud computing paradigm [1, 2].

In the era of Internet of Things, mobile devices are no longer simple mobile phones, tablets, etc., but include more abundant augmented/virtual reality devices, intelligent medical device, and moving vehicle. The application scenario also transfers from voice/video communication and other services to virtual space experience, intelligent manufacturing, and the Internet of vehicles [3, 4]. In cloud-based services, data transmission speed will be affected by network traffic, and heavy traffic will lead to long transmission time, thus increasing power consumption cost. Therefore, the adoption of mobile edge computing (MEC) can meet the needs of IoT devices.

As shown in Figure 1, the collection and processing of data is a very important part of the Internet of Things. However, all collected data will be transmitted to the cloud server and then rely on the server's computing power for data processing and analysis. This will cause the server to be heavily loaded and prone to failure or downtime. At the same time, the increase in the amount of data will also increase the cost of the storage server. In addition, because the network is limited by the network bandwidth and speed, the network bandwidth is put under pressure when a large amount of monitoring data is transmitted, and the data may have large transmission delays and packet loss during transmission. Edge computing data provides format conversion, caching, processing, analysis, and transmission services, and the load of cloud servers improves the efficiency of data processing. The edge cloud includes IoT gateways and collectors. These devices together form an edge node network and provide lightweight computing power for the edge layer of the system.

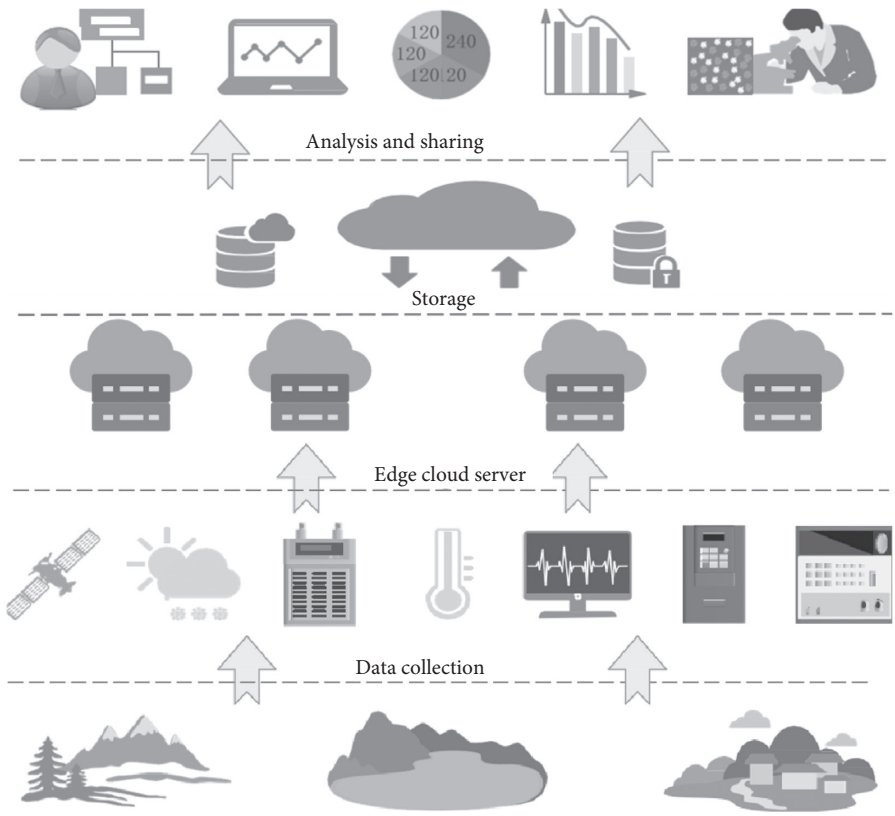


Figure 1: A typical data sharing model in edge cloud computing environment.

In the MEC-based Internet of Things, massive amounts of data are generated by a large number of sensors and various heterogeneous devices, and all storage devices are provided by different third-party vendors. Due to the distributed nature of MEC, data are stored in different network edges, which will increase the risk of data being attacked. For example, unauthorized users or opponents may modify or abuse the data uploaded in the storage, which will lead to data leakage and other problems. In order to solve these problems, this paper proposes identity verification based on password-based key agreement. This scheme can ensure both sides' identity authentication and data security.

In order to protect the data in the edge cloud from being tampered with, the administrator of the edge cloud server needs to authenticate with it when operating the server, so the sever can determine whether the administrator has been faked. To improve the security and verifiability of messages, Zheng [5] proposed a signcryption scheme, which can simultaneously sign and encrypt.

The key agreement protocol is the most commonly used method for two or more parties to communicate. Features of the protocol ensure that the data to be communicated are confidential, secure, and complete [6–10]. The protocol is to establish a session key jointly by two or more entities. The result of key agreement will be affected by any participant, and no trusted third party is required in the process. The session key is obtained by calculating the parameters generated by the participants. In order to enable both parties to authenticate each other, an authentication key agreement is proposed, and the protocol established a session key [11–13].

In 2005, the Diffie–Hellman key exchange in the encryption assumption protocol system is a secure and scalable authentication key exchange agreement, which performs key control and management during transmission [14–16]. In 2009, the elliptic curve cryptosystem (ECC) authentication scheme based on no pairing and few certificates was presented. The scheme was based on mobile devices communication and ID authentication with key agreement protocol. Furthermore, the proposed scheme is also to overcome more attacks [13, 17–20]. Many scholars believed that large prime numbers is difficult for hardware implementation of the elliptic curve cryptosystem, while the binary field was known as suitable [21, 22] in 2010–2012. In order to ensure the confidentiality and integrity of the sent and received messages, the authentication key agreement protocol must include a strong encryption algorithm. The key agreement protocol based on elliptic curve cryptography provides an important development for confidentiality, integrity, and user anonymity.

There are two types of key agreement protocols according to different authentication methods: password-based key agreement protocols and public-key-based key agreement protocols. The password-based authentication key agreement protocol was first proposed by Bellovin and Merritt [23]. In this protocol, both parties share a password in advance, which is used to authenticate each other's identity during communication and negotiate a short-term

session key. Public key-based key agreement can negotiate a session key through signature or public key verification. In this paper, password-based key agreement protocol is studied [15].

1.1. Motivations and Contributions. The proposed pKAS can ensure the security of the message and the authentication of the user identity when two parties communicate. We list our contributions as follows:

First, we put forward a secure password-based key agreement pKAS based on ECC for mutual authentication between the user and edge server. The proposed pKAS only needs to deliver the message twice, which greatly saves communication bandwidth. And, in this scheme, we use signcryption, signature verification, and hash operation etc., to ensure the confidentiality and integrity of the message, as well as the anonymity of the identity.

Second, we conduct strict security analysis on the proposed pKAS and compare it with other related schemes. The results show that the presented pKAS can resist various attacks.

Third, by comparing communication and calculation costs, the proposed pKAS has lower cost and is more secure than recent similar schemes.

1.2. Organization of the Paper. The structure of the paper is as follows. Sections 2 and 3 present the related works and the preliminaries. The system model and security requirements of the scheme proposed in this paper are shown in Section 4. Section 5 presents the proposed password-based key agreement scheme. Section 6 presents the performance and security analysis. Section 7 describes conclusion, future work, conflicts of interest, and data availability respectively.

2. Related Works

With the development of Internet technology, security in communications has become more and more significant. Therefore, how to identify remote users has become one of the most significant issues in the public network. In order to figure out the problem, many schemes have been presented. Lamport [24] first proposed the password-based scheme to ensure remote parties authentication scheme. Subsequently, many password-based key agreement schemes were proposed in [25–29].

In 2009, Xu et al. [25] presented an improved remote user authentication and key agreement scheme based on passwords and smart cards, and they certificated that their scheme is secure. Sood et al. [26] found that Xu et al.'s scheme is ineffective against password guessing attacks and impersonation attacks. Subsequently, Sood et al. put forward an improved authentication scheme. However, in 2012, Chen et al. [27] analyzed and pointed out that the scheme of Sood et al. only provided a single-party authentication function, and the legitimacy of the remote server was not authenticated. As a consequence, an improved key agreement scheme with stronger security was presented by Chen et al.\enleadertwodots, and the scheme achieved remote parties' authentication. Furthermore, they stated that their scheme could resist kinds of attacks. In those authentication schemes proposed by Sood et al., Chen et al., and many scholars [30–32], users must interact with the remote server to transmit information and repeat the login process and authentication process instead of completing the password change process on the client when he/she wants to change the password. In addition, these solutions will not find the wrong password entered during the login process. The wrong password can only be found in the final authentication process after a series of calculations and communications. Obviously, these schemes were inefficient and user-unfriendly, and failed to verify wrong password. Recently, Li et al. [28] analyzed that Chen et al.'s scheme could not ensure forward security and does not achieve perfect user anonymity. In addition, they proposed a scheme based on password and smart card, and the scheme can enhance remote user authentication and key agreement.

The message transmitted between the sender and the receiver may be eavesdropped by the adversary through public channels. The identity of users should be kept confidential during message transmission. Otherwise, the adversary will track the user by collecting the user's identity information. Some interesting bilinear pairing-based and ECC-based key agreement protocols were proposed in recent years [33–36]. Irshad et al. [33] presented the scheme which used bilinear pairing operations in the interaction between mobile devices and servers. A method that can use mobile devices to access the server was proposed by Tsai and Lo [35], but later proved that the scheme cannot resist impersonation attacks and man-in-the-middle attacks. It is a pity that Xiong et al. [37] believe that Irshad et al.'s scheme is very computationally expensive for mobile devices. The protocol based on ECC is more efficacious because point addition or multiplication in elliptic curves is more efficient than modular exponents. In addition, the elliptic curve encryption protocol which is based on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP) is more secure. In 2017, a lightweight password-base key agreement protocol was proposed by Mahmood et al. [34]. But later, the program was verified to have some security issues, such as no anonymity, no resistance to replay attacks, and no guarantee of data confidentiality. Recently, a key agreement scheme based on ECC was presented by Kaur et al. [36], and they stated their scheme can overcome many kinds of attacks. Nonetheless, we strictly analyzed and found the scheme of Kaur et al. proposed suffered from no resistance forgery attack and insider attack.

3. Preliminaries

3.1. One-Way Hash Function. Let message m be a message that requires a hash value. The length of m is a variable, while h is the fixed length. Given m, it is easy to obtain h. However, given h, it is infeasible to obtain m.

3.2. Elliptic Curve Cryptosystem (ECC). In 1985, the elliptic curve was used for data encryption by Miller firstly. Later, Koblitz based on the elliptic curve discrete logarithm problem (ECDLP) built a new encryption system, which is called the elliptic curve cryptosystem (ECC). ECC has lower computational overhead than other public key cryptographies such as RSA. Since then, ECC has been widely used in cryptographic protocols and security schemes. The following describes the basic knowledge of ECC and computational difficulties in ECC.

Elliptic curve cryptography is a public key cryptography method based on elliptic curve mathematics. The commonly used expression of elliptic curve in finite field F_p is: $y^2 = x^3 + ax + b(\mathrm{mod}p)$ $(a, b \in F_p,$ and $(4a^3 + 27b^2)(\mathrm{mod}p) \neq 0)$, all coefficients are elements in a finite field F_p (where p is a large prime number). Let $E_p(a, b)$ denotes the point set $\{(x, y)|0 \leq x < p,\ 0 \leq y < p,$ and x, y are both integers$\}$ on the elliptic curve defined by the equation and the infinity point O.

The addition on $E_p(a, b)$ is defined as follows:

For any point in $E_p(a, b)$, $P = P + O$.

Let Q, R be the two points in $E_p(a, b)$. $Q + R$ is defined as follows: draw a straight line passing through Q, R and the elliptic curve to intersect point P, then $Q + R = -P$.

Let Q be a point in $E_p(a, b)$, and the multiples of Q are defined as follows: draw a tangent to the elliptic curve at point Q, and set the tangent to intersect the elliptic curve at point S; then, $2 \cdot Q = Q + Q = -S$. Similarly, $n \cdot Q = Q + Q + \cdots + Q$ (n times), where $n \in Z_p$, $n > 0$.

3.3. Complexity Assumptions. The security foundation of ECC is an elliptic curve discrete logarithm problem (ECDLP), which can be defined as follows.

ECDLP: assume two random points P_1 and P_2 in (E/E_p), $P_2 = kP_1$, where $k \in Z_p^*$. It is easy to compute P_2 if knows k and P_1, while it is infeasible to compute k if knows P_1 and P_2.

4. System Model and Security Model

4.1. System Model. On analysis of the requirements of communication between the user and edge server, there are two types of roles related in our system, such as users communicating with server, a trust authority (TA) can be regarded as a completely trusted administrator and cannot be compromised by any adversary. With a view to user authentication and key agreement, a user (Assumed be U_i) must be registered in the TA, and then he/she can perform mutual authentication and key agreement with edge cloud server other users (such as U_j) only using the password and smart card.

The network model of our system can be illustrated in Figure 2. Before the users communicate with the edge server, the users must register with the TA through a secure channel and store the corresponding registration information on her/his smart cards. After successful registration, users can perform mutual authentication and key negotiation through edge server and implement operations such as secure data management on the edge cloud.

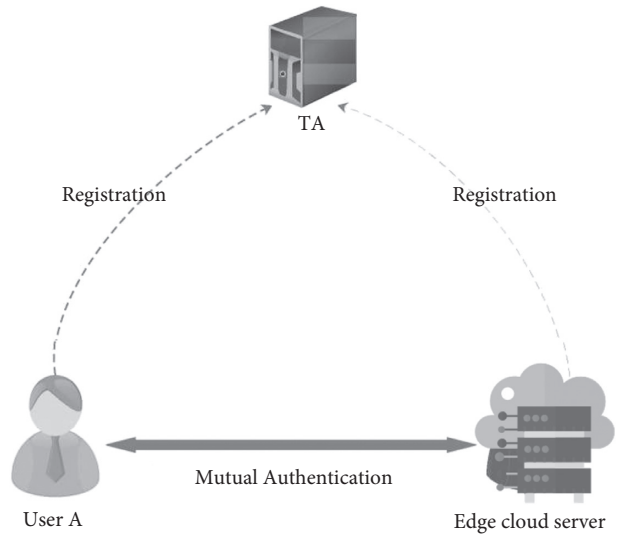


Figure 2: A typical key agreement model in edge cloud computing environment.

4.2. Security Requirements. Before analyzing security requirements, let us assume adversary's capabilities based on the application. An adversary $\mathscr{A}$ generally contains the following capabilities:

(i) The open channel can be controlled by $\mathscr{A}$, that is to say, the messages through the open channel $\mathscr{A}$ can be deleted, intercepted, modified, and resent

(ii) $\mathscr{A}$ can traverse the password space in polynomial time, that is, if it has known any other secret information, $\mathscr{A}$ can guess the password by brute force attack

(iii) $\mathscr{A}$ can obtain the user's password through a malicious terminal and can also extract data that are stored in smart card

On the capacities of the adversary $\mathscr{A}$, the security requirements of password-based key agreement scheme should include forward secrecy and must be resistant to know attacks, such as offline password guessing attack, replay attack, user impersonation attack, server spoofing attack, and parallel attack. Furthermore, the scheme must be mutual authentication and anonymity.

5. The Proposed Scheme (pKAS)

In this section, a key agreement scheme based on password (called pKAS for short) by using ECC was proposed. There are no bilinear paring operations in pKAS. Overall, pKAS has four phases: system initial phase, registration phase, login and key agreement phase, and offline password change phase. For simplicity, we list the symbols used in this paper and their corresponding meanings in Table 1.

Next, the following sections present the four phases of the proposed scheme.

5.1. System Initialization Phase. Trust authority (TA) is responsible for the system initialization phase. In this phase,

Table 1: Description of the symbols used.

Symbol	Description
TA	The trust authority
ID_i	The identity of user U_i
p	A big prime
$E_p(a,b)$	Point set of an elliptic curve: $y^2 = x^3 + ax + b \bmod p$
κ	A middle large integer
Δt	The limited time interval
P	The base point of the elliptic curve
G	A finite cycle additive group over the elliptic curve
$h(\cdot)$	One-way hash function
x_i	The secret key of user U_i
X_i	$X_i = x_i P$, the public key of user U_i

TA selects a big prime p; then, in finite field, F_p constructs a nonsingular ecliptic curve $E_p(a,b)$ and chooses base points P on $E_p(a,b)$ and generates a finite cycle additive group G of order q with P.

5.2. Registration Phase. Users, edge cloud sever, and TA complete the registration phase together. Assume the current user U_i's identity be ID_i, the registration is completed as follows:

Step R1: U_i sets his password PW_i, then chooses a random number $x_i, a_{i0} \in Z_q^*$, and computes $X_i = x_i P$, $a_{i1} = h(PW_i \| a_{i0})$, $a_i = h(h(ID_i) \oplus a_{i1} \bmod \kappa)$. At last, U_i sends $\{ID_i, X_i\}$ to TA in a channel that an adversary cannot eavesdrop on.

Step R2: when TA receives $\{ID_i, X_i\}$, it will store $\{ID_i, X_i\}$ in the server.

5.3. Login and Key Agreement Phase. We assume there are two users, user U_i and edge cloud sever U_j in this phase. They login by using their ID and password, then authenticate, and consult with session key each other.

Step A1: U_i inputs his/her ID_i' and PW_i', then smart cart computes $a_i' = h(h(ID_i) \oplus h(PW_i' \| a_{i0}) \bmod \kappa)$, and checks whether $a_i' = a_i$ holds or not. If it does not, the session is terminated.

Step A2: U_i randomly chooses $c_i \in Z_q^*$, computes $C_i = c_i P$, $PID_i = ID_i \oplus h(c_i X_j \| t_i)$, $f_i = h(ID_i \| ID_j \| PID_i \| C_i \| t_i)$, where t_i is current timestamp, $\sigma_i = c_i + x_i f_i \bmod q$. At last, U_i sends $M_1 = \{C_i, PID_i, t_i, \sigma_i\}$ to U_j.

Step A3: after receiving $\{C_i, PID_i, PID_j, t_i, \delta_i\}$, U_j checks whether $t_j - t_i < \Delta t$, if not, U_j terminates the session, else U_j computes $ID_i^* = PID_i \oplus h(x_j C_i \| t_i)$, $f_i{}' = h(ID_i^* \| ID_j \| PID_i \| C_i \| t_i)$ and checks whether $C_i = \sigma_i P - f_i' X_i$ holds or not. If not, U_j terminates the session. U_j chooses $C_j \in Z_q^*$ and computes $C_j = c_j P$, $sk_{ji} = h(ID_i \| ID_j \| t_i \| t_j \| C_i c_j)$, $f_j = h(ID_i \| C_i c_j \| sk_{ij} \| t_i \| t_j \| ID_j)$, $\delta_j = C_j + x_j f_j \bmod q$. At last, U_j sends $M_2 = \{C_j, f_j, \delta_j, t_j\}$ to U_i.

Step A4: after receiving $\{C_j, f_j, \sigma_j, t_j\}$ from U_j, U_i checks whether current timestamp t_i' meets $t_i' - t_j < \Delta t$ or not, if not, U_i terminates the session, else U_i computes $C_j' = \sigma_j P - f_j X_j$, $sk_{ij} = h(ID_i \| ID_j \| t_i \| t_j \| c_i C_j)$ and checks whether $f_j = h(ID_i \| C_i \| C_j \| sk_{ji} \| t_i \| t_j \| ID_j)$ holds or not. If not, U_i terminates the session, else U_i accepts this session.

At last, U_i and U_j have agreed an identical session key $sk_{ji} = sk_{ij}$. Figure 3 presents the flowchart of login and key agreement phase.

5.4. Offline Password Change Phase. In order to obtain a better user experience, while meeting the high requirements of security and efficiency, the user can complete this phase locally in the proposed scheme as follows:

Step C1: in order to verify the user's identity, the user must enter ID_i, PW_i in the smart card.

Step C2: the smart card computes $a_i' = h(hID_i) \oplus h(PW_i' \| a_{i0}) \bmod \kappa$ and checks if a_i' and a_i are equal. If not, the system will terminate the session. Else, it means the correctness of ID_i and PW_i is $\kappa - 1/\kappa \approx 99.61/100$, $\kappa = 2^8$, and it can go to the next step.

Step C3: user U_i inputs new password PW_i^{new} and computes $a_i = h(h(ID_i) \oplus h(PW_i^* \| a_{i0}) \bmod \kappa)$.

6. Security and Performance Analysis

Security analysis and proof of our scheme is presented in this section. As well as the proposed pKAS is proven to be able to resist all kinds of attacks. Besides, we analyze and compare the communication calculation and bandwidth consumption of similar schemes.

6.1. Security Analysis. In this section, the details of security analysis are described as following.

Proposition 1. *The proposed pKAS scheme can be secure against offline password guessing attack.*

Proof. Assume an adversary $\mathcal{A}$ has got U_i's smart card and obtained the data stored in the card. he/she can launch password guessing attack by the following steps:

Step D1: $\mathcal{A}$ guesses PW_i^* from password dictionary space and ID_i from identity diction space

Step D2: $\mathcal{A}$ retrieves a_{i0} and a_i and computes $a_i' = h(h(ID_i) \oplus h(PW_i^* \| a_{i0}) \bmod \kappa)$

Step D3: $\mathcal{A}$ checks whether $a_i' = a_i$ holds or not

Step D4: $\mathcal{A}$ repeats the step D1 to D3 until $a_i' = a_i$ holds

That is, $\mathcal{A}$ can guess correct ID_i and PW_i. However, $\mathcal{A}$ is still not sure they are the same identity and password. Then, $\mathcal{A}$ has to execute online guessing attack to test the correctness both. However, we use Hoeny_list to prevent online

User U_i		User U_j
Input $ID_i^{'}$ and $PW_i^{'}$,		
$a_i^{'} = h(h(ID_i) \oplus h(PW_i^{'}\|\|a_{i0}) \bmod \kappa)$,		
$a_i^{'} \stackrel{?}{=} a_i$		$t_j - t_i < \Delta t$,
$c_i \in Z_q^*, C_i = c_i P, PID_i = ID_i \oplus h(c_i X_j \|\| t_i)$,		$ID_i^* = PID_i \oplus h(x_j X_i \|\| t_i)$
$f_i = h(ID_i \|\| ID_j \|\| PID_i \|\| C_i \|\| t_i)$		$f_i^{'} = h(ID_i^* \|\| ID_j \|\| PID_i \|\| C_i \|\| t_i)$
$\sigma_i = c_i + x_i f_i \bmod q$	$\overrightarrow{M_1}$	$C_i \stackrel{?}{=} \sigma_i P - f_i^{'} X_i$
$M_1 = \{C_i, PID_i, t_i, \sigma_i\}$		$c_j \in Z_q^*, \ C_j = c_j P$
		$sk_{ji} = h(ID_i \|\| ID_j \|\| t_i \|\| t_j \| C_i c_j)$
		$f_j = h(ID_i \|\| C_i \|\| C_j \|\| sk_{ij} \|\| t_i \|\| t_j \|\| ID_j)$
		$\sigma_j = c_j + x_j f_j \bmod q$
		$M_2 = \{C_j, f_j, \sigma_j, t_j\}$
$t_i^{'} - t_j < \Delta t$	$\overleftarrow{M_2}$	
$C_j \stackrel{?}{=} \sigma_j P - f_j X_j$		
$sk_{ij} = h(ID_i \|\| ID_j \|\| t_i \|\| t_j \| c_i C_j)$		
$f_j \stackrel{?}{=} h(ID_i \|\| C_i \|\| C_j \|\| sk_{ij} \|\| t_i \|\| t_j \|\| ID_j)$		
Identical session key $sk_{ji} = sk_{ij} = h(ID_i \|\| ID_j \|\| t_i \|\| t_j \| c_i c_j P)$		

Figure 3: Login and key agreement phase.

guessing attack. As a result, the proposed pKAS can be secure against offline password guessing attack. □

Proposition 2. *The proposed pKAS scheme can be secure against online password guessing attack.*

Proof. In order to eliminate the threat of online password guessing attack, Hoeny_list is adopted in the proposed scheme. As analysis of Proposition 1, the proposed pASK can use Hoeny_list to prevent online guessing attack. Therefore, the proposed pKAS scheme can be secure against online password guessing attack. □

Proposition 3. *The proposed pKAS scheme can provide anonymous interactions among the users U_i and edge cloud sever U_j, and no adversary $\mathscr{A}$ can obtain both identity information during login and key agreement phase.*

Proof. In the login and key agreement phase of pKAS, user U_i's real identity ID_i is hidden in message $\mathrm{PID}_i = \mathrm{ID}_i \oplus h(c_i X_j \| t_i)$. If an adversary $\mathscr{A}$ can reveal the ID_i from the messages, he/she should solve the ECDLP problem because PID_i include ECDLP in their construction. Therefore, the proposed pKAS can provide anonymous interactions during user login and key agreement. □

Proposition 4. *The proposed pKAS scheme can provide forward secrecy during the session key agreement.*

Proof. Assume an adversary $\mathscr{A}$ has obtained the smart card and user's password and identity. However, $\mathscr{A}$ cannot retrieve the previously existing session key without knowing c_i because $\mathscr{A}$ should solve the ECDLP problem. Hence, the proposed pKAS scheme can give strong forward secrecy. □

Proposition 5. *The proposed pKAS scheme can be secure against forgery attack.*

Proof. In the proposed scheme, U_j can check that message M_1 has been forgery by computing $\mathrm{ID}^* = \mathrm{PID}^* \oplus h(x_j C_i \| t_i)$, $f_i' = h(\mathrm{ID}^* \| \mathrm{ID}_j \| \mathrm{PID}_i \| C_i \| t_i)$, and checking $C_i = \sigma_i P - f_i' X_i$ holds or not. U_i authenticates U_j by computing $C_j' = \sigma_j P - f_j X_j$, $\mathrm{sk}_{ji} = h(\mathrm{ID}_i \| \mathrm{ID}_j \| t_i \| t_j \| c_i C_j)$ and checking $f_j = h(\mathrm{ID}_i \| C_i \| C_j \| \mathrm{sk}_{ji} \| t_i \| t_j \| \mathrm{ID}_j)$ holds or not. When $\mathscr{A}$ modifies the message during the conversation, the tampered message cannot be verified. As a consequence, the proposed pKAS scheme can be secure against forgery attack. □

Proposition 6. *The proposed pKAS scheme can provide mutual authentication.*

Proof. In the presented scheme, U_j and U_i verify message M_1 and M_2 by checking equation $C_i = \sigma_i P - f_i' X_i$, $f_j = h(\mathrm{ID}_i \| C_i \| C_j \| \mathrm{sk}_{ji} \| t_i \| t_j \| \mathrm{ID}_j)$ hold or not, respectively. If it holds, the scheme achieves mutual authentication based on Proposition 5 that no adversary can successfully implement a forgery attack. Therefore, the presented pKAS scheme can give mutual authentication. □

Proposition 7. *The proposed pKAS can be secure against replay attack.*

Proof. In the proposed pKAS scheme, we use timestamps and random numbers to prevent replay attack. Messages M_1 and M_2 include timestamps t_i and t_j, respectively, which is a classic way to stop replay attacks. Random numbers are also used to prevent relay attack because users and server can check the validity of random number by verification algorithm each time and adversary $\mathcal{A}$ still cannot construct valid session key. Hence, the presented pKAS can be secure against replay attack. □

Proposition 8. *The proposed pKAS can be secure against impersonation attack.*

Proof. Let $\mathcal{A}$ can get U_i's smart card and know the data in the card by some way. However, $\mathcal{A}$ has to possess PW_i and ID_i into smart card to generate a legal message $M_1 = \{C_i, \mathrm{PID}_i, \sigma_i, t_i\}$. Without the two factors (PW_i and ID_i), $\mathcal{A}$ cannot compute a correct a_i to pass the verification of smart card that $\mathcal{A}$ cannot proceed to the next step to impersonate U_i to communicate with other. Therefore, the proposed pKAS can security resist impersonation attack. □

Proposition 9. *The proposed pKAS can be secure against parallel attack.*

Proof. Parallel attack usually occurs when an adversary $\mathcal{A}$ constructs a new conversation to impersonate a legal user by reusing historical messages that he/she intercepted in a public channel. However, $\mathcal{A}$ should know the parameters of messages or he/she cannot send a correct access request and gain a session key. However, $\mathcal{A}$ cannot obtain the random number that is chosen by users. As a result, the proposed pKAS can be secure against parallel attack. □

Proposition 10. *The proposed pKAS can be secure against insider attack.*

Proof. As shown in the user registration phase, user U_i send $\{\mathrm{ID}_i, X_i\}$ to U_j, where $X_i = x_i P$. Without knowing x_i, the server cannot impersonate U_i. Therefore, the proposed pKAS can be secure against insider attack. □

Proposition 11. *The proposed pKAS scheme can achieve user untraceability.*

Proof. In the proposed scheme, user U_i's real identity ID_i real identity ID_j are hidden in message $\mathrm{PID}_i = \mathrm{ID}_i \oplus h(c_i X_j \| t_i)$. Only when an adversary $\mathcal{A}$ can solve the ECDLP problem, $\mathcal{A}$ can reveal ID_i from the messages that are included by ECDLP in their construction. As a consequence, the proposed pKAS can achieve user untraceability. □

Proposition 12. *The proposed pKAS scheme can achieve key agreement.*

Proof. U_j computes his/her session key as $\mathrm{sk}_{ji} = h(\mathrm{ID}_i \| \mathrm{ID}_j \| t_i \| t_j \| C_i c_j)$, in the step A3. U_i computes his/her session key as $\mathrm{sk}_{ij} = h(\mathrm{ID}_i \| \mathrm{ID}_j \| t_i \| t_j \| c_i C_j)$, in step A4. Because $c_j C_i = c_i C_j = c_i c_j P$, U_i and U_j can compute an identical session key $\mathrm{sk}_{ji} = \mathrm{sk}_{ij}$. Therefore, the proposed pKAS scheme can achieve key agreement. □

Proposition 13. *The proposed pKAS scheme can achieve offline password change.*

Proof. As shown in introduction of the proposed scheme, offline password change phase is provided. Each user can achieve password change locally. If user inputs correct ID and PW, the correctness of ID_i and PW_i is $\kappa - 1/\kappa \approx 99.61/100$, $\kappa = 2^8$, i.e., user has a high probability of completing password local change. As a consequence, the proposed pKAS scheme can achieve offline password change. □

6.2. Performance Analysis. In this section, we compare our scheme with similar schemes in terms of security performance, communication consumption, and computing consumption. The results indicate that pKAS is more secure and effective than other similar schemes. In addition, the presented pKAS has lower communication and computation costs.

6.2.1. Comparison of Security Features. We define $F1$, $F2$, $F3$, $F4$, $F5$, $F6$, $F7$, $F8$, $F9$, $F10$, $F11$, and $F12$ are the functionality of "be secure against off-line password guessing attack," "be secure against online password guessing attack," "provide anonymous interactions," "provide forward secrecy," "be secure against forgery attack," "provide mutual authentication," "be secure against replay attack," "be secure against impersonation attack," "be secure against parallel attack," "be secure against insider attack," "achieve user untraceability," "achieve key agreement," and "achieve offline password change," respectively. In Table 2, we compare the security features of pKAS with related scheme, such as Irshad et al. [33], Tsai and Lo [35], and Kaur et al. [36].

6.2.2. Comparison of the Computation Cost. It is more convenient to define T_{BP}, T_{ME}, T_{PM}, T_{PA}, and T_{HO} are the running time (in ms) of a single bilinear pairing operation, modular exponentiation operation, elliptic curve point multiplication, point addition, and hash operation, respectively. In Table 3, we list the computing time of the server and the mobile terminal separately. The cost in Table 3 is based on [36]. We use simulation Alibaba's cloud server, and its configuration is Intel(R) Xeon(R) CPU E5-26300@ 2.30 GHz, 1 GB RAM and Ubuntu 14.04. In addition, the smartphone we use is configured with 2 GHz ARM CPU armeabi-v7a, 300 MiB RAM and Android 4.4 to simulate the mobile terminal.

Table 2: Security features comparison.

Schemes	$F1$	$F2$	$F3$	$F4$	$F5$	$F6$	$F7$	$F8$	$F9$	$F10$	$F11$	$F12$
Irshad et al. [33]	∨	×	∨	∨	×	∨	∨	×	×	×	∨	∨
Tsai and Lo [35]	×	×	∨	×	×	∨	∨	∨	∨	×	∨	∨
Kaur et al. [36]	×	×	∨	∨	×	∨	∨	∨	∨	×	∨	∨
pKAS	∨	∨	∨	∨	∨	∨	∨	∨	∨	∨	∨	∨

Note: ∨ means available; × means not available.

Table 3: Comparison of the computation cost on different devices.

Device	T_{BP}	T_{PM}	T_{PA}	T_{HO}	T_{ME}
Server	5.275	1.97	0.012	0.009	0.339
Client	48.99	19.919	0.118	0.089	3.328

Table 4: Comparison of the computation cost.

Schemes	User	Server
Irshad et al. [33]	$T_{PB} + 5T_{PM} + 2T_{PA} + 2T_{ME} + 6T_{HO} = 155.68$	$2T_{BP} + 4T_{PM} + 3T_{PA} + 2T_{ME} + 3T_{HP} = 19.171$
Tsai and Lo [35]	$5T_{PB} + 2T_{PA} + T_{ME} + 5T_{HO} = 247.309$	$2T_{BP} + 2T_{PM} + 2T_{PA} + 2T_{ME} + 5T_{HO} = 15.228$
Kaur et al. [36]	$4T_{PM} + 4T_{HO} = 80.032$	$3T_{PM} + 4T_{HO} = 5.946$
pKAS	$5P_{PM} + T_{PA} + 4T_{HO} = 100.069$	$5T_{PM} + T_{PA} + 4T_{HO} = 9.898$

Table 5: Comparison of the communication cost.

Schemes	Number of messages	Communication cost (bits)
Irshad et al. [33]	3	3072
Tsai and Lo [35]	3	3072
Kaur et al. [36]	3	1920
pKAS	2	1472

According to the time computation by each operation in Table 3, we compared the time in [33, 35, 36], and pKAS schemes, as shown in Table 4.

6.2.3. Comparison of the Communication Cost. The comparison results in Table 4 are based on assumptions such as result of hash function to be 160 bits, random number to be 128 bits, identifier to be 64 bits, time stamp to be 32 bits, and encryption/decryption and ECC point to be 320 bits. Table 5 shows a comparison of the communication cost between pKAS and other schemes [33, 35]

In summary, the presented pKAS which consumes lower communication and calculations than [33, 35]. Though the cost of [36] is lower than pKAS, the scheme cannot be secure against forgery attacks and insider attack, and its bandwidth consumption is relatively large. Furthermore, pKAS is more secure than [33, 35, 36]. So, pKAS is more suitable for user and server to verify each other.

7. Conclusion and Future Work

Aiming at the practical problems encountered in the key agreement between the user and server in the edge cloud computing environment, we propose a new password-based key agreement scheme. We use ECDLP to construct user anonymity and forward secrecy. By comparing security, communication, and calculation costs, the proposed pKAS has better security and lower cost. Furthermore, pKSA also meets all 12 security requirements.

Although pAKS is more secure and efficient than similar schemes, the lightweight key agreement scheme, such as no point multiply operation, is more favored. It is very challenging to design a secure and lightweight scheme. This will be the direction of our next research.

Data Availability

The data supporting the results of this study can be obtained from the corresponding author.

Conflicts of Interest

P. Liu is currentlty a lecturer at the Department of Computer Technology and Application, Qinghai University, Xining. Her research interest includes network protocol and protocol security (e-mail: 247750940@qq.com). Syed Hamad Shirazi is currentlty an Assistant Professor at the Department of Information Technology, Hazara University, Baffa, Pakistan. His research interest includes image processing and image security (syedhamad@hu.edu.pk). W. Liu is currently an assistant at the Department of Computer Technology and Application, Qinghai University, Xining. Her research interest includes network protocol and protocol security (e-mail: 1007759705@qq.com). Y. Xie is currently a Professor at the Department of Computer Technology and Application, Qinghai University, Xining. His research interest includes network protocol and protocol security (e-mail: mark.y.xie@qq.com).

Acknowledgments

This study was supported in part by the Science and Technology Foundation of Qinghai under grant no. 2019-ZJ-7065, the National Natural Science Foundation of China under grant no. 61572370, and Course Construction of Qinghai University under grant no. SZ19014.

References

[1] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and M. Guizani, "Edge computing in the industrial internet of things environment: software-defined-networks-based edge-cloud interplay," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 44–51, 2018.

[2] X. Liang, X. Wan, X. Du, X. Chen, G. Mohsen, and C. Dai, "Security in mobile edge caching with reinforcement learning," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 116–122, 2018.

[3] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He, and X. Zhou, "TripRes," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 17, no. 2, pp. 1–21, 2021.

[4] X. Xu, Q. Wu, L. Qi, W. Dou, S. B. Tsai, and M. Z. A. Bhuiyan, "Trust-aware service offloading for video surveillance in edge computing enabled internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1787–1796, 2020.

[5] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) ≪ cost (signature) + cost (encryption)," in *Proceedings of the Annual International Cryptology Conference*, pp. 165–179, Springer, Santa Barbara, California, USA, August 1997.

[6] E. J. Yoon, S. B. Choi, and K. Y. Yoo, "A secure and efficiency id-based authenticated key agreement scheme based on elliptic curve cryptosystem for mobile devices," *International journal of innovative computing, information & control: IJICIC*, vol. 8, no. 4, pp. 2637–2653, 2012.

[7] D. Mishra, A. K. Das, and S. Mukhopadhyay, "A secure and efficient ecc-based user anonymity-preserving session initiation authentication protocol using smart card," *Peer-to-Peer Networking and Applications*, vol. 9, no. 1, pp. 171–192, 2016.

[8] M. F. Sabzinejad and M. A. Ahmadian, "An id-based key agreement protocol based on ecc among users of separate networks," in *Proceedings of the ISCISC International ISC Conference on Information Security and Cryptology*, Tabriz, Iran, September 2012.

[9] S. H. Islam and G. P. Biswas, "A more efficient and secure id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1892–1898, 2011.

[10] X. Jia, D. He, N. Kumar, and K. Choo, "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing," *IEEE Systems Journal*, vol. 14, no. 1, pp. 1–12, 2019.

[11] M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *Proceedings of the International Workshop on Public Key Cryptography*, pp. 65–84, Springer, Les Diablerets, Switzerland, January 2005.

[12] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 139–155, Springer, Kyoto, Japan, December 2000.

[13] E. J. Yoon and K. Y. Yoo, "Robust id-based remote mutual authentication with key agreement scheme for mobile devices on ecc," in *Proceedings of the 2009 International Conference on Computational Science and Engineering*, vol. 2, pp. 633–640, IEEE, Vancouver, BC, Canada, August 2009.

[14] L. Harn, W.-J. Hsin, and M. Mehta, "Authenticated diffie-hellman key agreement protocol using a single cryptographic assumption," *IEE Proceedings - Communications*, vol. 152, no. 4, pp. 404–410, 2005.

[15] Y.-M. Tseng, "Efficient authenticated key agreement protocols resistant to a denial-of-service attack," *International Journal of Network Management*, vol. 15, no. 3, pp. 193–202, 2005.

[16] E. J. Yoon and K. Y. Yoo, "New efficient simple authenticated key agreement protocol," in *Proceedings of the Computing and Combinatorics, 11th Annual International Conference, COCOON*, Kunming, China, August 2005.

[17] M. Geng and F. Zhang, "Provably Secure Certificateless Two-Party Authenticated Key Agreement Protocol Without Pairing," in *Proceedings of the International Conference on Computational Intelligence & Security*, León, Spain, November 2010.

[18] M. Hou and Q. Xu, "A two-party certificateless authenticated key agreement protocol without pairing," in *Proceedings of the The 2nd IEEE International Conference on Computer Science and Information Technology*, pp. 412–416, Beijing, China, August 2009.

[19] J. H. Yang and C. C. Chang, "An id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Computers & Security*, vol. 28, no. 3-4, pp. 138–143, 2009.

[20] H. Hou and S. Liu, "Cpk-based authentication and key agreement protocols with anonymity for wireless network," in *Proceedings of the International Conference on Multimedia Information Networking & Security*, Jeju Island, Korea, December 2009.

[21] A. Weimerskirch, S. Douglas, and S. C. Shantz, "Generic gf(2m) arithmetic in software and its application to ecc," in *Proceedings of the Information Security and Privacy, 8th Australasian Conference, ACISP 2003*, Wollongong, Australia, July 2003.

[22] S. U. Nimbhorkar and L. G. Malik, "Exploration of schemes for authenticated key agreement protocol based on elliptic curve cryptosystem," in *Proceedings of the 2013 6th International Conference on Emerging Trends in Engineering and Technology (ICETET)*, Nagpur, India, December 2013.

[23] S. M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," in *Proceedings of the IEEE Symposium on Security & Privacy*, Oakland, CA, USA, May 1992.

[24] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, 1981.

[25] J. Xu, W. T. Zhu, and D. G. Feng, "An improved smart card based password authentication scheme with provable security," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 723–728, 2009.

[26] S. K. Sood, A. K. Sarje, and K. Singh, "An improvement of xu et al.'s authentication scheme using smart cards," in *Proceedings of the ACM bangalore annual conferenceCOMPUTE 2010*, Banglore, India, January 2011.

[27] B. L. Chen, W. C. Kuo, and L. C. Wuu, "Robust smart-card-based remote user password authentication scheme,"

International Journal of Communication Systems, vol. 27, no. 2, 2014.

[28] X. Li, J. Niu, M. K. Khurram, and J. Liao, "An enhanced smart card based remote user password authentication scheme," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1365–1371, 2013.

[29] W. B. Hsieh and J. S. Leu, "Exploiting hash functions to intensify the remote user authentication scheme," *Computers & Security*, vol. 31, no. 6, pp. 791–798, 2012.

[30] H. B. Tang, X. S. Liu, and L. Jiang, "A robust and efficient timestamp-based remote user authentication scheme with smart card lost attack resistance," *International Journal on Network Security*, vol. 15, no. 6, pp. 446–454, 2013.

[31] A. K. Awasthi, K. Srivastava, and R. C. Mittal, "An improved timestamp-based remote user authentication scheme," *Computers & Electrical Engineering*, vol. 37, no. 6, pp. 869–874, 2011.

[32] E. J. Yoon, K. Y. Yoo, and K. S. Ha, "A user friendly authentication scheme with anonymity for wireless communications," *Computers & Electrical Engineering*, vol. 37, no. 3, pp. 356–364, 2011.

[33] A. Irshad, M. Sher, H. F. Ahmad, B. A. Alzahrani, and R. Kumar, "An improved multi-server authentication scheme for distributed mobile cloud computing services," *Ksii Transactions on Internet & Information Systems*, vol. 10, no. 12, pp. 5529–5552, 2016.

[34] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Generation Computer Systems*, vol. 81, pp. 557–565, 2018.

[35] J. L. Tsai and N. W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, vol. 9, no. 3, pp. 805–815, 2017.

[36] K. Kaur, S. Garg, G. Kaddoum, M. Guizani, and D. Jayakody, "A lightweight and privacy-preserving authentication protocol for mobile edge computing," in *Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM)*, IEEE, Waikoloa, HI, USA, December 2020.

[37] L. Xiong, D. Peng, T. Peng, and H. Liang, "An enhanced privacy-aware authentication scheme for distributed mobile cloud computing services," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 12, pp. 6169–6187, 2017.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 1631496, 8 pages
https://doi.org/10.1155/2021/1631496

Research Article

Reliability of Hijacked Journal Detection Based on Scientometrics, Altmetric Tools, and Web Informatics: A Case Report Using Google Scholar, Web of Science, and Scopus

Mohammad R. Khosravi[1] **and Varun G. Menon**[2]

[1]*Department of Electrical and Electronic Engineering, Shiraz University of Technology, Shiraz, Iran*
[2]*Department of Computer Science and Engineering, SCMS School of Engineering and Technology, Kerala 683582, India*

Correspondence should be addressed to Mohammad R. Khosravi; m.khosravi@sutech.ac.ir

Received 10 May 2021; Revised 25 August 2021; Accepted 17 September 2021; Published 12 October 2021

Academic Editor: Hao Wang

This paper presents a case report on detecting hijacked journals. Towards identification of a fake journal website and preventing a hijacked paper, we can use different tools including Google Scholar and Web of Science (WoS) and Scopus (both as scientometric databases) to distinguish a fake website from a legal journal website. Our evaluation shows that analysis of a doubtful website for a targeted journal based on Google Scholar is not reliable. In fact, the use of scientometric tools for tracking prior publications of the targeted journal is compulsory. Another result of this study is that in some uncommon cases, fake websites (clone versions) may sometimes convince a scientometric database in order to be fully/partially indexed along with an abstracting of their hijacked papers while these websites steal identity of the legal journals. Therefore, as a result, we should check both of WoS and Scopus at the same time for verifying a fake website to obtain more reliability.

1. Introduction

Predatory journals and publishers have been a major threat to academic research community for many years [1–3]. The major challenge for researchers is always to identify and eliminate these predatory journals. Since the last few years, a similar type of fake publishers with many hijacked journals has attacked the research community extensively and intensively. Fake journals are indeed versions with hijacked identity for legitimate academic journals such that some duplicate/fake websites are created by malicious third party or criminals. Probably, a different type of *Phishing attack* is done by these fake websites to find some academicians as victim. These fake websites may copy all the contents available in the website of the legitimate journal such as Impact Factor (IF), ISSN, Editorial Board Members (EBMs) information, and indexing and archiving information. They then send attractive calls for papers to researchers around the world inviting them to publish with these fake journals for a high fee (but normally these fees are not much high compared with fees of legal open access journals in that area). These attackers particularly target the researchers in desperate need for publications. With the concept of *Publish or* Perish existing in many countries, many academicians fall in this trap. Authors receiving these e-mails are attracted by the indexing information such as Scopus, Web of Science (WoS), or the IF value of journal. They then click on the link of the hijacked version of the journal given in the e-mail and proceed with a submission and a short time later with payment of the authors' fees and finally publication of their papers. Thus, money, time, and the research work would be lost (maybe!) forever through these fake websites [4–6].

Currently, the academic community is in a dilemma and unable to efficiently resolve this problem. Many right persons have started to display the list of fake journals in their own websites to help potential authors; however, having a general guideline to reveal any new case is more of interest which is tried to be handled in this paper. The objective of us

here is to show to the research community how duplicate/fake journals can be identified using Google Scholar, Web of Science, and Scopus through a case study. We believe that the idea behind this paper would become a valuable reference for all the researchers [7–9].

The rest of this paper is organized as follows. First, the case study will be discussed, and then we try to verify its identity through different tools. At the end, a conclusion on the work will be given. This paper is the final publication for the preprint version published by *TechRxiv* [10].

2. Case Study

In this study, we aim to evaluate a major databases indexed journal entitled *Journal of Engineering Technology* (JET). This journal has some features which make it suitable for hijacking; see Table 1 for more details. According to Scopus/ScimagoJR in 2019, "JET is a refereed journal published semiannually, in spring and fall, by the Engineering Technology Division (ETD) of the American Society for Engineering Education (ASEE) and is indexed by the Engineering Index (EI) Compendex and the Science Citation Index (SCI). The journal was first published in 1984 and has since become one of the major publication venues of refereed scholarly works for engineering technology educators. The purpose of the *Journal of Engineering Technology* is spelled out in the JET Editorial Policy document." In Figures 1 and 2, some papers published in a fake website for this journal are observable.

Although our paper is about a case study, e.g., JET, the final solution presented by us is completely general. The case study in our research is a very unique case in the first step of the study, done in 2019 (see the Acknowledgments section), so that we can show some shortcoming of the hijacked platforms detection process through it whereas no other case could reveal the lacks.

3. Google Scholar Results

In this section, result of searching the fake website of JET through Google Scholar search engine is discussed. As seen in Figure 4, Google Scholar as an altmetric tool for promoting scientific work is not a reliable way to verify originality of a journal website. These works are based on an artificial intelligence-assisted Google robot to extract scientific information and then to make abstracting of them.

4. Scopus Results

Here, we want to detail how to use Scopus database for identifying fake websites. Checking articles claimed as published documents of doubtful websites in Scopus and Web of Science (WoS) is a very reliable way to take a decision about authenticity of a journal website. If there are many recent papers (do not select online first/ahead of print/early cite papers and also papers from the last published issue because indexing may be time-consuming) which have not been abstracted in the claimed indexing databases, the website is then fake. For example, we could find the doubtful website claiming the *Journal of Engineering Technology* with "ISSN: 0747–9964" (as per Table 1 and Figure 5, this ISSN belongs to a journal indexed by both WoS and Scopus), and then we are going to verify it using Scopus.

Figure 1 shows some papers published in this doubtful website (http://www.joetsite.com). Two papers were selected which are observable in two color boxes (orange and blue boxes). In this step, we do analysis on the paper of Figure 2. As seen in Figure 6, its title has been searched through Scopus search engine, and the search result is according to Figure 7. The result is "No documents were found." If we find many such cases in this website, therefore, we can surely say that this website is fake and there is a fraud here. In this specific case, it is fake because there many unavailable papers on Scopus.

In addition to the above case, some published papers of the original journal may be published in a fake website, so in the cases that some papers of a doubtful website are searchable through Scopus or WoS and some are not, you should carefully check its publishing information and resolve its Digital Object Identifier (DOI), if applicable (having a valid DOI related to a doubtful website is not important, so the main point is to resolve the mentioned DOI in Scopus or WoS towards that doubtful website). In a very infrequent case in 2017, we observed that the fake website could convince Scopus in order to indexing/abstracting of papers as the original source on which Scopus did abstracting for papers published in this website. A wonderful point in this experience was to see some papers of both original journal and hijacked version concurrently whereas they have different publishing information (volume, issue, and so on) but under a unique ISSN. As follows, we will detail the observation. Thus, as a result, we think that authors should check both WoS and Scopus, not just one of them (if applicable). In our case study on the journal described in Table 1, although Scopus has removed most of papers received from the fake website (in Nov. 2018 as starting point of our study, we could not find 2017 papers again on Scopus), we could still find a paper of this website published in 2018 among many papers from the original journal. Figure 3 indicates three papers for the journal with "ISSN: 0747–9964" whereas one of them is for the fake website and has some completely different publishing information. This paper is observable in blue box of Figure 1.

5. Web of Science Results

In this section, verification is performed by WoS. Fortunately, WoS in this specific case is clear and does not cover any document of the fake website (this last sentence does not mean that we believe WoS is prefered than Scopus! We only wish to say "check both for more reliability"). Its sample results are shown in Figure 8 without coverage of the fake website. In addition, statistics provided by WoS would clearly demonstrate a fact against the clone version (see Figures 9 and 10) interpreted as follows:

Table 1: A hijacked journal for the case study (data were collected in Dec. 2018).

Journal name	Indexing	ISSN	Original website	Fake website
Journal of Engineering Technology*	WoS (SCIE/JCR) Scopus EBSCO	0747–9964	N/A**,***	http://www.joetsite.com

* This journal mainly publishes extended versions of some conference papers presented at conferences of American Society for Engineering Education (ASEE). ** We could not find it in 2019; however, it seems that the website does not exist. Only some of titles published by real journal can be found as conference versions on the ASEE website; for example, see the case ordered as 7th in Figure 3 (by Sriraman et al., 2017) via this link https://www.asee.org/public/conferences/78/papers/17663/view#).*** Based on the Scopus record through ScimagoJR, the main website of this journal is https://www.engtech.org/jet; however, we could not approve it in 2019 through its contents.

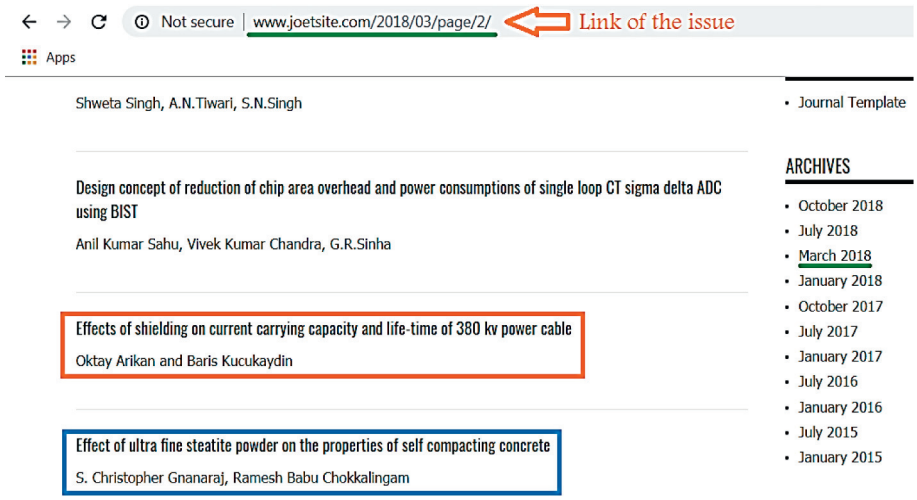


Figure 1: Fake website for the Journal of Engineering Technology (http://www.joetsite.com). This figure shows two evaluated articles in Volume 6, Issued in March (Special Issue), 2018.

Figure 2: A paper published by the fake website of JET (http://www.joetsite.com).

(i) Finding from Figure 9: it is obvious that the number of published papers is about 10 yearly according to WoS, and this number is much less than the records observed in the clone version of the journal (when it was alive in 2018–2020, later this sentence will be explained).

(ii) Finding from Figure 10: the clone version was an open access jounal, but Figure 10 shows a citation record as much less than a record for an open access venue considering the number of publications in Figure 9.

Since WoS has important features compared with the other two tools (Table 2), the checking with it as the final step is essential and highly recommended to be done.

6. General Findings and Discussion

Thus, researchers can avoid fake journals and can publish their research papers in legitimate journals with confidence [11–15]. Furthermore, we aim to come up with a comprehensive list of hijacked journals and a simple tool that can be used by researchers to detect these fake journals [13, 16, 17]. In the time of publishing this current version of our research, the clone version addressed in the paper is no longer available, maybe due to our first report about in 2019 [13] (also, see the Acknowledgments section) and repeating the same report in [16] based on our finding. However, this availability is not important because the same things will/may appear in future such that some of them have been reported in [16] just now. In total,

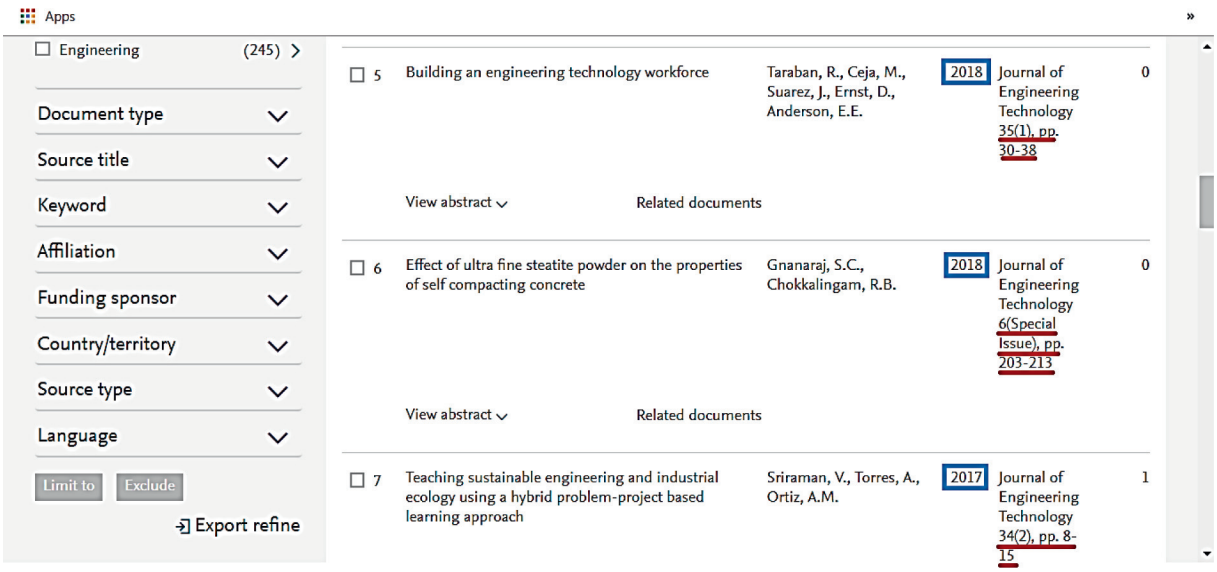


Figure 3: Three articles found for "ISSN: 0747–9964" on Scopus, data acquired in Nov 2018 (source: Scopus).

Figure 4: Google Scholar supports the fake website of JET (source: Google Scholar).

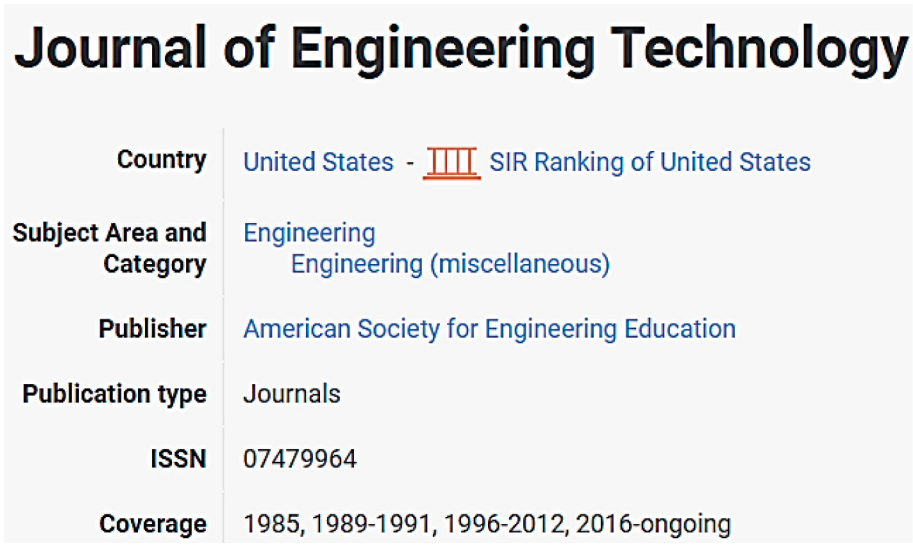


Figure 5: JET details provided by ScimagoJR, a product of Scopus (source: Scimago JR in 2019).

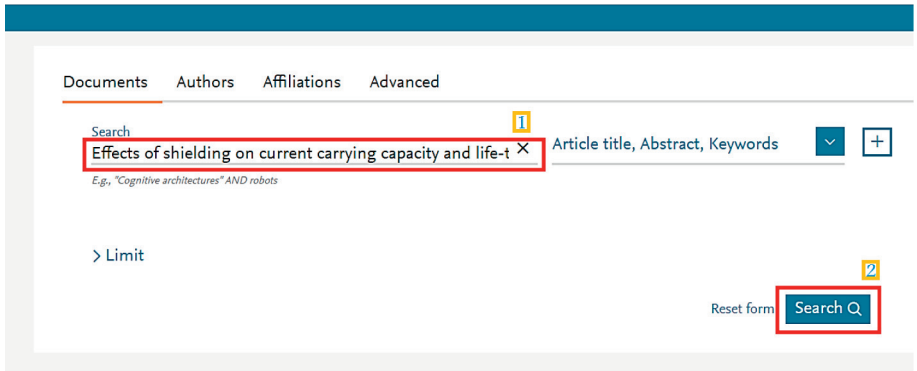


Figure 6: Searching article determined by orange color box in Figure 1 using Scopus search engine (source: Scopus).

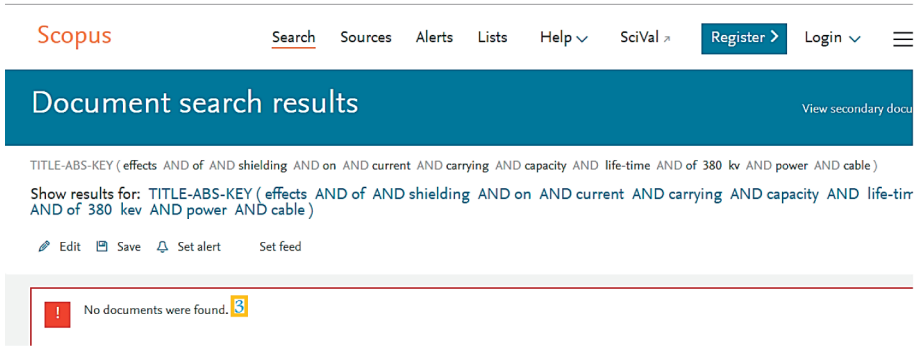


Figure 7: Result of the search engine for the case searched in Figure 6 (source: Scopus).

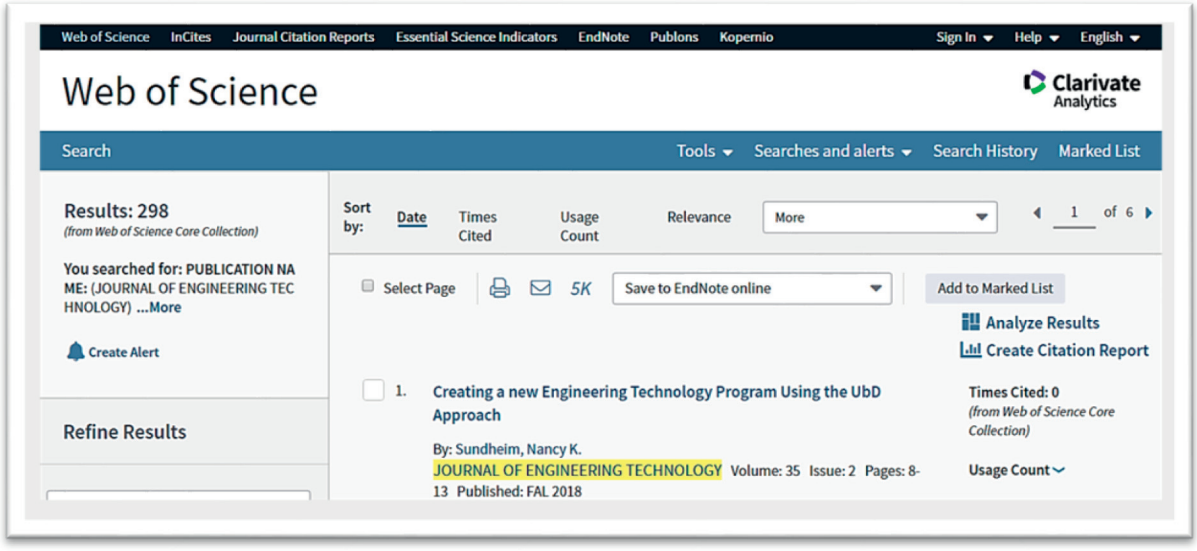


Figure 8: WoS search for JET does not include papers of the clone version, data acquired in Nov 2018 (source: Web of Science).

it is a critical point that making a safer data environment for cloud-based big data services must be taken more seriously [18, 19], especially for the science. We hope this research with its historical view on a case study during over four years evaluation (2017–2021) can help the scientists find a secure way of publishing their academic and industrial findings. Security and privacy not only must be provided with computational methods but also in the modern form of data exchange; it is covered by intelligent and policy-based solutions [20, 21], similar to the approach of the current paper. As brief, our contributions are as follows:

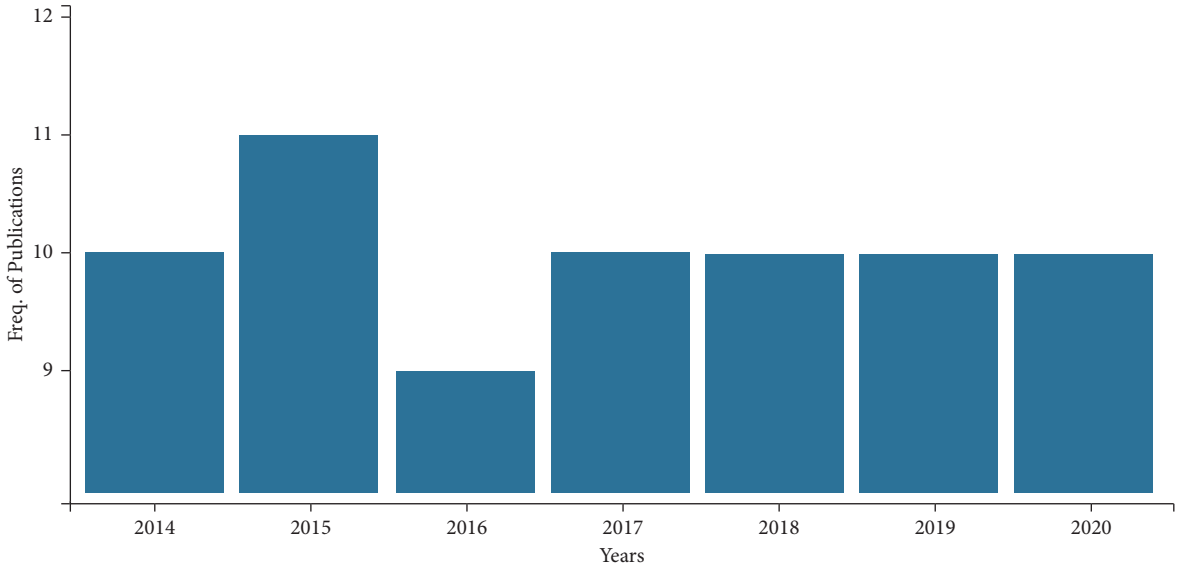


Figure 9: Number of publications per year for the main journal according to WoS, data acquired in June 2021; JET has had a continuous coverage by WoS since 1998 (source: Web of Science).

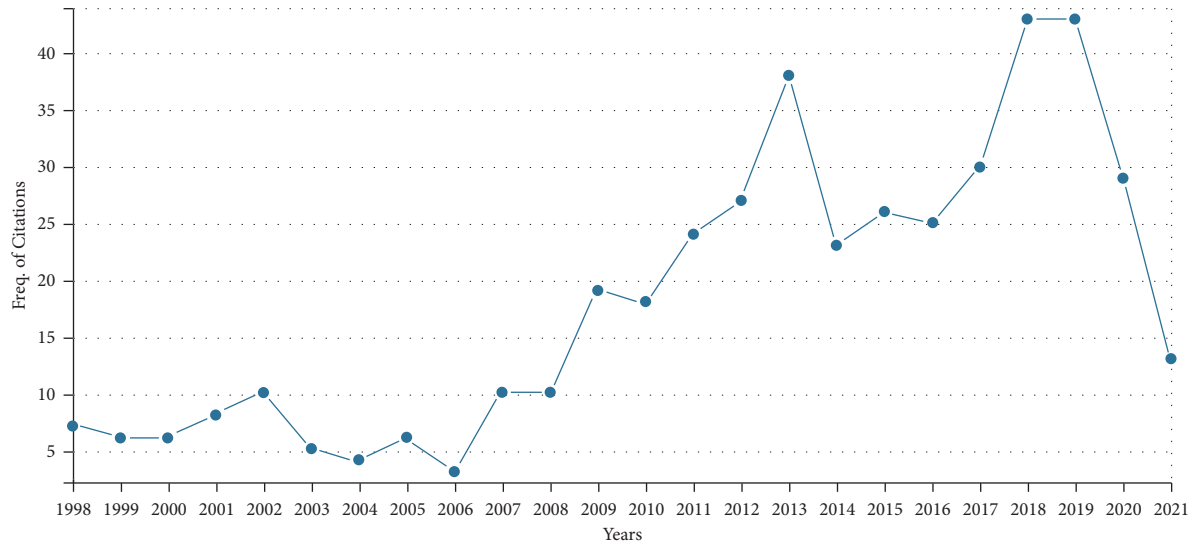


Figure 10: Citation rate (times per year) of the main journal in WoS, data acquired in June 2021 (source: Web of Science).

Table 2: Comparison of the three scientific tools.

Service	Type	Operation	Reliability
Google Scholar	Altmetric	Robot-based processing	Very low
Scopus	Scientometric	Evaluation departments, publishers	High
WoS	Scientometric	Evaluation departments, publishers	Very high

(i) This research is one of the first investigations around a clone version of an indexed journal with the ability of abstracting in one of the major indexing services, i.e., Scopus. As mentioned, the author of [16] could later list several similar cases including the case study of our work. Such clone versions are named *advanced clones* because of their success in convincing the major indexing services.

(ii) We could suggest several solutions to avoid publishing in the advanced clones which are mainly a fake website for WoS/Scopus-indexed journals as follows:

Table 3: The history behind JET.

Year	Description	Related documents
2017 to Mid-2018	Detecting the clone version with several abstracted papers on Scopus	N/A
Mid-2018 to 2019	First report about the clone while one sample from the clone website could still be found on Scopus	Reference [13]; doi: 10.1108/LHTN-11-2018–0070 [10]; doi: 10.36227/techrxiv.11385849.v1
2020	Further checking and updating the data, no case was found on Scopus for the clone	[10]; doi: 10.36227/techrxiv.11385849.v2
2021	Integrating all reports and reaching the final output of the case study while the clone website had been suspended, and some new evidence about the legitimate journal is accessible	The current paper published by *SCN*

Paying attention to the volume/issue number of the published papers of any doubtful website through Scopus/WoS and comparing with other abstracted papers to find any inconsistency. The experience says that the inconsistency is mostly found in Scopus, but WoS must be also double-checked, specifically for explaning any found difference.

The number of citations in Scopus/WoS is very important for the corresponding ISSN of the main journal (legitimate version)/fake website (clone version) to be monitored. Since clones are mainly open access and the main journal is not (whether having a website or not to have), thus we should expect a citation record similar to open access journals. When a clone version cannot do the abstracting of most/all of its publications, therefore, no citation will be found for those publications; for example, the JET citations are not considerable in WoS.

The number of abstracted papers in WoS/Scopus should be checked and compared with the number of papers in a clone version (under-doubt website).

Inconsistency in parallel WoS/Scopus indexing (= abstracting of published content here) of an under-doubt website claims both major indexes at the same time.

(iii) In other cases, similar checklists can be redesigned per case according to all claims and demands.

(iv) In the information retrieval about the case study in 2020/2021, some new happenings are seen as follows:

The clone version of JET has been suspended from service. The last time of access to the clone was in 2020 from our side.

Scopus has removed all publications of the clone version of JET.

ScimajoJR as a linked product of Scopus has introduced the legitimate website of JET (seems to be a newly launched homepage) and an e-mail address for JET. This action of Scopus and launching the formal web page of JET can be also strong reasons of unavailability of the clone version in addition to our first report in [13] and the link provided in the Acknowledgments section. The Scopus-related new information can be found in [22]. Table 3 summarizes the background of our study on JET.

7. Conclusions

Initially, we discussed the importance of eliminating predatory publishers and journals and then highlighted a similar version of predatory journals, i.e., the hijacked journals. Currently, the major challenge faced by the research community is inaccurate identification of these fake websites. Our research aimed at displaying the research community, how duplicate/fake journals can be identified using Google Scholar, Web of Science, and Scopus with a case study. We used the example of the legitimate and well established journal *Journal of Engineering Technology* and showed how fake journals exist with similar name and content. We also showed the results with Google Scholar, Scopus, and Web of Science tools and made some deep analysis based on them. This solution can be easily replicated by other researchers and can be used to identify potential fake journals in any scientific field of research. Information science security concerning journal hijacking and clones is a kind of cybercrime analysis in computer and web technologies.

Abbreviations

ISSN: International standard series number
JET: Journal of Engineering Technology.

Data Availability

All the data can be tracked through the databases, and a copy of all is also accessible through the corresponding author.

Disclosure

An initial draft of this paper was published as a preprint by TechRxiv preprint service in 2019 and has been updated in 2020. The aim behind posting this version was to collect insightful comments from the research community and monitor any updates regarding the original and clone versions. The readers can follow this preprint version and all

its updates through: https://doi.org/10.36227/techrxiv.11385849. This current document published in 2021 is the third and last update of our study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Beall, "Polish journal is hijacked," 2014, https://web.archive.org/web/20160305164252/.

[2] J. Beall, "Hijacked journal's list," 2016, https://beallslist.weebly.com/hijacked-journals.html.

[3] Cabells, "The Journal Blacklist," 2018, https://www2.cabells.com/about-blacklist.

[4] C. Analytics, "Master Journal List," 2018, http://mjl.clarivate.com/cgi-bin/jrnlst/jlresults.cgi?PC=MASTER&ISSN=0449-0576.

[5] M. Dadkhah, "Paper hijacking: hijackers are attacking journals for hijacking unpublished papers," *Journal of Digital Information Management*, vol. 13, no. 4, pp. 281-282, 2015.

[6] M. Dadkhah and G. Borchardt, "Hijacked journals: an emerging challenge for scholarly publishing," *Aesthetic Surgery Journal*, vol. 36, no. 6, pp. 739–741, 2016.

[7] M. Dadkhah, "Types of hijacking in the academic world - our experiment in the scholarly publishing," *Library Hi Tech News*, vol. 33, no. 3, pp. 1-2, 2016.

[8] M. Jalalian, "Occitan literature," *The Virgil Encyclopedia*, vol. 6, no. 4, pp. 925-926, 2014.

[9] M. Jalalian and M. Dadkhah, "The full story of 90 hijacked journals from August 2011 to June 2015," *Geographica Pannonica*, vol. 19, no. 2, pp. 73–87, 2015.

[10] M. R. Khosravi and V. G. Menon, "Reliability of hijacked journal detection based on scientometrics, altmetric tools and Web informatics: a case report using Google scholar, Web of science and Scopus," *TechRxiv Preprint Server*, vol. v1, 2019.

[11] S. Khazaei and J. Kolahi, "Journal hijacking: a new challenge for medical scientific community," *Dental Hypotheses*, vol. 6, no. 1, pp. 3–5, 2015.

[12] M. R. Khosravi, "Reliability of scholarly journal acceptance rates," *Library Hi Tech News*, vol. 35, no. 10, pp. 7-8, 2018.

[13] V. G. Menon and M. R. Khosravi, "Preventing hijacked research papers in fake (rogue) journals through social media and databases," *Library Hi Tech News*, vol. 36, no. 5, pp. 1–6, 2019.

[14] V. G. Menon, "Hijacked Journals: What They Are and How to Avoid Them: Publons (Clarivate Analytics)," 2019, https://publons.com/blog/hijacked-journals-what-they-are-and-how-to-avoid-them.

[15] V. G. Menon, ""How are predatory publishers preying on uninformed scholars? Don't Be a victim", IGI global's webinar Series," 2018, https://www.igi-global.com/symposium.

[16] A. Abalkina, "Hijacked Journals in Scopus," 2021, https://www.researchgate.net/publication/352062052.

[17] A. Abalkina, "How Hijacked Journals Keep Fooling One of the World's Leading Databases," 2021, https://retractionwatch.com/2021/05/26/how-hijacked-journals-keep-fooling-one-of-the-worlds-leading-databases.

[18] S. Goyal, S. Bhushan, Y. Kumar et al., "An optimized framework for energy-resource allocation in a cloud environment based on the whale optimization algorithm," *Sensors*, vol. 21, no. 5, p. 1583, 2021.

[19] A. W. Khan, M. U. Khan, J. A. Khan et al., "Analyzing and evaluating critical challenges and practices for software vendor organizations to secure big data on cloud computing: an ahp-based systematic approach," *IEEE Access*, vol. 9, pp. 107309–107332, 2021.

[20] X. Xu, Q. Geng, H. Cao et al., "Blockchain-powered service migration for uncertainty-aware workflows in edge computing," in *Dependability in Sensor, Cloud, and Big Data Systems and Applications. DependSys 2019*, G. Wang, M. Z. A. Bhuiyan, S. De Capitani di Vimercati, and Y. Ren, Eds., vol. 1123, pp. 217–230, Springer, Singapore, 2019.

[21] X. Chi, C. Yan, H. Wang, W. Rafique, and L. Qi, "Amplified locality-sensitive hashing-based recommender systems with privacy protection," *Concurrency and Computation: Practice and Experience*, pp. 1–21, 2020.

[22] Scopus, "The Details for Journal of Engineering Technology," 2021, https://www.scimagojr.com/journalsearch.php?q=12487&tip=sid&clean=0.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 5109163, 16 pages
https://doi.org/10.1155/2021/5109163

Research Article

Edge Server Placement for Service Offloading in Internet of Things

Rong Ma

Basic Teaching Department, Nanjing University Jinling College, Nanjing, 210089, China

Correspondence should be addressed to Rong Ma; 030239@jlxy.nju.edu.cn

Received 6 August 2021; Revised 6 September 2021; Accepted 13 September 2021; Published 30 September 2021

Academic Editor: Xuyun Zhang

With the rapid development of the Internet of Things, a large number of smart devices are being connected to the Internet while the data generated by these devices have put unprecedented pressure on existing network bandwidth and service operations. Edge computing, as a new paradigm, places servers at the edge of the network, effectively relieving bandwidth pressure and reducing delay caused by long-distance transmission. However, considering the high cost of deploying edge servers, as well as the waste of resources caused by the placement of idle servers or the degradation of service quality caused by resource conflicts, the placement strategy of edge servers has become a research hot spot. To solve this problem, an edge server placement method orienting service offloading in IoT called EPMOSO is proposed. In this method, Genetic Algorithm and Particle Swarm Optimization are combined to obtain a set of edge server placements strategies, and Simple Additive Weighting Method is utilized to determine the most balanced edge server placement, which is measured by minimum delay and energy consumption while achieving the load balance of edge servers. Multiple experiments are carried out, and results show that EPMOSO fulfills the multiobjective optimization with an acceptable convergence speed.

1. Introduction

Internet of Things (IoT) is a network that connects any object to the Internet through sensors to realize intelligent identification, tracking, and control. With the rapid development of information technology, the IoT is playing an increasingly important role in daily life [1–3]. In recent years, with the popularity of smart mobile devices and the explosion of computationally intensive mobile applications, the lack of computing resources of smart mobile devices and sensors cannot guarantee the real-time processing of these computationally intensive tasks [4, 5]. As a strategy to alleviate the pressure on computing resources, cloud computing is introduced into the IoT [6–8]. The data collected by sensors or computing-intensive tasks from smart devices are transmitted to a cloud platform with powerful storage and computing capabilities, where the computation results can be stored in the cloud for subsequent operations [9, 10].

However, considering long distance between sensors and cloud platform, the transmission delay is unacceptable in some services, like the real-time identify, track, and control [5, 11]. To reduce the transmission delay and improve the quality of service and user experience, edge computing is introduced to reduce the delay and realize real-time control with edge servers, which are closer to user devices. In detail, edge service technology assigns computing and storage capabilities to edge servers and provides edge services with lower latency and better user experience [12, 13].

In most of the existing edge computing research, researchers are more inclined to focus on the process of migrating tasks to the edge server under the premise that the edge server has been deployed. In this process, service providers tend to deploy small number of edge servers for the high price of edge servers and environmental reasons [14, 15]. However, fewer studies are focusing on the layout of edge servers on the effect of edge services while the location of edge servers has a crucial impact on latency [15]. Inefficient edge server layout will cause unacceptable latency and make poor quality of user experience. Therefore, it is necessary to design an efficient edge server layout strategy to ensure the quality of edge services [16]. Not only that, edge servers are not always deployed around the sensors, and it is also necessary to ensure that the data from relatively far away sensors can be processed in time. In addition, to ensure the

quality of edge services and the stability of the edge server system and avoid excessive load on some edge servers while other edge servers are not fully utilized, it is urgent to achieve overall load balancing of edge servers. Considering the above requirements, it is a challenge to find an edge server layout strategy that can realize real-time control and guarantee the overall edge service quality.

To solve above challenges, an edge server placement strategy is designed , which aims to reduce the delay in the task transmission and the energy consumption of the edge servers. Specially, the main contributions can be concluded as follows:

(i) We propose an edge server placement method named EPMOSO in which Genetic Algorithm and Particle Swarm Optimization are effectively combined to obtain a set of placement strategies, and most balanced edge server placement considering delay and energy consumption is determined by Simple Additive Weighting Method.

(ii) Several experiments have been carried out, and results prove that the method achieves multiobjective optimization with acceptable convergence speed.

The rest of the article is organized as follows: Section 2 describes the related work. In Section 3 and Section 4, an edge computing system model for offloading services under the IoT environment and the layout strategy of edge servers are described. Massive experiments are conducted in Section 5. In Section 6, we conclude this article.

2. Related Work

In this section, the related work is divided into two parts: (1) disadvantages of cloud computing remote services solved by edge server placement, and (2) research status of edge service placement and service offloading:

(1) A large amount of data generated by sensors and devices in the IoT needs to be processed and stored. However, the central data center cannot meet such high demand for processing and storage resources [17, 18], introducing cloud computing into the IoT, which brings new changes to the data processing and storage. Hong et al. [19] studied whether the cloud system in the IoT can provide a unified platform for the continuous processing of complex data, analyzed the requirements for the engineering design of the IoT cloud system, and discussed the main engineering principles that need to be implemented. Dinh et al. [20] proposed an interactive model that integrates location-based IoT and cloud services to process cloud computing applications. In this model, the IoT cloud system provides sensing services based on mobile users' interests and locations. In addition, in response to the problem of privacy leakage in data transmission in the IoT, Christos et al. [21] integrated the IoT cloud platform and big data and built a security wall between the cloud server and the Internet to eliminate the problem of privacy leakage. However, when dealing with delay-sensitive tasks, the long delay caused by the long physical distance between the sensor and the cloud platform is unacceptable, which promotes the generation and development of edge computing to meet the needs of these delay-sensitive tasks [22]. Edge computing deploys Edge Servers (ES) with rich computing resources and storage resources on the edge of the network to process these delay-sensitive tasks, which effectively reduces delays and the pressure of transmission, and relieve the resource pressure of sensors and equipment [23, 24]. It has become a new trend to integrate edge computing and cloud computing into the IoT to deal with many computing tasks. Hassan et al. [25] put forward the key requirements for the application of edge computing in the IoT and discussed the scenarios where edge computing can be applied. To improve the security and efficiency of the IoT cloud platform, Wang et al. [8] integrated the trust evaluation mechanism, service template, and edge computing into the IoT cloud platform framework. The framework establishes a service parameter template on the cloud platform and establishes a service parsing template in the edge platform to improve the efficiency of the framework and also improves the security of the entire system through the trust evaluation mechanism. In addition, in order to improve the flexibility of the edge physical network platform, Morabito et al. [26] studied how to introduce Lightweight Virtualization (LV) into the edge physical network platform and discussed the challenges that must be solved first to effectively utilize the advantages of LV.

(2) The introduction of edge computing has greatly reduced the delay of task transmission in the IoT. However, when the number of tasks is at a high peak, processing multiple delay-sensitive tasks in the same edge server will cause additional waiting delays that reduce the quality of edge services and bring a poor user experience. Therefore, researchers began to work on solving the problem of service offloading that migrated tasks to relatively idle edge servers. Wang et al. [27] used Dynamic Voltage Scaling (DVS) to optimize the computing speed, transmit power, and offload rate of smart mobile devices, thereby reducing the energy consumption of smart mobile devices and the time delay in task execution. Yu et al. [28] considered that when multiple mobile users offload repetitive computing tasks to the edge of the network, they developed a cache-enhanced service offloading strategy based on sharing computing results to reduce the delay in task processing. As the subject of machine learning becomes more and more popular, researchers have also begun to study how to apply machine learning to edge IoT systems [29–31]. Liu et al. [28] grouped users according to priority. For grouped users of different priorities, Markov decision was adopted to design corresponding service

offloading strategies to reduce system costs, and the deep Q-network was used to train the model and determine the best service offloading strategy. Sangaiah et al. [32] proposed a method of using machine learning to protect the confidentiality of the user's location in response to the privacy leakage problem in the process of service offloading, which improves the security of users when using edge services.

The current research on edge server layout strategy based on the determined service offloading strategy are relatively small, but it is necessary to ensure the performance of edge services and the quality of real-time control by studying the layout of edge servers [15]. Therefore, this article proposes an edge server layout method for offloading services in the IoT, which aims to reduce the delay in the task transmission and the energy consumption of the edge servers and realize the overall load balance of the edge servers.

3. Edge Computing System Model Orienting Service Offloading

This section proposes an edge computing system model orienting service offloading and gives the corresponding computing formulas for the three optimization goals of delay, energy consumption, and load balancing in detail. The symbols and descriptions of this article are shown in Table 1.

3.1. Model Design. The edge computing model orienting services offloading in the IoT is shown in Figure 1.

In Figure 1, the edge computing model is divided into three levels: user layer, edge service layer, and cloud service layer. At the user level, smart devices transfer computationally intensive tasks to sensors. At the edge service layer, sensors $S = \{s_1, s_2, \ldots, s_N\}$ are deployed to collect the computationally intensive tasks of users in the coverage area. A small number of edge server $\mathrm{ES} = \{es_1, es_2, \ldots, es_M\}$ are deployed near some sensors to process the tasks collected by the sensors and perform corresponding operations based on the computing results to provide edge services for the devices. At the cloud service layer, the cloud platform handles tasks that require cloud services.

In this article, the delay, energy consumption, and load variance of different edge server layouts are used to judge the advantages and disadvantages of the edge server layout strategy. To facilitate comparison, for different edge server layout strategies, the sensor task transmission path determination rules are the same. After determining the mission transmission path of all sensors, three indicators of time delay, energy consumption, and load variance are computed. In addition, this article assumes that the CPU of the edge server is a single-core processor, and the configuration of each edge server is the same. Therefore, when there are tasks in the edge server being processed, the unprocessed sensor tasks will enter the waiting queue of the edge server waiting to be processed. At the same time, if a certain edge server has a lot of load tasks, the edge server can hand over the task that latest arrives at the edge server to an edge server that is relatively close and has less load.

Table 1: Symbols and corresponding descriptions.

Symbol	Description
ES	The collection of edge servers, $\mathrm{ES} = \{es_1, es_2, \ldots, es_M\}$
S	The collection of sensors, $S = \{s_1, s_2, \ldots, s_N\}$
Z	The number of rounds of the task in the waiting queue
ASL	The average delay of the edge server
TE	Total energy consumption
ALV	The average load variance of the edge server.
K	Number of chromosomes

3.2. Time Delay Model. The transmission and computing delays of sensor tasks include the task transmission delay of the task from the sensor to the edge server, the task processing delay of the edge server processing the task, the task waiting time of the task waiting in the edge server, and the return time of server returning the computing result to the sensor. Because the sensor can only cover a certain range, the edge server has two states: within the coverage area of the sensor and not within the coverage area of the sensor. When there is an edge server in the coverage area of the sensor, the sensor directly transmits the task to the edge server. When there is no edge server in the sensor coverage, the mission transmission path of the sensor needs to be further determined. Using a binary state variable Y_n^m to determine whether the m-th edge server es_m is within the coverage of the n-th sensor s_n,

$$Y_n^m = \begin{cases} 1, & es_m \text{ is with in the coverage of } s_n, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

then the computing formula of the time for the sensor s_n to transmit the task to the server es_m is as follows:

$$AT_n = \left(1 - Y_n^m\right) \cdot sn_n^m \cdot \frac{ds_n}{\lambda}. \tag{2}$$

where ds_n is the data volume of the sensor s_n computing task, λ is the data transmission rate of the sensor to the sensor or edge server, and sn_n^m is the number of sensors passed by the sensor s_n to the edge server es_m.

The computing formula for the time spent by the edge server to compute the task of the sensor s_n is shown in formula (3), where c_n^m is the total number of clock cycles required for processing the task of the sensor s_n in the edge server es_m, and f_m is the main frequency of the edge server es_m.

$$BT_n = \frac{c_m^n}{fm'}. \tag{3}$$

Because the task scheduling in the server adopts the First Come First Service (FCFS) algorithm, the computing formula for the waiting time of the task of the sensor s_n in the server is as follows:

$$CT_n = \sum_{z=1}^{Z_n^m} ET_z^m - AT_n, \tag{4}$$

where Z_m^n is the number of rounds that the task of sensor s_n waits for computation in the edge server es_m, and ET_z^m is the time that the edge server es_m spends in the z-th round of task computation, but the task of edge server s_n does not always

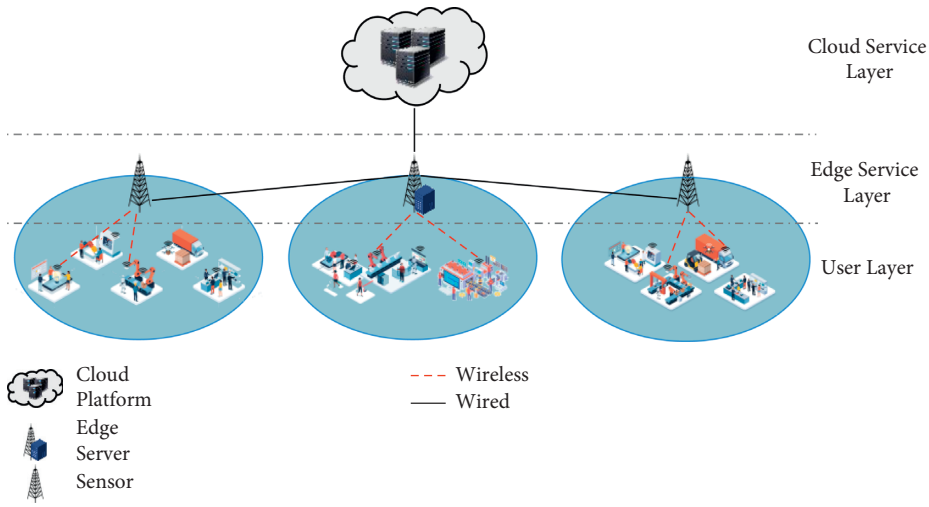


Figure 1: The edge computing model orienting services offloading in the IoT.

arrive at the edge server immediately, so the waiting time of this task needs subtracting the time delay spent in the task transmission process.

The computing formula for the time taken by the edge server to return the computing result to the sensor s_n is shown in formula (5), where ds_n' represents the task computing result of the sensor s_n.

$$DT_n = \left(1 - Y_n^m\right) \cdot sn_n^m \cdot \frac{ds_n'}{\lambda}, \tag{5}$$

where Y_n^m is a binary state variable, sn_n^m is the number of sensors passed by the sensor s_n to the edge server es_m.

Therefore, the task transmission delay of the task from the sensor to the edge server, the task processing delay of the edge server processing the task, the task waiting time of the task waiting in the edge server, and the result return time of the server returning the computing result to the sensor together constitute the task transmission delay and calculation delay of sensor s_n:

$$ST_n = AT_n + BT_n + CT_n + DT_n, \tag{6}$$

and then the formula for computing the average delay of all sensor tasks is as follows:

$$\text{AST} = \frac{1}{N} \sum_{n=1}^{N} ST_n. \tag{7}$$

3.3. Energy Consumption Model. Considering that sensors are always collecting environmental information and transmitting tasks to the edge server, the running time of the edge server is the key to computing energy consumption. The time computing formula for the service provided by the *m-th* edge server es_m is as follows:

$$\text{SPT}_m = \sum_{z=1}^{Z_m} ET_z^m, \tag{8}$$

where Z_m is the total number of rounds of the *m-th* edge server es_m computing task.

Defining ρ as the operating power of edge servers, the basic energy consumption to keep all edge servers running continuously is as follows:

$$AE = \sum_{m=1}^{M} \left(\text{SPT}_m \cdot \rho\right). \tag{9}$$

Defining σ as the power of edge server task processing, the energy consumption of the m-th edge server es_m task processing is as follows:

$$\text{VME}_m = \text{SPT}_m \cdot \sigma, \tag{10}$$

and then the operating energy consumption of all corresponding edge server computing tasks is as follows:

$$BE = \sum_{m=1}^{M} \text{VME}_m. \tag{11}$$

Defining τ as the power when a single edge server is idle, and the energy consumption of the *m-th* edge server es_m without task processing is as follows:

$$\text{EVE}_m = \left(\max_{m=1}^{M} \text{SPT}_m - \text{SPT}_m\right) \cdot \sigma, \tag{12}$$

where the SPT_m is the running time of the *m-th* edge server es_m, and σ is the power of edge server task processing. Then, the corresponding energy consumption when the edge server is idle:

$$CE = \sum_{m=1}^{M} \text{EVE}_m. \tag{13}$$

Therefore, the total energy consumption of the edge server is as follows:

$$TE = AE + BE + CE. \tag{14}$$

3.4. Load Model. Considering that load describes the number of computing tasks in all edge servers, it is more appropriate to use load variance to evaluate whether load

balancing is achieved between edge servers. Defining a binary variable determines whether the m-th edge server es_m is being occupied:

$$EO_m = \begin{cases} 1, & \text{edge server } es_m \text{ is being occupied,} \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The total number of edge servers occupied is as follows:

$$NE = \sum_{m=1}^{M} EO_m. \tag{16}$$

The load of the *m-th* edge server es_m is measured by the number of tasks processed in es_m, and the average load of the edge server is shown in formula (17), where TP_m represents the number of tasks processed in the *m-th* edge server es_m:

$$\text{ARU} = \frac{1}{NE} \sum_{m=1}^{M} TP_m. \tag{17}$$

Therefore, the load variance of the m-th edge server es_m is as follows:

$$LV_m = (\text{ARU} - TP_m)^2. \tag{18}$$

Finally, the average load variance of all occupied edge servers is as follows:

$$\text{ALV} = \frac{1}{NE} \sum_{m=1}^{M} LV_m. \tag{19}$$

3.5. Problem Definition. This article aims to minimize the average delay in formula (7), the total energy consumption in formula (14) and the variance of the average load in formula (19). The multiobjective optimization problem is defined as

$$\begin{aligned} &\min \text{AST}, \\ &\min TE, \\ &\min \text{ALV}, \\ &\text{s.t.} m \le n, \\ &\text{s.t.} \max_{m=1}^{M} ds_n \le \min_{m=1}^{M} wl_m, \end{aligned} \tag{20}$$

where wl_m represents the maximum processing task size of the *m-th* edge server es_m, and formula (21) guarantees that the total number of edge servers is less than the total number of sensors. In addition, formula (20) guarantees that the task size of a single sensor is less than the maximum processing task size of a single edge server.

4. Method of Edge Server Placement Orienting Service Offloading

In a scenario where hundreds of sensors are densely distributed, a certain number of edge server layout strategies are iteratively generated to minimize the delay in the service offloading process, minimize the energy consumption of the edge server, and balance the load of the edge server. In this case, a low-complexity suboptimal algorithm for solving the NP-hard problem is needed to find the optimal solution for the edge server layout. This article proposes an edge server layout strategy optimization method EPMOSO based on GA and PSO to iteratively optimize the edge server layout strategy.

4.1. Optimizing the Edge Server Layout Strategy Based on GA. GA is a commonly used heuristic algorithm for solving MINLP. It is an algorithm that simulates the law of survival of the fittest in nature to search for the optimal solution randomly. It only requires that the problem to be solved is computable. The GA process is divided into four steps: Initialization, Selection, Crossover, and Mutation. The details of each step of GA are as follows.

4.1.1. Determination of Initialization and Task Transmission Route. As GA simulates the law of survival of the fittest in nature, genes and chromosomes are important optimization objects for GA. In the process of using GA to iteratively optimize the layout strategy of edge servers, the layout position of a single edge server is regarded as a gene, and the genes corresponding to the layout positions of all edge servers together constitute a chromosome, let ESP_m denote the position of the m-th edge server and then $ESP = \{ESP_1, ESP_2, \ldots, ESP_M\}$ constitutes a chromosome. During the initialization, each chromosome randomly assigns geographic locations to each edge server, forming an initial set of edge server layout strategies for subsequent iterations.

Because this article studies the impact of edge server layout on edge service quality, to compare the advantages and disadvantages of different edge server layout strategies, the task transmission method is determined in advance. The sensor transmits the task to the edge server that is the closest physical distance to itself to reduce the task transmission delay. A load threshold is set for the edge server to achieve load balancing of the edge server as much as possible. When the load of the edge server to which the sensor is to be transmitted exceeds the load threshold, the sensor transmits the task to the next edge server with the closest physical distance excluding this edge server. The task sensor path confirmation algorithm is shown in Algorithm 1.

4.1.2. Selecting Fitness Function. In GA, the fitness function is used to measure the adaptability of the chromosome. The fitness function is based on the average delay of the task transmission process shown in formula (7), and the total energy consumption of the edge server shown in formula (14). The average load variance of the edge server shown in formula (19) is used to compute the chromosome fitness. Tower the transmission delay, the lower the total energy consumption of the edge server and the lower the average load variance of the edge server, the better the server layout strategy of the corresponding edge server. Therefore, when solving the problem of edge server layout, the lower the fitness of a chromosome, the more it indicates that the chromosome has strong adaptability in the chromosomes of this iteration. The fitness function is defined as formulas

Inputs: Sensor S, Edge server ES, Task threshold of ES
Output: Determined task transmission path
(1) **for** $n = 1$ **to** N **do**
(2) Find the nearest edge server es_m
(3) **if** the number of task of edge server $es_m \leq$ task threshold of ES **then**
(4) Delete es_m from ES
(5) Go to step 2
(6) else
(7) Confirm that the sensor sn will transmit the task to the edge server es_m
(8) Number of tasks of edge server $es_m + 1$
(9) end if
(10) end for
(11) **return** Determined task transmission path

Algorithm 1: Task transmission path confirmation algorithm.

(23)–(26). Formula (23) represents the fitness function of transmission delay, formula (24) represents the fitness function of edge server energy consumption, formula (25) represents the fitness function of edge server load balancing, and formula (26) represents the comprehensive fitness function of the *k-th* chromosome, where w_d, w_e, and w_l are the weights of delay, energy consumption, and load balancing fitness, respectively.

After computing the fitness, GA performs selection operations based on the adaptive capacity of each chromosome. In the selection operation, the commonly used selection algorithms include Roulette Wheel and Tournament. The roulette algorithm puts all chromosomes into a wheel for selection, and the probability of each chromosome being selected is proportional to its fitness. Therefore, the roulette algorithm is more suitable for the problem of maximum optimization. However, the championship algorithm not only requires low computing resources but also does not need to modify the code itself when applied to the minimum optimization problem. Therefore, this article uses the tournament algorithm as the selection algorithm. The tournament algorithm will select the few chromosomes with the strongest adaptability, that is, the lower adaptability, to copy directly for the next iteration.

$$\begin{cases} DF_k = \dfrac{\mathrm{ASK}_k}{\sum_{k=1}^{K} \mathrm{ASK}_k}, \\ EF_k = \dfrac{TE_k}{\sum_{k=1}^{K} TE_k}, \\ LF_k = \dfrac{\mathrm{ALV}_k}{\sum_{k=1}^{K} \mathrm{ALV}_k}, \end{cases} \tag{21}$$

$$AF_k = w_d \cdot DF + w_e \cdot EF + w_l \cdot LF_k. \tag{22}$$

4.1.3. Crossover and Mutation. In order to increase the diversity of chromosomes and extend the search range of strategies to generate better edge server layout strategies and search for global optimal solutions and avoid falling into local optimal solutions, GA introduces crossover and mutation operations. Selection operations retain a certain number of ancestral chromosomes, while crossover and mutation operations are used to generate a certain number of descendant chromosomes.

In the process of crossover, two ancestral chromosomes are randomly selected to exchange several corresponding genes, thereby generating two new offspring chromosomes. When dealing with the multiobjective optimization problem, the probability of crossover is designed for the crossover operation, and each gene is based on the possibility of crossover to exchange, that is, if the gene needs to be crossed, the corresponding edge server positions of the two ancestral chromosomes will be exchanged. If other edge servers of the exchanged chromosomes have been deployed in the position to be exchanged, the edge server will be reset to a new location. An example of the crossover operation is shown in Figure 2(a).

In the process of mutation, two ancestral chromosomes are randomly selected, and each gene on the two chromosomes mutates according to the mutation probability, thereby generating a new descendant chromosome. When dealing with the multiobjective optimization problem, each gene on the chromosome is mutated according to the mutation probability, that is, if the gene needs to be mutated, the edge server will be randomly deployed in a random location, if other edge servers have been deployed in this new location, then a random algorithm is used to reallocate the location until no edge server is deployed in the location. An example of the mutation operation is shown in Figure 2(b).

Due to the use of the PSO algorithm to optimize the performance of the GA algorithm, to facilitate the implementation of the subsequent PSO algorithm, in addition to the basic steps of the GA, the chromosome corresponding to the historical minimum fitness and the historical minimum fitness during the iteration process of the numbered chromosome will also be recorded. The corresponding chromosomes are used for PSO algorithm iteration, and these chromosomes are updated after each round of GA iteration. Algorithm 2 describes the specific process of Genetic Algorithm iterative optimization of edge server layout strategy.

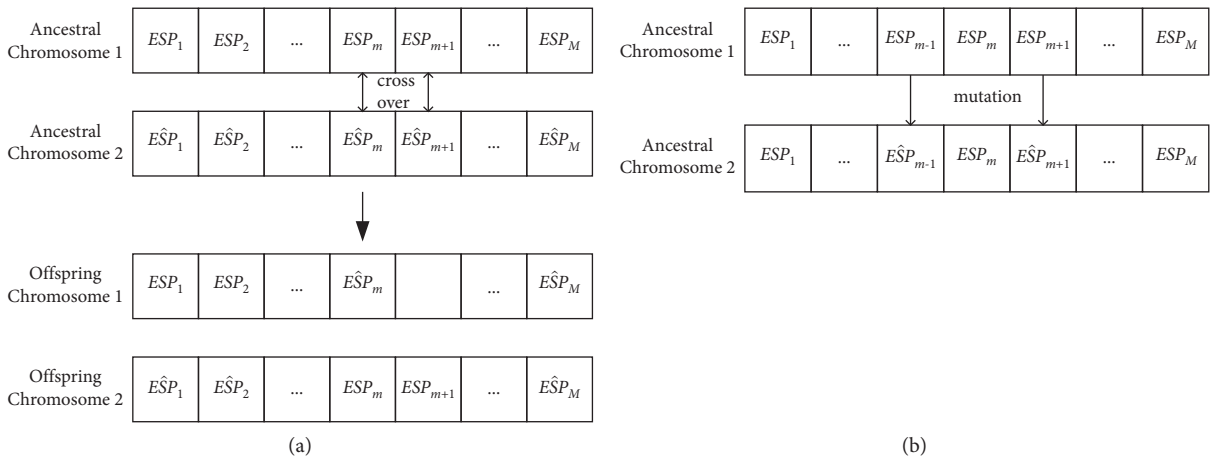


Figure 2: Chromosome crossover and mutation. (a) Chromosome crossover operation. (b) Chromosome mutation operation.

4.2. Optimizing the Edge Server Layout Strategy Based on PSO. PSO is also a heuristic algorithm commonly used to solve MINLP. It is an intelligent population optimization algorithm based on social information sharing. The algorithm originated from the study of the information sharing behavior of bird flocks during predation. The purpose of individuals in a flock is to search for food, but the individual does not know the specific location of the food. When an individual in the flock finds food, it will share the location of food with the flock, and then, other individuals will move in that direction. With the continuous sharing of information, all individuals in the flock can find food. PSO does not have the crossover and mutation operations of GA and has the advantages of high accuracy and fast convergence.

Particles are the optimization objects of the PSO algorithm. Because this article uses PSO to optimize the performance of GA, the particles of the PSO algorithm are equivalent to the genes of GA. But to facilitate the realization of the PSO algorithm, a speed attribute is added to the GA gene. Therefore, the particle (gene) has two attributes: position and speed. The position attribute of the particle represents the layout position of a single edge server in the current state, and the speed attribute of the particle controls the direction and distance of the layout position of the edge server. The process of PSO includes two steps: initialization and particle attribute update:

(1) Initialization: considering that PSO is used to optimize the convergence speed of GA, the PSO initialization part in this article cancels the initialization of the position attributes of each particle and uses the chromosome generated by GA iteration as input to initialize the speed of each particle.

(2) Particle attribute update: the particles are deployed separately in the scene, and the optimal solution found is recorded as the individual optimal position, and the optimal solution found in the population is recorded as the global optimal position. To avoid the result of the population falling into the local optimal solution, it is necessary to adjust two attributes of speed and position according to the global optimal position when each particle adjusts the two attributes of speed and position.

$$V_{t+1}^{k,m} = V_t^{k,m} + c_1 \cdot \left(\text{pbest}_t^m - X_t^{k,m}\right) + c_2 \cdot \left(\text{gbest}^m - X_t^{k,m}\right), \quad (23)$$

$$X_{t+1}^{k,m} = X_t^{k,m} + V_{t+1}^{k,m}. \quad (24)$$

The velocity and position of the particles are iterated according to formulas (23) and (24), where $V_t^{k,m}$ represents the velocity of the m-th particle of the k-th chromosome at the t-th iteration, and $X_t^{k,m}$ represents the position of the m-th particle of k-th chromosome. $c1$ and $c2$ are acceleration constants, generally set to 0.5. $pbest_t^m$ represents the best position of the m-th particle in the t-th iteration, and $gbest^m$ represents the best position of the m-th particle in the historical iteration.

In the process of using PSO to iteratively optimize the edge server layout strategy, each gene of the chromosome in the iteration process adjusts its position based on the chromosome the historical minimum fitness recorded by GA and the historical minimum fitness during the iteration. Algorithm 3 describes the specific process of using PSO to optimize GA iteratively optimized chromosomes.

4.3. Selecting the Optimal Edge Server Layout Strategy Based on SAW. A set of optimized edge server layout strategies optimized by GA and PSO iteratively; the SAW decision-making method is used to determine the final edge server layout strategy. First, the three indicators of delay, energy consumption, and load of the edge server layout strategy were standardized:

Inputs: K chromosomes, Number of iterations T, variable t, the number of chromosomes to operate k
Output: K chromosomes after GA iteration optimization
(1) for $t = 0$ to T do
(2) if $t == 0$ then
(3) Initialize K chromosomes
(4) else
(5) Compute the fitness of K chromosomes and select k chromosomes with lower fitness
(6) Record the chromosome with the lowest fitness
(7) Select two chromosomes randomly and perform crossover operations
(8) Select two chromosomes randomly and perform mutation operations
(9) end if
(10) end for
(11) return K chromosomes

Algorithm 2: Optimizing edge server layout strategy based on GA.

Inputs: K chromosomes that have been iteratively optimized by GA, Number of iterations T, Variable t
Output: K chromosomes optimized by PSO
(1) for $t = 0$ to T do
(2) if $t == 0$ then
(3) Initialize the velocity of each particle of K chromosomes
(4) **else**
(5) Adjust the position of each particle according to formulas (23) and (24)
(6) end if
(7) end for
(8) **return** K chromosomes

Algorithm 3: Edge server layout strategy based on PSO.

$$DU = \begin{cases} \dfrac{\mathrm{AST}_{\max} - \mathrm{AST}}{\mathrm{AST}_{\max} - \mathrm{AST}_{\min}}, & \mathrm{AST}_{\max} \neq \mathrm{AST}_{\min}, \\ 1, & \mathrm{AST}_{\max} = \mathrm{AST}_{\min}, \end{cases}$$

$$EU = \begin{cases} \dfrac{TE_{\max} - TE}{TE_{\max} - TE_{\min}}, & TE_{\max} \neq TE_{\min}, \\ 1, & TE_{\max} = TE_{\min}, \end{cases} \tag{25}$$

$$LU = \begin{cases} \dfrac{\mathrm{ALV}_{\max} - \mathrm{ALV}}{\mathrm{ALV}_{\max} - \mathrm{ALV}_{\min}}, & \mathrm{ALV}_{\max} \neq \mathrm{ALV}_{\min}, \\ 1, & \mathrm{ALV}_{\max} = \mathrm{ALV}_{\min}, \end{cases}$$

where $AST_{\max}$, $AST_{\min}$, $TE_{\max}$, $TE_{\min}$, $ALV_{\max}$, and $ALV_{\min}$, respectively, represent the highest and lowest transmission delay of the chromosomes optimized iteratively, the energy consumption of the edge server, and the variance of the overall load balance of the edge server. In the standardization operation, there are two ways to compute the numerator: the maximum value of the optimization index minus the current value of the optimization index and the current value of the optimization index minus the minimum value of the optimization index. Taking the delay of the transmission into account, the total energy consumption of the edge server and the variance of the average load of the edge server are both minimum optimization problems. Therefore, in the standardization operation, the current value of the optimization index is subtracted from the maximum value of the optimization index.

After standardizing each index, the utility value of the corresponding chromosome is computed according to the corresponding weight of delay, energy consumption, and load balance, as shown in formula (26), where DU, EU, and LU are standardized values of delay, energy consumption, and the variance of load, respectively, and w_d, w_e, and w_l are the weights corresponding to the delay, energy consumption, and variance of load, respectively. The utility value is used to determine whether the edge server layout strategy corresponding to the chromosome is the optimal edge server layout strategy. After computing the utility value, the edge server layout strategy with the highest utility value is selected as the final edge server layout strategy:

$$AU = w_d \cdot DU + w_d \cdot EU + w_l \cdot LU. \tag{26}$$

4.4. Summary of the Method. Both GA and PSO are heuristic algorithms commonly used to solve MINLP. The GA algorithm simulates the natural phenomenon of retaining excellent genes and eliminating genes that are not suitable for the environment during the population iteration process,

thereby solving the target problem. The advantage of GA is that it can better find the global optimal solution, but its convergence speed is slow, and a satisfactory solution can only be obtained when the number of iterations is large. The advantage of the PSO algorithm is that the algorithm is simple, and the convergence speed is fast, but it can often only find the target solution in a limited search area, so it is easy to fall into the local optimal solution instead of the global optimal solution [33]. Therefore, in this article, to integrate the advantages of GA and PSO, use PSO to optimize GA and propose an edge server layout strategy research method for offloading services in IoT, EPMOSO, which uses GA and PSO to iteratively optimize a set of excellent edge server layout strategy, and finally use SAW method to determine the final edge server layout strategy. An example of EPMOSO is shown in Figure 3.

Algorithm 4 describes the core process of EPMOSO. Algorithm 3, taking the sensor set, edge server set, and iteration number as input, first randomly allocates the initial test position for the edge server and initialize the chromosomes, use the set of chromosomes as the input of GA, and perform coarse-grained optimization. Then, selection, crossover, and mutation are performed; the chromosome of this iteration is used as the input of PSO, and the position of each gene is updated. After the fine-grained optimization of PSO, this iteration ends. Until the number of iterations is full and output a set of optimized edge server layout strategies. For this group of edge server layout strategies, compute its utility value and use the SAW method to determine the final edge server layout strategy.

5. Experiment Analysis

In this section, the effectiveness of EPMOSO is evaluated by analyzing the results of comparative experiments. First, introduce the experimental configuration of the experiment in this article and introduce the content of the algorithm compared with EPMOSO. Then, according to the experimental results, the pros and cons of the EPMOSO and other methods proposed in this article are analyzed from different angles.

5.1. Experimental Configuration. In comparison experiments, the effectiveness and efficiency of EPMOSO were compared with the other algorithm in the case of different edge server sizes. The parameter settings of this experiment are shown in Table 2, and the two comparison algorithms in the experiment are shown as follows:

(1) Genetic Algorithm [33]: the genetic algorithm and EPMOSO are used separately to compare the convergence trends of the two methods to evaluate the convergence speed of EPMOSO, to highlight that the EPMOSO method still has a good convergence speed when the global optimal solution can be found.

(2) Particle population algorithm [34] the particle population algorithm PSO and EPMOSO are used separately to verify whether the edge server layout strategy iterated by EPMOSO is the global optimal solution and highlight the advantages of EPMOSO over PSO in solving the global optimal solution.

5.2. Comparative Analysis. In the comparative analysis, the effectiveness of EPMOSO, GA, and PSO is analyzed from four aspects: the delay of the transmission, the total energy consumption of the edge server, the average load variance, and the utility value of the edge server. In addition, the convergence rate of the three methods is evaluated by analyzing the convergence curves of the indicators of the three algorithms in the iterative process. Finally, under different edge server scales, the optimal solution selected by SAW selection iteration is presented in Figure 10.

5.2.1. Comparison of Delay. In the experiment of this article, all edge servers have the same configuration, so the task processing speed of each edge server is the same. Under the same edge server scale, the task processing time of the edge server during the iteration of EPMOSO, GA, and PSO is always the same. EPMOSO, GA, and PSO mainly optimize the transmission time of the task and the waiting time of the task in the edge server. In addition, because tasks have more edge servers that are closer to each other as task transmission options, and there is no need to migrate tasks to edge servers that are farther away, the task transmission time will decrease as the number of edge servers increases. In addition, due to the increase in the number of edge servers, the number of tasks gathered on the same edge server is reduced, so the waiting time of tasks in the edge server is also reduced. In Figure 4, it can be seen from the image that when the edge server scale is 15, 20, 25, and 30, compared with GA and PSO, EPMOSO has a stronger ability to optimize latency.

5.2.2. Comparison of Energy Consumption. In the experiment, considering that the energy consumption of a single edge server is related to the number of tasks gathered in the edge server and the processing time of these tasks, as the number of edge servers increases, the number of tasks gathered in a single edge server is relatively reduced. The energy consumption of the server is also relatively reduced. Although the energy consumption of a single edge server is relatively reduced, the increase in the number of edge servers leads to an increase in the total energy consumption of edge servers. Figure 5 compares the total energy consumption of edge servers under different edge server sizes. It can be seen from the figure that when the edge server scale is 15, 20, 25, and 30, compared with GA and PSO, EPMOSO has relatively small advantages in energy consumption optimization.

5.2.3. Comparison of Load Variance. In the experiment, the task load threshold is set for each edge server. Therefore, before the algorithm is optimized, all edge servers have basically realized load balancing, but with the continuous optimization of the algorithm, the layout of edge servers tends to be reasonable, then the load variance of the edge server will still achieve a certain optimization. Figure 6 presents the edge server load variances optimized by

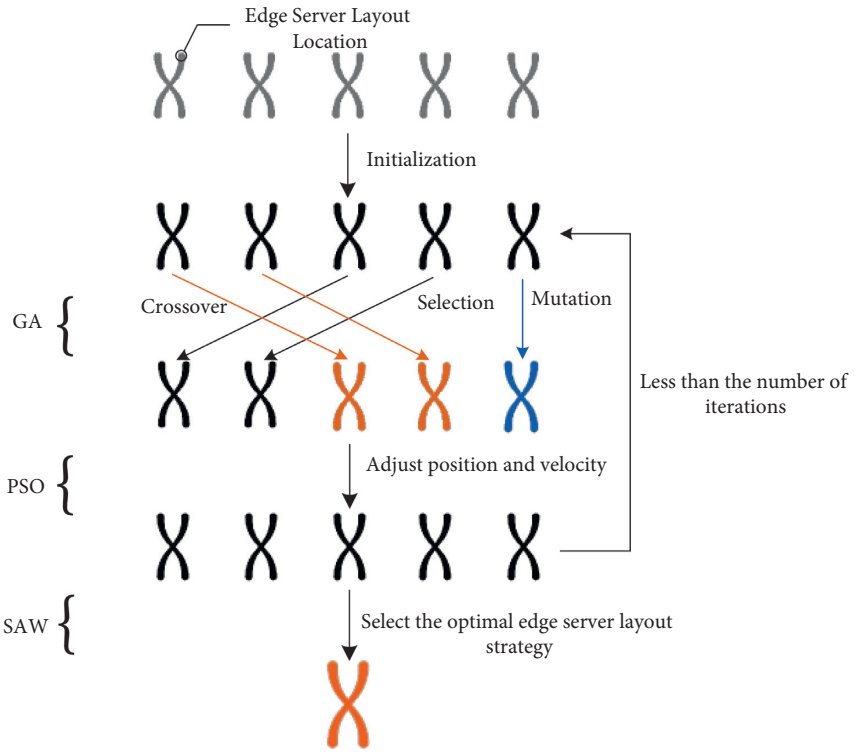


Figure 3: Example of EPMOSO.

Inputs: Sensor collection S, Edge server collection ES, Number of iterations T, Variable t
Output: Determined final edge server layout strategy
(1) Randomly set the location of the edge server and initialize the chromosome
(2) Initialize the velocity of each gene of each chromosome
(3) for $t = 0$ to T do
(4) Execute GA algorithm according to Algorithm 2
(5) Execute PSO algorithm according to Algorithm 3
(6) end for
(7) Compute the utility value of K chromosomes
(8) Use SAW algorithm to determine the final edge server layout strategy
(9) **return** Best Edge Server Layout Strategy

Algorithm 4: Summary of EPMOSO method.

Table 2: Experimental parameters.

Parameter	Value
Number of sensors N	200
Number of edge servers M	15, 20, 25, 30
Threshold of the number of edge server tasks	18, 14, 12, 8
Sensor task size	[30–100] MB
Task transfer rate	10 MB/s
Idle power of edge server	100 W
Operating power of edge server	300 W
Number of chromosomes	10
Number of iterations	50
Number of chromosomes selected	6
Number of chromosome crossovers	2
Probability of chromosome crossovers	0.6
Number of chromosome mutations	2
Probability of chromosome mutations	0.8

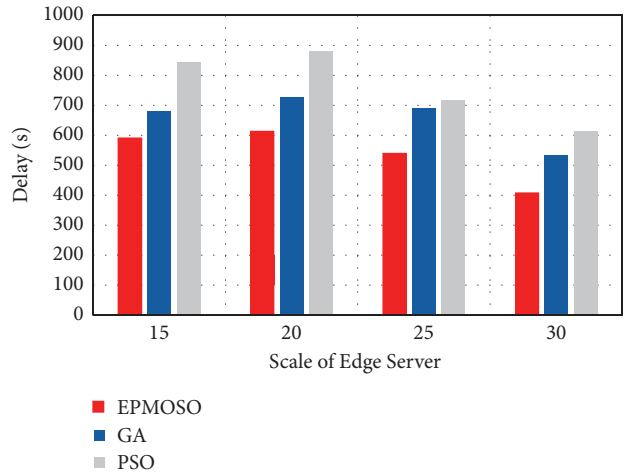


Figure 4: Comparison of delay under different edge server scales.

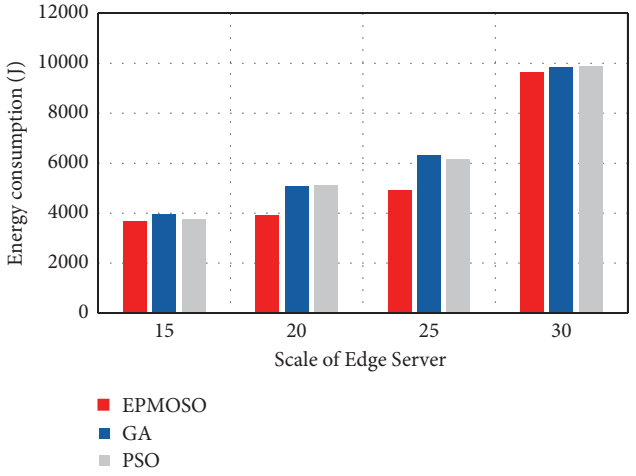


Figure 5: Comparison of energy consumption under different edge server scales.

EPMOSO, GA, and PSO under different edge server sizes. It can be seen from the image that when the edge server size is 15, 20, 25, and 30, EPMOSO still shows better optimization capabilities.

5.2.4. Comparison of Utility. In this experiment, the SAW method is used to determine the final edge server layout strategy. The SAW method first computes the utility value of the chromosome. The higher the utility value, the better the edge server layout strategy corresponding to the chromosome. Therefore, in the continuous iterative optimization process, the utility value of the chromosome will be higher and higher, indicating that the solution searched by the optimization algorithm is getting closer and closer to the global optimal solution. Figure 7 presents the utility value of the edge server layout strategy determined by SAW and iterative optimization of EPMOSO, GA, and PSO under different edge server scales. Obviously, when the edge server scale is 15, 20, 25, and 30, the edge server layout strategy

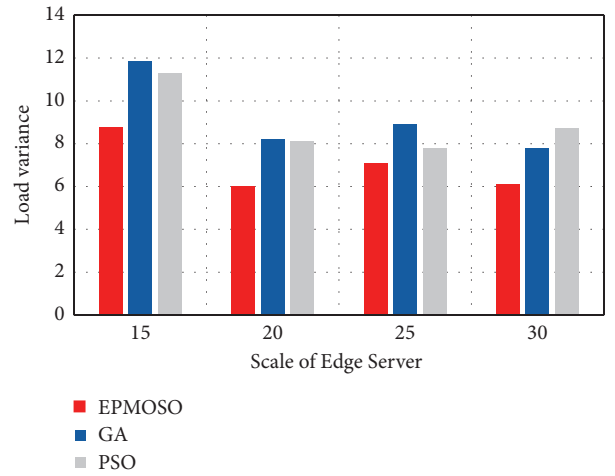


Figure 6: Comparison of load variance under different edge server scales.

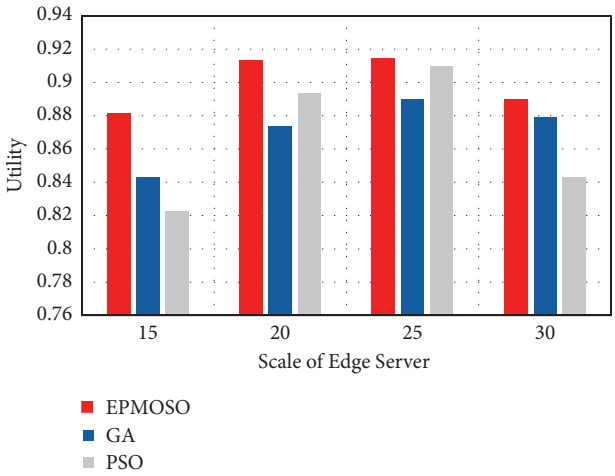


Figure 7: Comparison of utility under different edge server scales. (a) Edge server scale: 15. (b) Edge server scale: 20. (c) Edge server scale: 25. (d) Edge server scale: 30.

optimized by EPMOSO has a higher utility value. Therefore, compared with GA and PSO, using EPMOSO for iterative optimization can obtain a better edge server layout strategy.

5.2.5. Comparison of Convergence Speed. In the experiment, considering that if there are too many iterations, EPMOSO, GA, and PSO will gradually tend to converge, and it is impossible to directly determine the convergence speed of EPMOSO through comparison. Therefore, the comparison experiment of EPMOSO, GA, and PSO iteratively optimizes the edge server layout strategy is set to 50 times. After EPMOSO, GA, and PSO are initialized, they each iterate 50 times to iteratively optimize a set of edge server layout strategies. Figures 7–9 compare the process of EPMOSO, GA, and PSO to optimize the delay of the task transmission process, the total energy consumption of the edge servers, and the load variance of the edge serves.

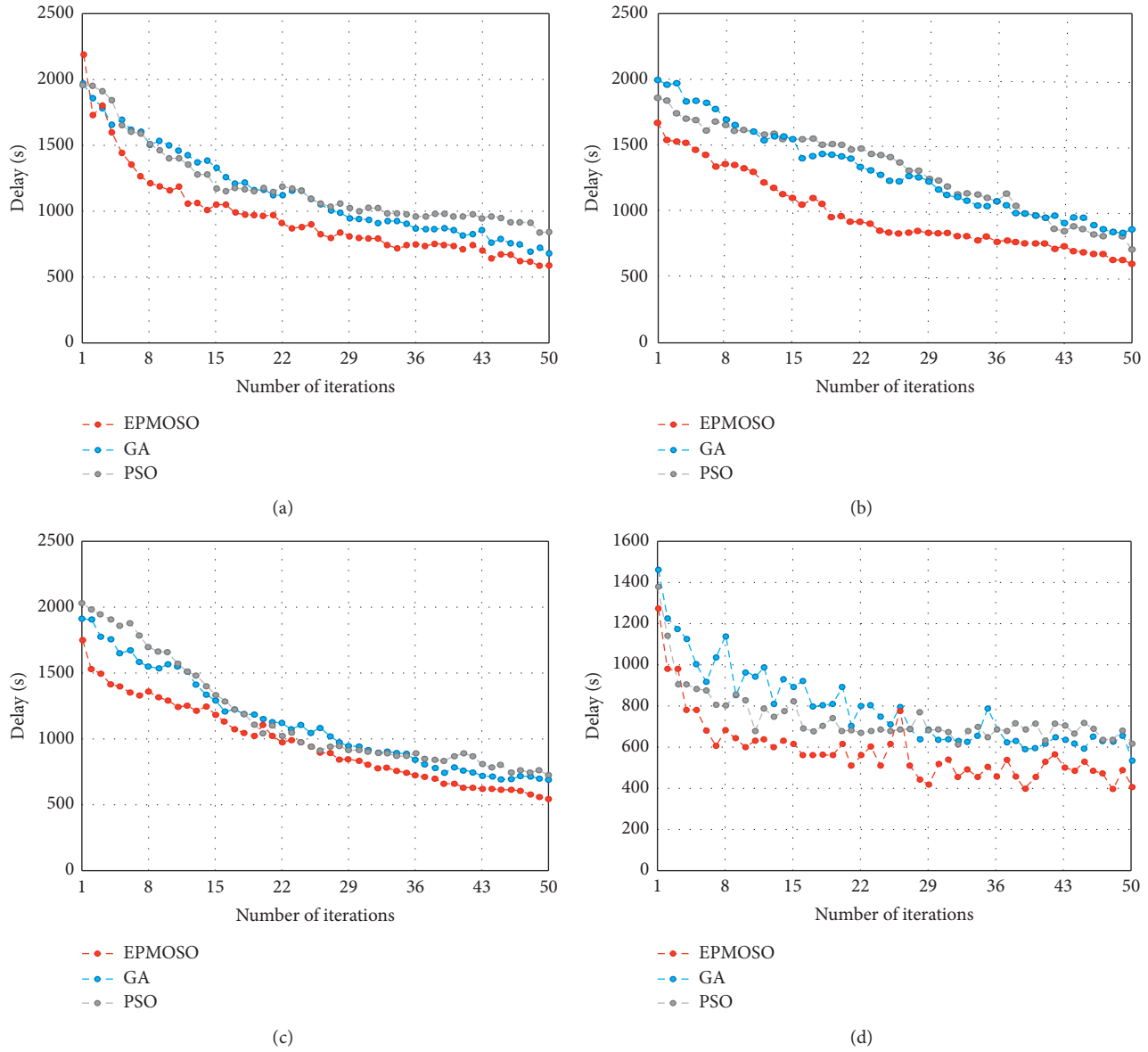


Figure 8: Comparison of delay convergence rate under different edge server scales. (a) Edge server scale: 15. (b) Edge server scale: 20. (c) Edge server scale: 25. (d) Edge server scale: 30.

In Figure 8, Figures 8(a)–8(d) compare the changes of delay in the iterative process when the edge server size is 15, 20, 25, and 30. In Figures 8(a)–8(d), EPMOSO showed a faster convergence rate in the first 20 iterations and a lower delay after optimization. In the next 30 iterations, the edge server layout strategy in the iteration is getting closer and closer to the global optimal solution, so the convergence speed of the three algorithms slows down. Therefore, compared with GA and PSO, EPMOSO not only has a better effect in optimizing the delay of the transmission process but also has a better convergence speed. Therefore, EPMOSO is effective for optimizing the time delay of the task transmission process.

In Figure 9, Figures 9(a)–9(d) compare the changes in energy consumption during the iteration. When the scale of edge server is 15, EPMOSO spend lower energy initially than GA and PSO, but as the number of iterations increases, the energy consumption of EPMOSO and PSO is similar, even when the number of iterations reaches 50, the energy consumption of three is almost same. When the scale of edge server is 20, the performance of the three is almost the same, but EPMOSOS has a slight advantage throughout the iteration process, the more the number of iterations, the more obvious its advantage. When the scale of edge server is 25, the same conclusion can be drawn as when the scale is 20. When the scale of edge server is 30, EPMOSO's performance is slightly worse than the other two algorithms initially; when the number of iterations reaches 10, its energy consumption is slightly better than the other two algorithms but fluctuates up and down. Although compared with GA and PSO, EPMOSO's advantage in convergence speed is not obvious, but EPMOSO's optimization of the total energy consumption of edge servers is effective.

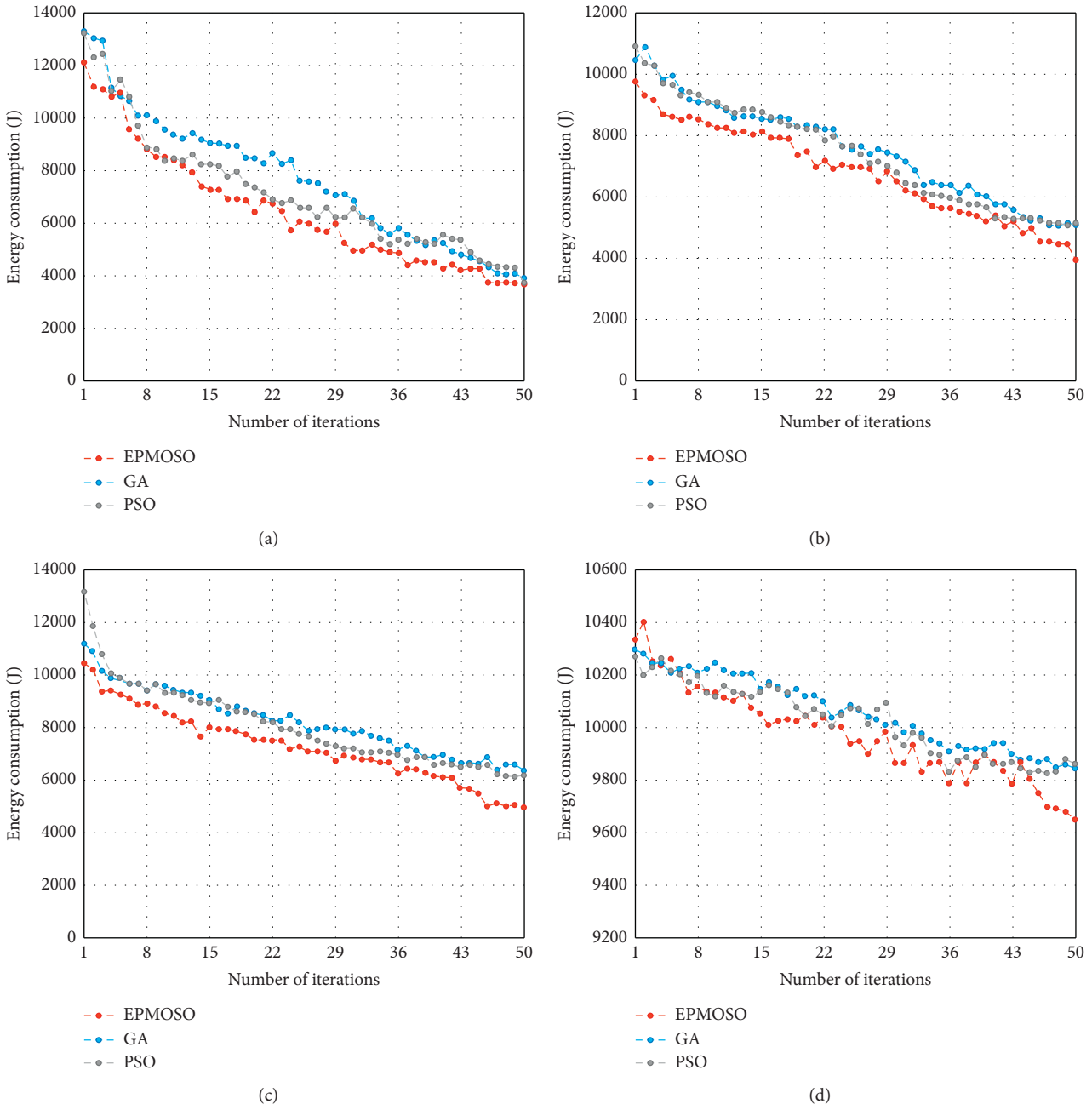


Figure 9: Comparison of the convergence rate of energy consumption under different edge server scales. (a) Edge server scale: 15. (b) Edge server scale: 20. (c) Edge server scale: 25. (d) Edge server scale: 30.

In Figure 10, Figures 10(a)–10(d) compare the difference in load variance during the iteration. When the scale of edge server is 15, EPMOSO showed a faster convergence rate in the first 35 iterations and a lower load variance, but the convergence rate of the three afterward is almost the same. When the edge server scale is 20, the convergence speed of the three is not much different, but EPMOSO is slightly better than the other two. When the scale of edge server is 25, the same conclusion can be drawn as when the scale is 20. When the scale of edge server is 30, the convergence speed of EPM and PSO in the first 20 iterations are basically the same, but it is better than the other two afterward. Considering that this article has set task thresholds for all edge servers, it is reasonable that the optimization effect of edge server load variance is not obvious. However, it can be seen from the figure that EPMOSO can still optimize the load variance of edge servers to a certain extent.

In summary, EPMOSO can optimize the delay of the task transmission process, the total energy consumption of the edge server, and the load variance of the edge server. In the

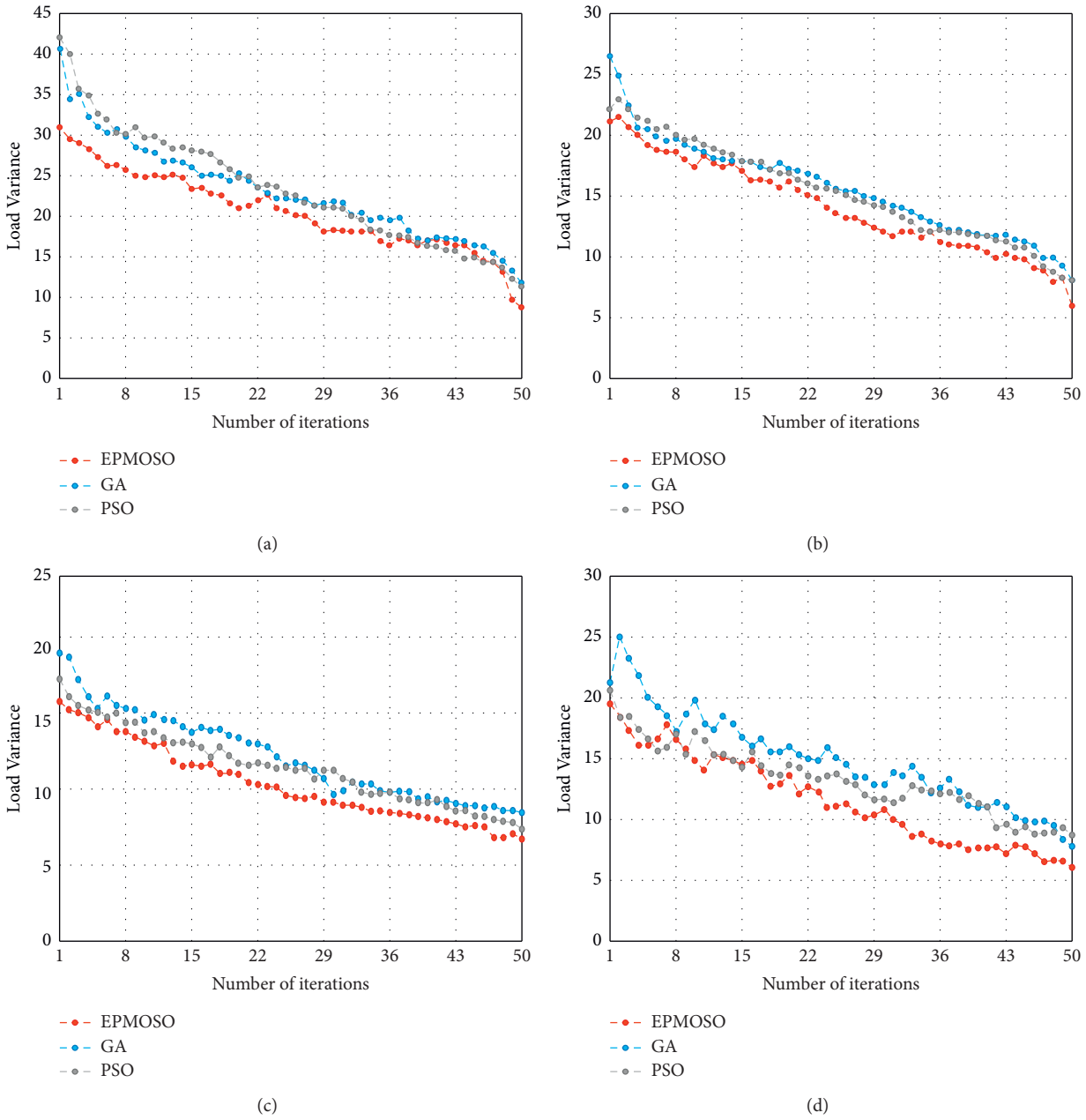


Figure 10: Comparison of the convergence rate of load variance under different edge server scales. (a) Edge server scale: 15. (b) Edge server scale: 20. (c) Edge server scale: 25. (d) Edge server scale: 30.

comparison between EPMOSO and GA, although GA also can find the global optimal solution, when the number of iterations is 50, GA obviously cannot achieve better convergence, and EPMOSO can not only find the global optimal solution but also achieve relatively better convergence. In comparing EPMOSO and PSO, PSO converges faster in the early stage of the iterative process. However, PSO is more likely to fall into a local optimal solution, so the convergence speed of PSO slows down in the later stage of the iterative process, and the optimization effect becomes worse. While EPMOSO maintains a good convergence rate, the optimization effect is still good.

5.2.6. Optimal Layout Strategy. The experiment in this article uses GA and PSO to iteratively optimize a set of excellent edge server layout strategies, and the SAW method determines the final edge server layout strategy. The optimal edge server layout strategy under different edge server scales is given in Figure 11. In theory, the layout of the edge server should be symmetrical and regular but considering that the task size of the sensor in this experiment is only a fixed range, and the task size of different sensors is set randomly; therefore, the layout of the edge server of the experimental results is asymmetric and irregular.

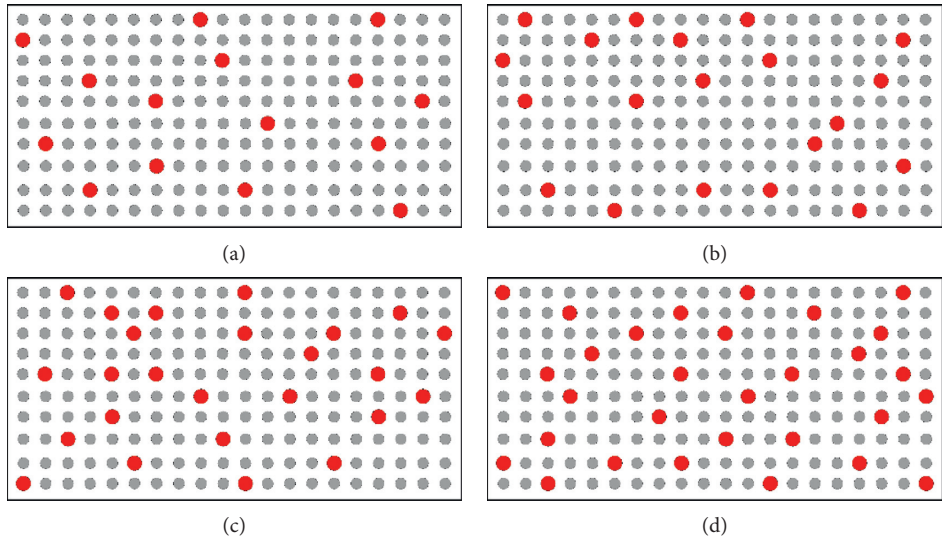


Figure 11: Edge server layout under different edge server scales. (a) Edge server scale: 15. (b) Edge server scale: 20. (c) Edge server scale: 25. (d) Edge server scale: 30.

6. Conclusion

Aiming at the problem of poor edge service quality and poor real-time control effect under a relatively small number of edge server scales, a research method of edge server layout strategy for offloading services in IoT is proposed to provide services with lower latency while reducing the edge server energy consumption and ensuring the stability of the edge server system. This method quantifies the above problem as a multiobjective optimization problem, which aims to reduce the time delay of the task transmission process and the energy consumption of the edge server while realizing the overall load balance of the edge server and proposes EPMOSO, which is a research method of edge server layout strategy for offloading services in IoT. This method first uses GA and PSO to iteratively optimize a set of excellent edge server layout strategies and then uses the SAW method to determine the optimal edge server layout strategy. The results of comparative experiments show that the EPMOSO method proposed in this article has the advantage of a better convergence speed when the global optimal solution can be found.

Data Availability

The raw data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent IoT applications: a cloud-edge based framework," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 35–44, 2020.

[2] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for IoV in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[3] W. H. Hassan, "Current research on Internet of Things (IoT) security: a survey," *Computer Networks*, vol. 148, pp. 283–294, 2019.

[4] L. D. Xu, Y. Lu, and L. Li, "Embedding blockchain technology into IoT for security: a survey," *IEEE Internet of Things Journal*, vol. 8, no. 13, Article ID 10452, 2021.

[5] H. Elazhary, "Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: d," *Journal of Network and Computer Applications*, vol. 128, pp. 105–140, 2019.

[6] L. Kong, M. K. Khan, F. Wu, G. Chen, and P. Zeng, "Millimeter-wave wireless communications for IoT-cloud supported autonomous vehicles: overview, design, and challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 62–68, 2017.

[7] T. Wang, G. Zhang, A. Liu, B. Md Zakirul Alam, and Q. Jin, "A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4831–4843, 2018.

[8] X. Xu, Q. Huang, Y. Zhang, S. Li, L. Qi, and W. Dou, "An LSH-based offloading method for IoMT services in integrated cloud-edge environment," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 3s, pp. 1–19, 2021.

[9] M. Jia, Z. Yin, D. Li, and Q. Guo, "Toward improved offloading efficiency of data transmission in the IoT-cloud by leveraging secure truncating OFDM," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4252–4261, 2018.

[10] T. Wang, Y. Lu, J. Wang, H. N. Dai, X. Zheng, and W. Jia, "EIHDP: edge-intelligent hierarchical dynamic pricing based on cloud-edge-client collaboration for IoT systems," *IEEE Transactions on Computers*, vol. 70, 2021.

[11] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in IoT edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1133–1146, 2020.

[12] L. Lei, H. Xu, X. Xiong, K. Zheng, and W. Xiang, "Joint computation offloading and multiuser scheduling using approximate dynamic programming in NB-IoT edge computing system," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5345–5362, 2019.

[13] H. Tian, X. Xu, T. Lin et al., "DIMA: distributed cooperative microservice caching for Internet of Things in edge computing by deep reinforcement learning," *World Wide Web*, 2021.

[14] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 160–168, 2019.

[15] G. Cui, Q. He, F. Chen, H. Jin, and Y. Yang, "Trading off between user coverage and network robustness for edge server placement," *IEEE Transactions on Cloud Computing*, 2020.

[16] B. Shen, X. Xu, L. Qi, X. Zhang, and G. Srivastava, "Dynamic server placement in edge computing toward Internet of vehicles," *Computer Communications*, vol. 178, pp. 114–123, 2021.

[17] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of wireless sensor networks: an up-to-date survey," *Applied System Innovation*, vol. 3, no. 1, p. 14, 2020.

[18] H.-L. Truong and S. Dustdar, "Principles for engineering IoT cloud systems," *IEEE Cloud Computing*, vol. 2, no. 2, pp. 68–76, 2015.

[19] T. Dinh, Y. Kim, and H. Lee, "A location-based interactive model of Internet of Things and cloud (IoT-Cloud) for mobile cloud computing applications," *Sensors*, vol. 17, no. 3, p. 489, 2017.

[20] C. Stergiou, K. E. Psannis, B. B. Gupta, and Y. Ishibashi, "Security, privacy & efficiency of sustainable cloud computing for big data & IoT," *Sustainable Computing: Informatics and Systems*, vol. 19, pp. 174–184, 2018.

[21] M. Ashouri, P. Davidsson, and R. Spalazzese, "Cloud, edge, or both Towards Decision Support for Designing IoT applications," in *Proceedings of the 2018 Fifth International Conference on Internet of Things: Systems, Management and Security*, pp. 155–162, IEEE, Valencia, Spain, October 2018.

[22] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[23] N. Abbas, Y. Zhang, A. Taherkordi, and Tor Skeie, "Mobile edge computing: a survey[J]," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.

[24] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob, and M. Imran, "The role of edge computing in Internet of Things," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 110–115, 2018.

[25] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, "Consolidate IoT edge computing with Lightweight virtualization," *IEEE Network*, vol. 32, no. 1, pp. 102–111, 2018.

[26] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: partial computation offloading using dynamic voltage scaling," *IEEE Transactions on Communications*, vol. 64, no. 10, pp. 4268–4282, 2016.

[27] S. Yu, R. Langar, X. Fu, L. Wang, and Z. Han, "Computation offloading with data caching enhancement for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, Article ID 11098, 2018.

[28] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, 2019.

[29] S. Wang, T. Tuor, T. Salonidis et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[30] X. Xu, D. Zhu, X. Yang, S. Wang, L. Qi, and W. Dou, "Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–17, 2021.

[31] X. Xia, F. Chen, Q. He et al., "Data, user and power allocations for caching in multi-access edge computing," *IEEE Transactions on Parallel and Distributed Systems*, p. 1, 2021.

[32] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2651–2664, 2018.

[33] A. A. Al-Habob, O. A. Dobre, A. G. Armada, and S. Muhaidat, "Task scheduling for mobile edge computing using genetic algorithm and conflict graphs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8805–8819, 2020.

[34] Y. Zhang, Y. Liu, J. Zhou, J. Sun, and K. Li, "Slow-movement particle swarm optimization algorithms for scheduling security-critical tasks in resource-limited mobile edge computing," *Future Generation Computer Systems*, vol. 112, pp. 148–161, 2020.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 8800234, 15 pages
https://doi.org/10.1155/2021/8800234

Research Article

Latency-Aware Computation Offloading for 5G Networks in Edge Computing

Xianwei Li and Baoliu Ye

Hohai University, Information Department, School of Computer and Information, Nanjing 21106, China

Correspondence should be addressed to Xianwei Li; lixianwei@njxzc.edu.cn

Received 30 July 2021; Revised 27 August 2021; Accepted 4 September 2021; Published 22 September 2021

Academic Editor: Xuyun Zhang

With the development of Internet of Things, massive computation-intensive tasks are generated by mobile devices whose limited computing and storage capacity lead to poor quality of services. Edge computing, as an effective computing paradigm, was proposed for efficient and real-time data processing by providing computing resources at the edge of the network. The deployment of 5G promises to speed up data transmission but also further increases the tasks to be offloaded. However, how to transfer the data or tasks to the edge servers in 5G for processing with high response efficiency remains a challenge. In this paper, a latency-aware computation offloading method in 5G networks is proposed. Firstly, the latency and energy consumption models of edge computation offloading in 5G are defined. Then the fine-grained computation offloading method is employed to reduce the overall completion time of the tasks. The approach is further extended to solve the multiuser computation offloading problem. To verify the effectiveness of the proposed method, extensive simulation experiments are conducted. The results show that the proposed offloading method can effectively reduce the execution latency of the tasks.

1. Introduction

With the development of wireless communication technology and the Internet of Things (IoT), a variety of emerging applications, such as intelligent access control based on facial recognition, path planning, and virtual reality, meet the needs of people and provide great convenience [1, 2]. However, these applications are usually resource-hungry and delay-sensitive while the physical limitations and the computing power of the mobile devices cannot undertake such applications [3, 4]. Mobile cloud computing is seen as an effective solution to provide computing resources for resource-constrained mobile devices. By offloading computing-intensive tasks to resource-rich cloud data centers for execution, the computing capabilities of mobile devices can be greatly expanded [5, 6].

5G is a major leap in the development of mobile communications [7]. 5G networks deploy ultra-dense distributed networks in small cell infrastructures to provide continuous connectivity [8, 9]. However, this does not mean that the requests of many users can be satisfied at the same time, because most of the computing tasks of applications are deployed in centralized data centers for execution. With the explosive growth of IoT devices in the 5G era, there will be massive computing tasks to be migrated to cloud data centers for executing, causing extreme pressure for the Internet and bringing high network latency. At the same time, the long distance between mobile devices and cloud data centers will also cause unpredictable delays [10–12].

In 5G mobile edge computing, how to offload application services reasonably and content to the edge network is a key issue. Compared with traditional cloud data centers, edge servers are constructed by machines with limited resources and use appropriate strategies to offload services to edge servers [13–15]. In 5G, multiple heterogeneous edge servers can be deployed in the edge network to provide computing services to different users [16, 17]. As a new network paradigm, Software Defined Network (SDN) can realize the logical centralized control of distributed user equipment [18]. Each user equipment transmits its

task-related information to the SDN controller at the beginning that can take appropriate methods from a global perspective to determine where and when to perform these tasks belonging to different users [19, 20].

Considering the increasing complexity of IoT applications in the 5G era, the tasks of a single application are often composed of a series of subtasks. Initially, the task is decomposed into multiple subtasks to support multithreaded processing and improve the efficiency of task execution [15]. Most of the existing research treats tasks as a whole and ignores the connection of internal subtasks. Using subtasks as the unit of computation offloading and performing fine-grained computation offloading, the parallel processing of certain subtasks can be realized, thereby further reducing the delay of tasks [16, 21].

Based on the above observation, a latency-aware computation offloading method is proposed. Through analyzing the dependencies between subtasks, the method rationally arranges scheduling between subtasks and executes fine-grained computation offloading to solve the delay problem caused by increasing computing demands. Specifically, the main contributions of our work are as follows:

(i) Propose a latency-aware computation offloading method in 5G networks which effectively reduces the overall completion time of the tasks.

(ii) Sovle the multiuser computation offloading problem by extending fine-grained computation offloading method with designed algorithm.

(iii) Through extensive simulation experiments, our proposed offloading method can effectively reduce the execution latency of the tasks.

The rest of the paper is organized as follows: Section 2 describes related work. In Sections 3 and 4, the goal of reducing the delay of tasks is raised and the algorithms for computing offloading decision-making are proposed. Experiments are conducted in Section 5. In Section 6, we conclude this paper.

2. Related Work

In recent years, many IoT applications need to be offloaded to the cloud data center for processing. The emergence of 5G has accelerated this process and further increased the demand for computation offloading [22]. To solve this problem, edge computing, as an effective computing paradigm, is widely used in 5G networks for computing offloading. By providing computing and storage resources at the edge of the network, task waiting time is reduced and user experience is better [23–25].

Most of the current research on edge computing offloading focuses on optimizing certain specific goals through reasonable computing offloading strategies. Jararweh [26] proposed a framework based on edge computing, using the expanded computing power of edge computing and 5G to effectively manage and optimize the energy cloud system, while improving its reliability and safety. Li et al. [27] studied Mobile Edge Computing (MEC) of Unmanned Aerial Vehicle (UAV) and maximized energy efficiency of unmanned aerial vehicles to achieve the smallest UAV energy consumption by optimizing UAV trajectory, user transmission power, and computing load distribution. Merluzzi et al. [28] proposed an energy-saving algorithm for dynamic computing offloading in multiaccess edge computing scenarios, using limited block length and reliability constraints to consider Ultra Reliable Low-Latency Communication (URLLC). The proposed algorithm is based on stochastic optimization, which achieves the best balance between service delay and energy spent on mobile devices while ensuring the target probability of service interruption. Yang et al. [29] built a multi-UAV-assisted mobile edge computing system to provide computing offloading services for terrestrial IoT nodes with limited local computing capabilities. To balance the load of UAVs, a multi-UAV deployment mechanism based on Differential Evolution (DE) is proposed. It uses a near-optimal algorithm to solve the decisions of computation offloading. It guarantees coverage constraints and satisfies the IoT node Quality of Service (QoS) while achieving load balancing of these drones.

The 5G network based on edge computing has advantages in effectively offloading large-scale traffic, which is a promising architecture to alleviate the conflict between transmission performance and Quality of Experience (QoE). However, due to the mutual interference between wireless channels in the 5G network, it is difficult to provide satisfactory services to mobile users with existing solutions. Therefore, the optimal method of edge offloading in the 5G network has caused more and more research. Cao et al. [30] proposed a reliable and efficient multimedia service optimization framework. First, a reliable video service mechanism was constructed to help mobile users to distinguish between credible and economical services. Second, an effective wireless resource allocation strategy was established, using the Stackelberg model and other potential game models to achieve low-latency and energy-efficient video service optimization. In addition, Yang proposed a joint optimization scheme for task sharing and resource allocation in a 5G communication network based on edge computing. First, three modes for processing computationally intensive tasks are proposed, including local computing, fuzzy node computing, and edge node computing. For these three computing modes, the problem of computing task offloading is transformed into a joint optimization problem of time and energy consumption, and the authors used the interior point method to solve this problem. Yang [31] studied the computing offloading and subcarrier allocation problems in the MEC system based on multicarrier NOMA and used a deep reinforcement learning method for online computing offloading to solve this problem and greatly improve the computing speed of the MEC system.

For edge computing offloading, latency is a key indicator, and latency-aware edge computing offloading issues have gradually become a current research hotspot [32, 33]. To solve the problem how to generate the best mix of suitable microservices for applications in the mobile edge computing environment, Xia et al. [34] first attempted to study the Data, User, and Power Allocation problem in the edge environment and proposed a two-stage game theory decentralization algorithm to achieve the Nash equilibrium as the

solution, which maximizes the user's overall data rate. Harris et al. [35] defined the problems of virtual network function placement and distribution and provided algorithms with guaranteed performance to realize the placement of delay-sensitive services in appropriate network locations according to the specific needs and related requirements of each service. In response to the need for Mobile edge orchestrator (MEO) to expand capacity on many devices, Nguyen et al. [36] proposed a fuzzy-logic based MEO that separates tasks from mobile devices and maps them to the cloud servers and edge servers, reducing the delay of task processing. Specially, the fuzzy-based MEO was employed to make multi-criteria decision-making which selects the appropriate host to perform tasks by considering multiple parameters in the same framework and find the optimal task segmentation strategy.

Although there was some work dedicated to solving the optimization problem of edge computing offloading in 5G, there is still relatively little work on latency-aware edge computing offloading while network delay is a key requirement for some delay-sensitive programs. Therefore, the delay-aware edge computation offloading method takes delay as the main optimization goal and reduces the total delay of task execution as much as possible to meet the needs of delay-sensitive tasks in 5G.

3. Models and Problem Definition

The delay and energy consumption models of edge computation offloading in 5G network are analyzed in this part, followed by the problem definition. The main symbols used in this section with their descriptions are shown in Table 1.

3.1. The Delay Model. Due to different task migration strategies, the delay of completing the task is also different, so the time delay is computed separately according to different computation offloading methods. Figure 1 illustrates a system framework for edge computing. In this framework, we consider a scenario where "S edge servers as providers of computing resources cover N mobile devices" has been changed to "S edge servers, as providers of computing resources, cover N mobile devices." Each mobile device can execute the task locally according to the specific situation or upload the task to edge servers, but only one migration strategy can be selected for a task.

For tasks executed locally, since data transmission is not performed, the time delay includes only the local execution delay. For the subtask $t_{i,j}$ on the specific user equipment u_i, the calculation method of local execution delay is as follows:

$$T_{i,j}^{\text{local}} = \frac{c_{i,j}}{f_i}, \tag{1}$$

where f_i represents the computing capability of the user equipment u_i.

For tasks that need to be migrated to the edge server, the time delay is divided into three parts: transmission delay, execution delay, and queuing delay. The computation method for the transmission rate $r_{i,j}$ between a certain user equipment u_i and the edge server e_j is as follows:

$$r_{i,s} = B \log_2\left(1 + \frac{h_{i,s}p_i}{\sigma + \sum_{i'=1,a_i=a_i}^{N} h_{i',s}p_{i'}}\right), \tag{2}$$

where B is the channel bandwidth, $h_{i,s}$ represents the channel gain between user equipment i and edge server s, p_i represents the transmission power of user equipment u_i, σ represents the basic noise power of the transmission channel, and $\sum_{I'=1,a_i=a_{I'}}^{N} h_{I',s}p_{I'}$ represents the wireless interference caused by other user equipment that transmits tasks to e_j.

The transmission delay of sending the subtask $t_{i,j}$ from the user equipment u_i to the edge server e_s is as follows:

$$T_{i,j}^{\text{trans}} = d_{i,j} \cdot \frac{1}{r_{i,s}}. \tag{3}$$

Since the computing power of different edge servers is different, the execution time of tasks on different edge servers is also different. The execution delay of tasks $t_{i,j}$ is as follows:

$$T_{i,j}^{\text{exec}} = \frac{c_{i,j}}{f_i^s}, \tag{4}$$

where f_i^s represents the computing power of the edge server e_s.

The queuing delay of task $t_{i,j}$ depends not only on the execution completion time of the predecessor task, but also on the migration strategy of tasks on other devices. Therefore, the queuing delays obtained by various algorithms are different and since the start time and end time of each subtask are not fixed, they change according to the execution of the specific task. Two variables EST and EFT are defined for each subtask to represent the objective function. EST(j, s) represents the earliest execution time when the *j-th* subtask is offloaded to the edge server e_s while EFT(j, s) represents the earliest completion time of the *j-th* subtask in the edge server. The value 0 of EST is set for the first subtask which means that the subtask should be executed on the user's device.

The EST and EFT computations of other subtasks are computed recursively since the first subtask. To compute the EST of a subtask, the offloading strategy of all predecessor tasks of the subtask must have been determined and the computation must base on the completion time of the precursor task. Specifically, the EST of a subtask is as follows:

$$\text{EST}(j,s) = \max\left\{T_s^{\text{avail}}, \max\left(\text{EFT}(j') + C_{j,j'}\right)\right\} (j' \in \text{pred}(j)), \tag{5}$$

where T_s^{avail} represents the time that the edge server e_s is idle, and $C_{j,j'}$ represents the data transmission delay from the subtask $t_{i,j'}$ to $t_{i,j'}$, expressed as

$$C_{j,j_I} = \begin{cases} 0, & a_{i,j} = a_{i,j_I}, \\ T_{i,j}^{\text{trans}}, & \text{otherwise.} \end{cases} \tag{6}$$

After the scheduling strategy of the subtask is determined, the EFT of the subtask is computed according to the execution time of the task:

Table 1: Symbols and corresponding descriptions.

Symbol	Description
$c_{i,j}$	The number of CPU cycles required to execute $t_{i,j}$
$d_{i,j}$	Input data volume of subtask $t_{i,j}$
T_i^{local}	The execution delay of the subtask $t_{i,j}$ in the local execution
$T_{i,j}^{\text{trans}}$	Transmission delay of subtask $t_{i,j}$
$T_{i,j}^{\text{exec}}$	Execution delay of subtask $t_{i,j}$ after migration
f_i	The computing power of the user's device ui
$r_{i,j}$	Transmission rate between u_i and e_j
B	The bandwidth of the transmission channel
$h_{i,j}$	Channel gain between user equipment i and edge server j
$E_{i,j}^{\text{local}}$	Execution energy consumption of subtask $t_{i,j}$ executed locally
$E_{i,j}^{\text{trans}}$	Transmission energy consumption of subtasks
EST	The earliest start time when the subtask gets the execution
EFT	Earliest completion time of subtask execution

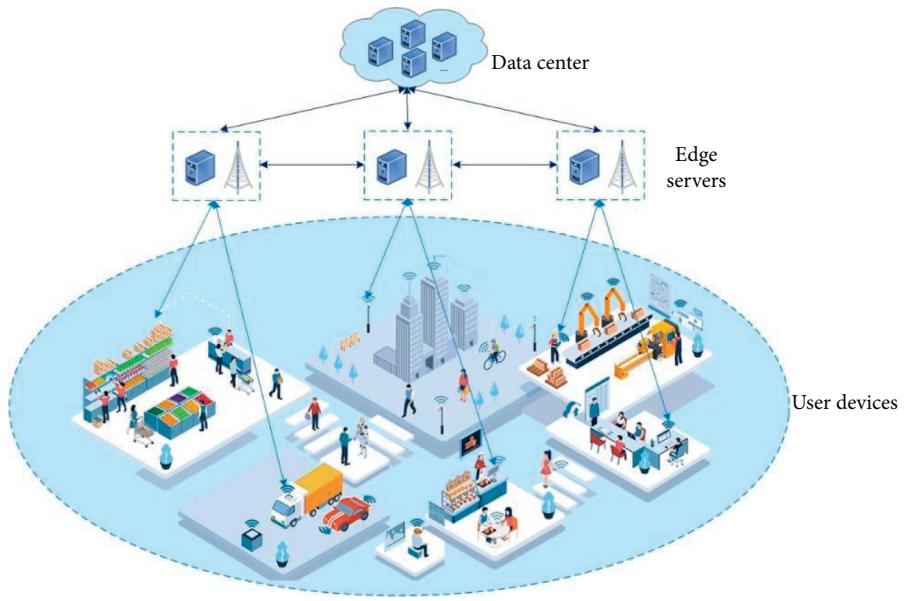


Figure 1: The architecture of computation offloading for 5G networks in edge computing.

$$\text{EFT}(j) = \begin{cases} \text{EST}(j,0) + T_{i,j}^{\text{local}}, & t_{i,j} \text{are executed locally,} \\ \text{EST}(j,s) + T_{i,j}^{\text{exec}}, & \text{otherwise.} \end{cases} \tag{7}$$

For a task t_i, its completion time is determined by the EFT of its last completed task. The computation method is as follows:

$$T_i^{\text{finish}} = \text{EFT}(\text{last}) + T_{\text{last}}^{\text{local}}. \tag{8}$$

Since the last task is usually to collect and process computation results, it is generally executed locally on the user device, and $T_{\text{last}}^{\text{local}}$ is used to represent the execution time of the last task, so as to obtain the end time of the entire task.

3.2. The Energy Consumption Model. The energy consumption of the user equipment u_i mainly includes two parts, which are the execution energy consumption caused by the execution of tasks on the user equipment and the transmission energy consumption of offloading the tasks to the edge server.

If the task $t_{i,j}$ decides to be executed locally, the execution energy consumption is as follows:

$$E_{i,j}^{\text{local}} = \delta_i \cdot c_{i,j}, \tag{9}$$

where δ_i is the energy consumption of per unit CPU cycle of the user equipment u_i.

If task $t_{i,j}$ decides to migrate to the edge server e_s for execution, the corresponding transmission energy consumption is as follows:

$$E_{i,j}^{\text{trans}} = p_i \cdot d_{i,j} \cdot \frac{1}{r_{i,j}}. \tag{10}$$

The total energy consumption of task t_i is determined according to the different migration strategies of each subtask. Set a binary variable flag_j to indicate whether the j-th subtask is to be migrated. The computation method is as follows:

$$\text{flag}_j = \begin{cases} 0, & t_{i,j} \text{ are executed locally,} \\ 1, & t_{i,j} \text{ are executed in edge servers.} \end{cases} \tag{11}$$

The total energy consumption of task t_i is as follows:

$$E_i^{\text{total}} = \sum_{j=1}^{M} \text{flag}_j \cdot E_{i,j}^{\text{trans}} + \left(1 - \text{flag}_j\right) \cdot E_{i,j}^{\text{local}}. \tag{12}$$

3.3. Problem Definition. This research aims to find a set of edge computing offloading strategies to minimize the time delay while meeting the constraints of user equipment on energy consumption. The problem is formalized as

$$\min \sum_{i=1}^{N} T_i^{\text{finish}}, \tag{13}$$

$$\text{s.t. } E_i^{\text{total}} \le L_i, \tag{14}$$

where L_i represents the battery power of the user equipment u_i. Equation (14) indicates that the energy consumption of the user equipment to perform tasks cannot exceed the power of the user equipment.

4. Algorithm Design

In this section, based on the problem of computing offloading in the edge network proposed in the previous chapter, a delay-aware optimization algorithm is proposed. Because the execution effect of the entire task depends on the scheduling strategy of each subtask and the dependencies between the subtasks, a method for selecting the optimal offloading strategy for each subtask is proposed, and then the optimal scheduling strategy for the entire task is obtained. To better cope with the computing offloading needs of multiple users, the proposed computing offloading strategy selection algorithm is further expanded, so that it can solve the problem of computing offloading decision-making in a multitasking environment. At the same time, to cope with the possible delay in the unknown network environment, the proposed method is further improved.

4.1. The Subtask Selection Algorithm. How to choose the most suitable subtasks for computational offloading requires solving the following two problems. First, because different subtasks have different benefits for computing offloading, how to determine which subtasks can be offloaded from the topological graph of the computing task. Second, what kind of subtask offloading combination can provide greater potential performance. Therefore, it is necessary to analyze the topological structure diagram of the subtasks to obtain the opportunity for computing offloading. Due to the interdependence between different subtasks, the first thing to be solved is to sort each subtask to determine the order of scheduling. At the same time, there are multiple parallel subtasks in a task. How to sort these parallel subtasks and determine the scheduling priority also has a certain impact on reducing the delay of the entire task.

Using $d_{a,b}$ represents the amount of data transmitted from subtask $t_{i,a}$ to subtask $t_{i,b}$, and the communication delay $\omega_{a,b}$ between edges (a, b) can be expressed as

$$\omega_{a,b} = \begin{cases} d_{a,b} \cdot \dfrac{1}{r_{a,b}}, & t_a \text{ and } t_b \text{ scheduling strategies are different,} \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Obviously, when two consecutive subtasks are executed at the same place, there is no need for data transmission between two subtasks, so the communication delay is zero. Therefore, the mean value of the communication delay between the two subtasks can be expressed as

$$\overline{\omega}_{a,b} = d_{a,b} \cdot \frac{1}{2r_{i,a}}. \tag{16}$$

Similarly, the average execution delay of a subtask $t_{i,j}$ can be expressed as

$$\overline{T}_{i,j} = \frac{T_{i,j}^{\text{exec}} + T_{i,j}^{\text{local}}}{2}. \tag{17}$$

It is the average of the delays of local execution and migration to edge server execution.

According to these two mean values, the computing method of scheduling priority for a certain subtask $t_{i,j}$ is defined, which is specifically expressed as

$$\text{prior}(j) = \overline{T}_{i,j} + \max\left(\overline{\omega}_{j,j'} + \text{prior}\left(j'\right)\right)\left(j' \in \text{pred}(j)\right), \tag{18}$$

where pred(j) is the predecessor subtask set of $t_{i,j}$. prior(j) represents the scheduling priority of subtask $t_{i,j}$. Obviously, the value of priority(0) for the first subtask of the task is 0. The lower the priority value, the higher the scheduling priority of the subtask.

After determining the scheduling sequence, to select the optimal computation offloading method for each subtask, an algorithm for shortest time-to-completion first of the subtask is proposed which chooses a suitable offloading strategy based on computing volume and its predecessor subtask. Sort from high to low according to the previously computed scheduling priority and select the subtask with the highest scheduling priority among the unscheduled subtasks to process. For each subtask, the EFT executed locally and the EFT offloaded to the edge server according to (7) are computed firstly, and the delay of two schemes and the energy consumption constraint of the user equipment at this time are compared to select the optimal offloading strategy for the subtask.

The Algorithm 1 describes the specific process of selecting a subtask scheduling strategy. The EFP first inputs the directed acyclic graph representing the entire task and the relevant parameters needed to compute the EFT and then computes the values of scheduling priority of all the input subtasks. After all the prior values are obtained, the prior values are sorted in a nonincreasing manner. At this time,

the task with the lowest prior value will be scheduled first (line 2–5). The decision-making process of the specific offloading strategy for a single subtask is shown (line 6–14). By looping through each subtask, the earliest end time EFT of the local execution task and the offloading task to the edge server is computed for each task and according to the value of the two EFTs, decide whether to offload to the edge server for the subtask. At the same time, it is necessary to limit the number of subtasks executed locally to meet the energy consumption constraints of user equipment. The loop is ended when the migration strategy is determined for each subtask, and the output result at this time is the computation offloading strategy of task t_i.

4.2. The Offloading Strategy Selection Algorithm. In 5G network, multiple users often need to offload delay-sensitive tasks to the edge server at the same time to obtain computing resources of the edge server to improve task execution efficiency. The shortest completion time priority algorithm proposed in Section 4.1 is only designed to select the best offloading strategy for a single task. To better cope with the computing offloading needs of multiple users, the computing offloading strategy selection algorithm is further expanded, so that it can solve the problem of computing offloading decision-making in a multitasking environment.

First, because there are multiple edge servers which provide computing resources in a multitasking environment, it is necessary to make decisions on each edge server to select the migration strategy with the least delay. Secondly, in the migration process of multiple users, signal interference will also be brought to the data transmission channel, and a decision must be made in consideration of the interference between users. Considering the above problems, the EFP algorithm is further extended to make it possible to solve the problem of edge computing offloading in a multi-user environment.

For a specific scheduling strategy of a task t_i, specifically for a task with j subtasks, of_i = $\{a_1, a_2, \ldots, a_j\}$, where a_n represents the offloading strategy corresponding to the n-th subtask, if $a_n = 0$, the subtask will be executed locally; otherwise, it will be offloaded to the corresponding edge server. For users who need to compute offloading at the same time, list all possible migration strategies for all users, and then compute the delay currently for each possible migration strategy, and then compare to get the optimal offloading strategy. For a specific migration strategy, first, according to the determined strategy, record the number of tasks that will be offloaded to an edge service, and update some variables that change with the environment during the offloading process, such as the task's transmission rate $r_{i,s.}$. After that, the EFP algorithm proposed in Section 4.1 is called. For a user's task, first the scheduling priority of the subtasks within each task is computed, and then the offloading strategy of the subtasks is determined according to the priority to place the task in local or offloading tasks to edge servers to achieve lower latency. After computing the delay of each subtask, the total delay for a computing offloading strategy can be obtained. After comparing the experiments of all feasible offloading strategies, the computing offloading strategy with the least delay is selected as the output of the algorithm.

Algorithm 2, referred to as MEFP, describes the specific process of edge computing offloading decision-making in a multiuser environment. Firstly, enumerate all possible migration strategies according to the set U of all users to be offloaded at a certain time. Then by using a set to store tasks to be migrated to the same edge server in each migration strategy, all possible migration strategies that convenient to update the transmission interference between different tasks are looped through. Then for each user's specific tasks, the transmission rate is updated according to the network status, and the EFP algorithm is called to decide whether to migrate (line 3–10). After the decision is completed, the total task execution delay under the offloading strategy is obtained. After traversing all possible migration strategies, the migration strategy with the smallest total delay is selected and output as the result (line 11–13).

4.3. The Shortest Completion Time Priority Algorithm. The offloading strategy selection algorithm proposed in Section 4.2 provides the most optimized computing offloading strategy in a multiuser environment. However, when limited resources lead to high resource contention rate, some unreasonable decisions are made due to lack of overall network information. For example, there are many tasks waiting to be executed on the edge server at the same time while the client is not aware and still migrates tasks to the edge servers. Due to the long queuing delay, the execution delay of the entire task may be longer than executed locally, which occurs more frequently when the computing resources are insufficient.

To solve such problems, based on the Carrier Sense Multiple Access (CSMA) used in computer networks, a multiuser shortest completion time priority algorithm in a resource contention environment is proposed. In the CSMA algorithm, instead of directly sending data packets when detecting that the channel is idle, the sender refuses to send with a certain probability to avoid conflicts. Similar strategies are adopted to avoid conflicts in computing migration requirements when multiple tasks compete for edge server computing resources. After the comparison of execution time of processing task locally and migrating task to the edge server, the task migration is rejected with a certain probability. Because executing the task locally can prevent multiple tasks from waiting for the computing resources, reducing queuing delay and task execution delay.

The computation method of the transfer probability possibility(j) of subtask $t_{i,j}$ can be expressed as

$$\text{possibility}(j) = \frac{T_{i,j}^{\text{local}} - \left(T_{i,j}^{\text{exec}} + T_{i,j}^{\text{trans}}\right)}{T_{i,j}^{\text{queue}} + \tau}, \tag{19}$$

where $T_{i,j}^{\text{queue}}$ represents the waiting time at the edge server, τ represents a small time constant.

When the difference between the delay of local execution and the delay of task migration execution is larger than the

Inputs: $G = (T, DP)$, M, $r_{i,j}$, f_i, f_i^s, L_i
Output: Decision (i)
(1) $i = 0$
(2) **for** $i = 0$ **to** M **do**
(3) Compute the prior values of each subtask $t_{i,j}$ according to formula (17)
(4) **end for**
(5) non-increasing sorting of the prior values of all subtasks
(6) **while existing subtasks that have not confirmed scheduling strategies**
(7) Select the subtask $t_{i,j}$ with the highest priority
(8) Computing the EFT(j) of the task according to formula (7)
(9) **if** $\text{EFT}(j)_{\text{local}} < \text{EFT}(j)_{\text{edge}}$ **and** $E_i^{\text{totals}} < L_i$
(10) The subtask $t_{i,j}$ still executed locally
(11) **else**
(12) Offload $t_{i,j}$ to edge server
(13) **end if**
(14) **end while**

Algorithm 1: Shortest time-to-completion first EFP.

Inputs: E, U, $G_i = (T_i, DP_i)$, M, $r_{i,s}$, f_i, L_i
Output: Optimal offloading strategy $of_{\min}$ for multiusers
(1) List all optional offloading strategies $\text{OF} = \{of_1, of_2, \ldots, of_N\}$
(2) **for** of_i in OF **do**
(3) **for** a_j in of_i **do**
(4) **for** e_k in E **do**
(5) Create a set U_k for each users with $a_j = e_k$
(6) Update the transfer rate $r_{i,s}$ of each tasks
(7) **end for**
(8) **for** t_i in U_k **do**
(9) Call EFP algorithm
(10) **end for**
(11) **end for**
(12) Compute the total execution delay T_i^{finish} for offloading strategy of_i
(13) **end for**
(14) Select the offloading strategy $of_{\min}$ with minimal execution delay

Algorithm 2: Shortest time-to-completion first with multiusers MEFP.

waiting delay, the migration probability value at this time is at most 1 when the task must be offloaded to the edge server. On the contrary, when the local execution delay is less than the execution delay of the task migration, the migration probability value is 0 when the task is executed locally.

The migration probability of a task changes linearly with the waiting delay and the difference between the local execution delay and the migration execution delay. The larger the waiting delay of the last execution, the more congested the network conditions at this time, at which case the probability of task being executed locally is larger than being executed in edge servers. When the waiting delay is small, the probability of the task being computed and offloaded is greater. When the computing task is simple and the performance difference between the local and offloading to the edge server is not big, the computing task will have a greater probability to be executed locally to avoid the situation where multiple tasks are waiting to be executed at the edge server at the same time. When the computing task is more complex and the performance on the edge server is significantly better than the local execution, the computing task will be offloaded to the edge server. For the subtasks that use the EFP algorithm to determine the offloading strategy and need to perform computing offloading, it is necessary to further determine whether the task should be offloaded to the edge server according to the calculated probability.

Algorithm 3 describes the execution process of the multiuser shortest completion time priority algorithm in a resource contention environment. All possible migration strategies are enumerated and the EFP algorithm is employed to make decisions. Then the probability of selecting migration is computed at this time (line 1–8). If the random number generated is greater than the probability, the migration is rejected, and the task is executed locally. Otherwise, the task is still offloaded to the edge server (line 9–17). At the end of the algorithm, according to the delay of

Inputs: E, U, $G_i = (T_i, \mathrm{DP}_i)$, M, $r_{i,s}$, f_i, L_i,
Output: Optimal offloading strategy $\mathrm{of}_{\min}$ for multiusers
(1) List all optional offloading strategies $\mathrm{OF} = \{\mathrm{of}_1, \mathrm{of}_2, \ldots, \mathrm{of}_N\}$
(2) **for** of_i in OF **do**
(3) **for** a_j in of_i **do**
(4) **for** e_k in E **do**
(5) Create a set U_k for each user with $a_j = e_k$
(6) **for** t_i in U_k **do**
(7) Update the transfer rate $r_{i,s}$ of each tasks
(8) Call EFP algorithm
(9) **if** Decision(i, j) ! = 0
(10) Compute migration probability possibility(j)
(11) random = Random(0, 1)
(12) **if** random ≥ possibility(j) **and** < L_i
(13) Execute $t_{i,j}$ locally
(14) **else**
(15) Offload $t_{i,j}$ to edge server
(16) **end if**
(17) **end if**
(18) **end for**
(19) **end for**
(20) **end for**
(21) Compute the total execution delay T_i^{finish} for computing offloading strategy of_i
(22) **end for**
(23) Select the offloading strategy $of_{\min}$ with minimal execution delay

Algorithm 3: Shortest Time-to-Completion First with multiusers under low resource p-MEFP.

the corresponding computing offloading strategy, the offloading strategy with the smallest delay is obtained as the output result for all users who seek to computation offloading (line 20–22).

5. Experiment Evaluation

A series of experiments to simulate the process of multiuser edge computing offloading in 5G are carried out to verify the effectiveness of the proposed method. Firstly, experimental configuration like parameter settings is introduced, and then comparison schemes are selected to simulate the environment under different number of tasks and edge servers. Through comparison, the advantages of the proposed p-MEFP algorithm are shown obviously.

5.1. Experimental Configuration. The experiment simulates an environment with multiple edge servers that can provide computing resources at the same time, and there are multiple user devices randomly distributed around these edge servers in the network environment, and each user device has a set of tasks that consist of several subtasks; a single-edge server can perform multiple subtasks at the same time according to its own computing capabilities. To verify the feasibility of the proposed method for multiple tasks, the directed acyclic graph of the task is randomly generated within a certain range. The size of each subtask is randomly generated in [50 KB, 1000 KB], and the required CPU cycles vary randomly from 50 M cycles to 1000 M cycles. The specific parameters and corresponding values in the experiment are listed in Table 2.

To achieve comparison, another two-edge computing offloading algorithms implemented are introduced as follows:

(1) Benchmark: For each subtask in the task, according to the order of execution, all tasks are migrated to the edge server for execution, the task is not executed locally, and finally the execution structure of the task is transmitted back to the user device.
(2) CEFO1 [38] is an SDWN-based edge computing offloading method. Through the task data uploaded by each user device, the SDWN central controller determines the specific migration strategy for each task. First, enumerate all the optional offloading decisions. For each offloading scheme, the task graph of users which are offloaded to the same server is regarded as an integrated DAG graph through the combination of graphs, and the delay of each different offloading scheme is computed, and finally the offloading scheme with the minimum waiting time is selected as the offloading strategy.

5.2. Experiments Results. In this section, the performance of the proposed p-MEFP algorithm and the other two comparison algorithms Benchmark and CEFO are compared in detail from different user numbers and different edge servers, showing the effectiveness of the three methods in reducing execution delays. At the same time, compare the effectiveness of the p-MEFP algorithm for different task types. The experimental results are shown in Figures 1–9.

Table 2: Parameters and values.

Parameters	Values
Basic noise power of the transmission channel σ[40]	100 dBm
Task transmission power p[40]	150 mW
CPU frequency of edge server f_i^s[40]	20 GHZ
CPU frequency of user equipment f_i [40]	10 GHZ
Task execution power [37]	650 mW

5.2.1. Performance under Different Number of Users. This part compares the average task delay of each method after the simulation experiment of Benchmark, CEFO and p-MEFP under different resource contention environments. By changing the layout of the number of edge servers in the network environment and adjusting the capacity of each edge server, the network environment is set to a high contention environment (the number of edge servers is 2, and each edge server can perform at most 1 task), medium contention environment (the number of edge servers is 5, and each edge server can perform up to 2 tasks), and low contention environment (the number of edge servers is 8, and each edge server can perform up to 4 tasks).

Figure 2 shows the comparison of the average task queuing delay of different methods in a high resource contention environment. In an environment of high resource contention, as the number of tasks increases, the queuing delay is increasing rapidly, and the queuing delay gap of the method is obvious. When the number of tasks is 30, the maximum difference is 60 ms, and when the number of tasks is 50, the average queuing delay gap is up to 100 ms, indicating that the p-MEFP method can reduce the queuing delay of the task well.

Figure 3 shows the average task delay of the three algorithms under different number of tasks in a high contention environment. In a high contention environment, the number of edge servers that can provide computing resources is relatively small, and the use of edge servers for tasks is more obvious. It can be reflected from the figure that when the number of tasks is small, the execution effect of the three methods is similar, and the p-MEFP method is only slightly better. With the continuous increase in the number of tasks, the situation of users competing for edge servers becomes more and more serious, and the queuing delay accounts for an increasing proportion of the total delay. As the gap in queuing delay becomes larger, the average task delay difference of the three methods gradually becomes larger. When the number of tasks is 10, the difference between optimal and worst performance is only 10 ms. When the number of tasks is 30, the difference is 40 ms, but when the number of tasks is 50, the average delay of p-MEFP is reduced by nearly 70 ms compared to CEFO and is 80 ms less than Benchmark. It is extremely effective in reducing delay, and it significantly reduces execution delay. Through comparison, as the number of users continues to increase, the performance of p-MEFP gets better, which shows the effectiveness of p-MEFP in reducing task delay in a high resource contention environment. At the same time, with the increase in the number of tasks, the increase in delay is very fast. For every 10 additional tasks in p-MEFP, the increase in delay is within 100 ms, while for Benchmark, the increase in delay even reaches nearly 130 ms. Finally, when the number of tasks is 50, the delay of p-MEFP is 383 ms, while the delay of the other two methods is about 450 ms. In comparison, p-MEFP greatly improves the execution effect of the task and reduces the task delay. Comparing Figures 1 and 2, the largest proportion of the task delay at this time is the queuing delay, and p-MEFP can greatly reduce the queuing delay, and thus has a better delay performance.

Figure 4 shows the comparison of the average queuing delay of different methods in the medium resource contention environment. Initially, the queuing delay of the three methods is similar. When the number of tasks reaches 40 and even more, p-MEFP can reduce the queuing delay of nearly 20 ms and 30 ms compared with the other two respectively and has a great advantage in reducing the queuing delay. It proves that our method can achieve good results under the pressure of a large number of tasks.

Figure 5 shows the effectiveness of Benchmark, CEFO and p-MEFP in reducing the average task delay in a medium resource contention environment. In the case of medium resource contention, when the number of tasks is small, the difference between the three is only 6 ms, and as the number of tasks continues to increase, the difference gradually becomes larger, but the largest difference is only about 20 ms, but in all scale tasks, p-MEFP still has certain advantages. As the number of tasks continues to increase, the advantages of p-MEFP are becoming more and more obvious. In terms of the value of delay, the execution delay of tasks in a medium resource contention environment is significantly less than that in a high resource contention environment, and as the number of tasks increases, the rate of increase in delay is relatively stable, and the increase rate remains within 40 ms. And at this time, the queuing delay still accounts for a large proportion of the total execution delay of the task, so the reduction of the queuing delay can still greatly improve the execution effect of the task and reduce the execution delay of the task.

Figure 6 shows the comparison of the effectiveness of the three methods in reducing task queuing delay in a low resource contention environment. When the computing resources in the environment are abundant because there is more space in the edge server, it can accommodate more tasks to be executed at the same time. Currently, the queuing delay accounts for a relatively small proportion of the total delay and the impact of different migration strategies on the delay Smaller. For queuing delay, p-MEFP has no advantage in queuing delay due to the large number of idle edge services at the beginning, but the difference is also about 1 ms. As the number of tasks continues to increase, the

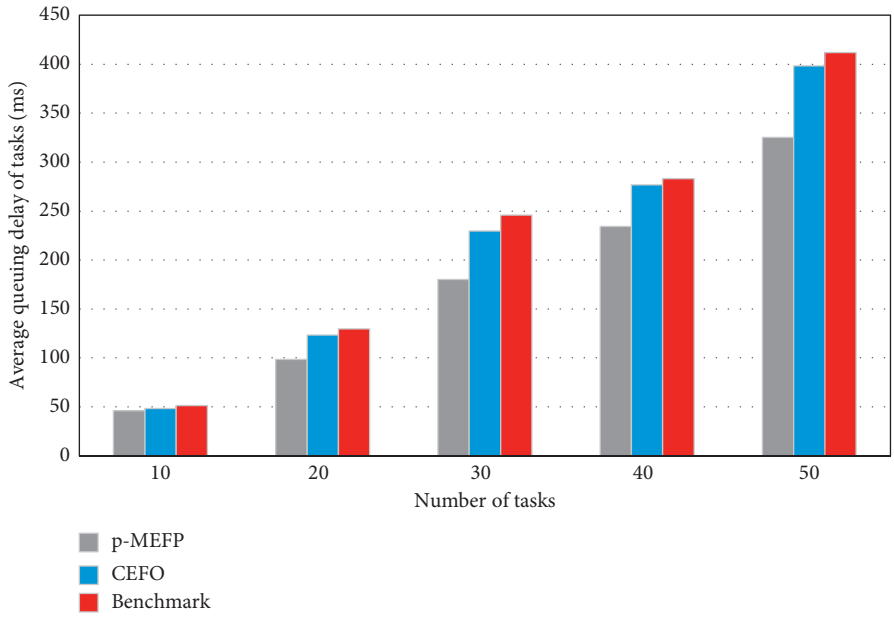


Figure 2: Comparison of the average queuing delay in a high resource contention environment.

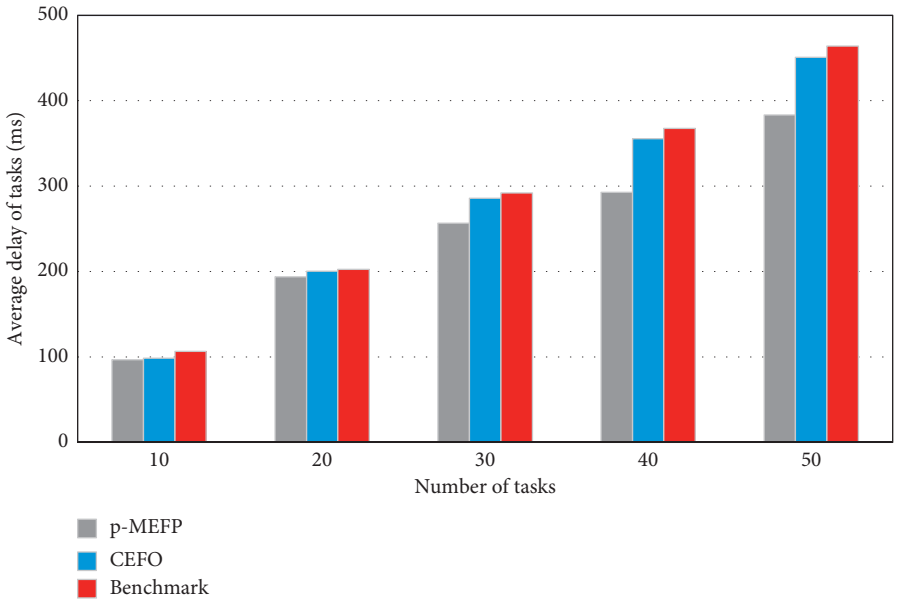


Figure 3: Comparison of the average delay in a high resource contention environment.

average queuing delay is significantly lower than the other two methods where tasks use the p-MEFP method.

Figure 7 is a comparison of the effectiveness of the three methods in reducing the average total delay in a low resource contention environment. Obviously, as the number of tasks increases, the average delay of tasks increases, which is caused by a large number of tasks queuing at the edge. When the number of tasks is low, p-MEFP can reduce task delay, but has no obvious advantage compared to other methods. Because in the case of sufficient resources and few tasks, the delay will be small. As the number of tasks increases, the effectiveness of p-MEFP gradually exceeds the other two methods.

5.2.2. Performance under Different Number of Edge Servers. This part compares the average task delay of the three algorithms of Benchmark, CEFO, and p-MEFP under different edge server numbers. The comparison of the average task delay of Benchmark, CEFP, and p-MEFP with different edge server numbers is shown in Figure 8.

When the number of edge servers is 2, the delay of p-MEFP is less than 400 ms, while the delays of the other two comparison methods are more than 450 ms. The effectiveness of p-MEFP is more obvious. It can reduce the delay of nearly 100 ms compared with Benchmark and nearly 50 ms compared with CEFO. When the number of edge servers is small, p-MEFP has a 30–50 ms advantage over the other two

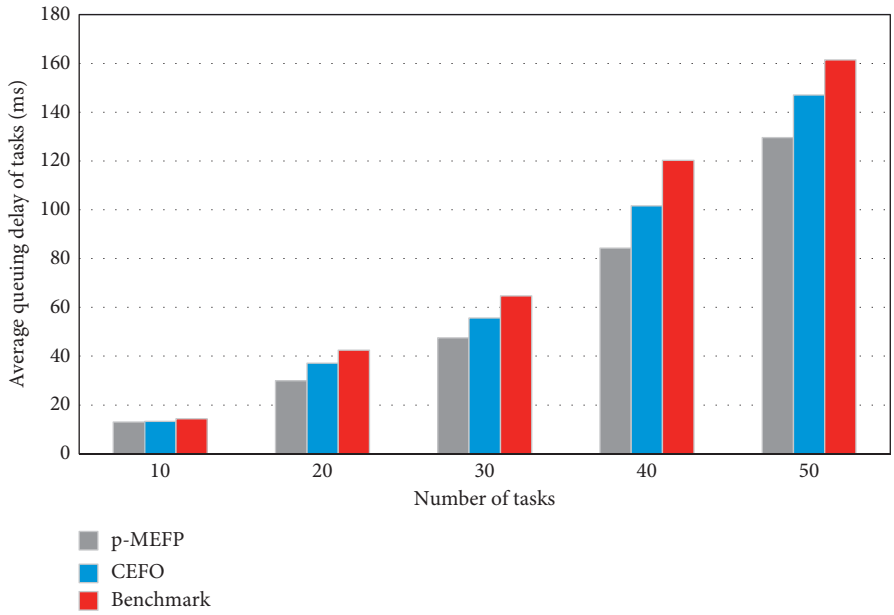


Figure 4: Comparison of the average queuing delay in a medium resource contention environment.

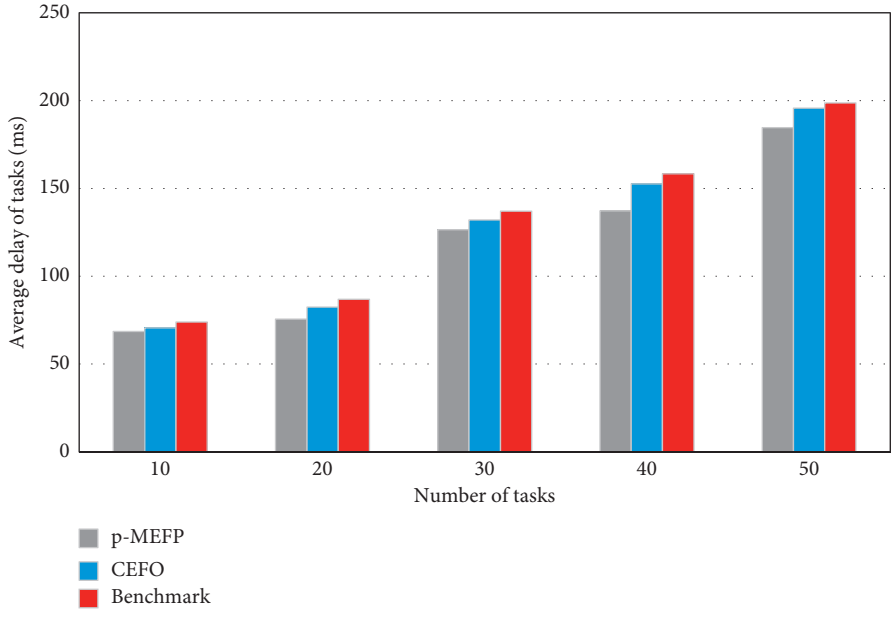


Figure 5: Comparison of the average delay in a medium resource contention environment.

methods. With the increase in the number of edge servers, the advantages of p-MEFP gradually decrease, but the overall p-MEFP has a certain optimization effect compared with the other two methods, which can reduce the average delay of task execution. When the number of edge servers is 5, the task execution delay of p-MEFP is 17 ms and 22 ms less than the other two methods, respectively. For the task execution delay of less than 200 ms at this time, the delay reduction effect is still obvious. After the number of edge servers is further increased to 8, the computing resources are sufficient at this time, and the requirements of computation offloading can be fully met, which is nearly 300 ms lower than when the number of edge servers is 2. The average task execution delay obtained by several comparison methods is not much different, and they are all reduced to about 120 ms. The p-MEFP method only reduces the execution delay by about 7 ms compared with other methods.

On the whole, p-MEFP still maintains its effectiveness in reducing latency, and it can be seen that the number of edge servers has a great impact on the latency of task execution.

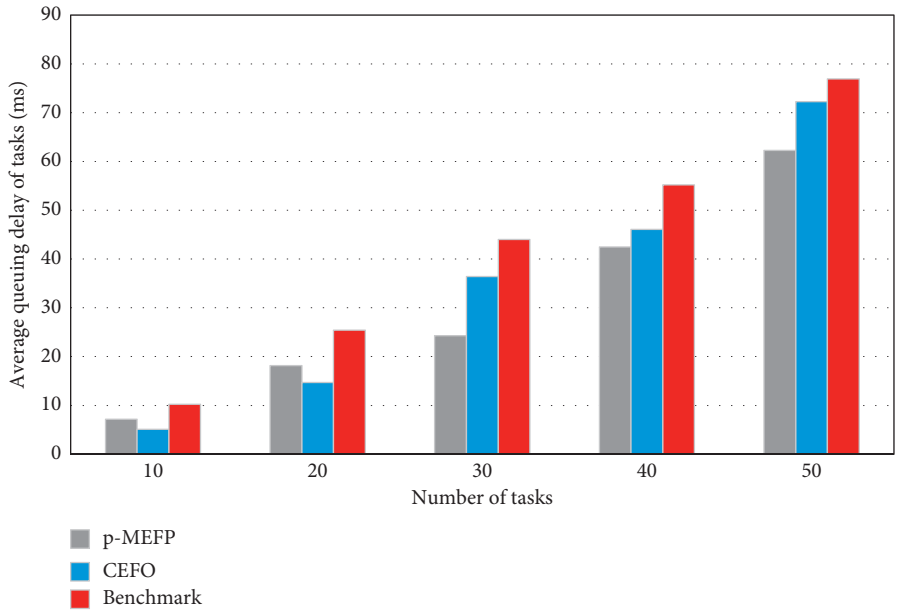


Figure 6: Comparison of the average queuing delay in a low resource contention environment.

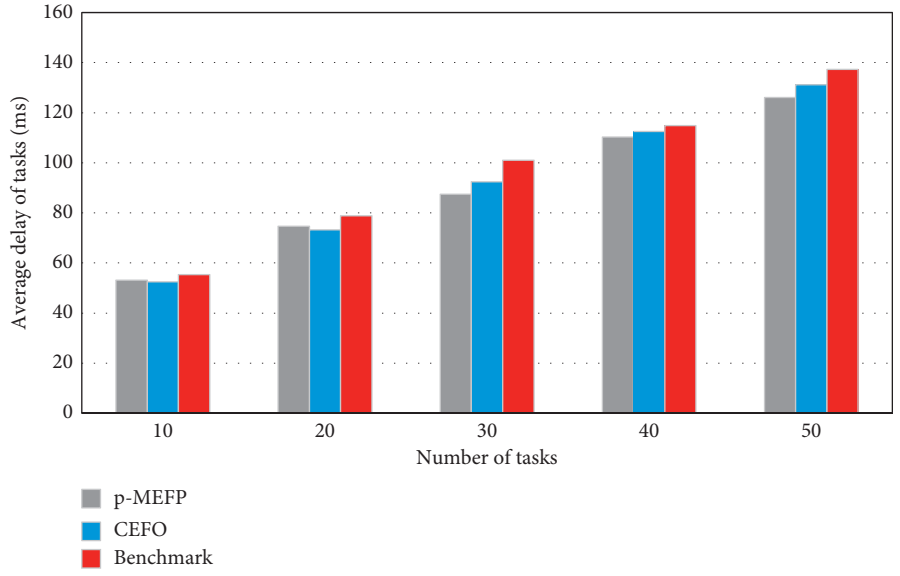


Figure 7: Comparison of the average delay in a low resource contention environment.

With the continuous increase of edge servers, the average execution latency of tasks will be reduced to one-third of the original.

5.2.3. Performance of Directed Acyclic Graphs for Different Tasks. Since the tasks discussed in this article may be decomposed into multiple subtasks, and the parallelism of the subtasks will have a certain impact on the execution effect of the task, in this section, experiments are carried out on different task directed acyclic graphs to compare the differences that the parallelism of the subtasks on the task execution effect. Due to the specific discussion of the directed acyclic graph of the task, five specific tasks with inconsistent parallelism were selected for experiments. The directed acyclic graph of the five tasks is shown in Figure 9. Task type 1 is the serial execution of five subtasks, and each subtask must wait for the completion of its predecessor task. Subtask 2 and subtask 3 of task type 2 can be executed in parallel, and subtask 4 can be executed only after they are all completed. Subtask 2, subtask 3, and subtask 4 of task type 3 can all be executed in parallel, and subtask 5 can only be executed after all three subtasks are completed. Task type 4 and task type 5 are similar; in that

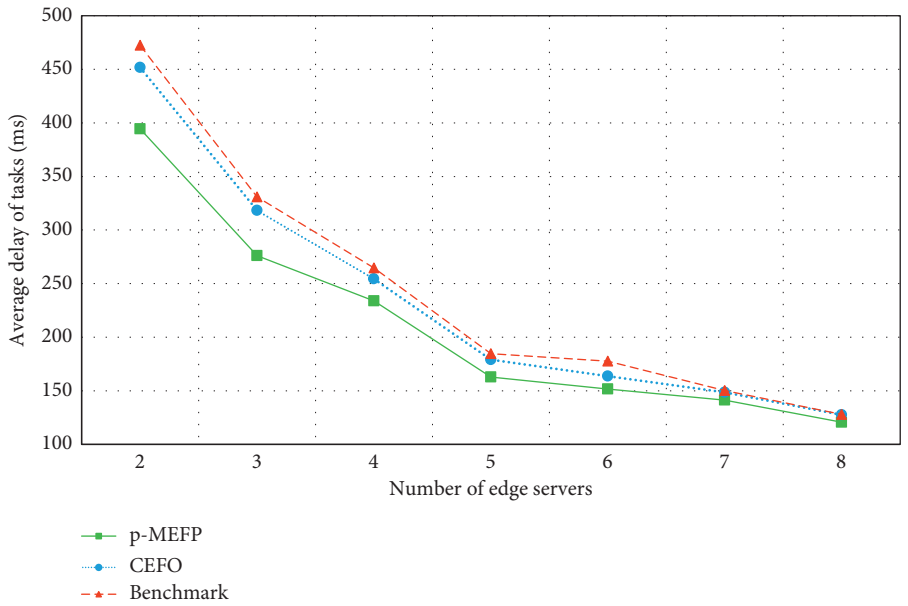


Figure 8: Comparison of the average delay of three algorithm with different number of edge servers.

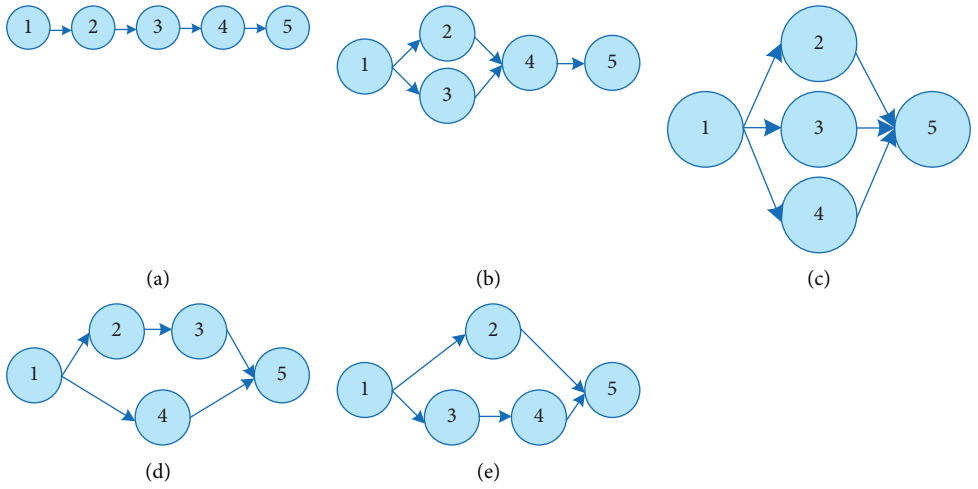


Figure 9: Directed acyclic graph participating in computing offloading task. (a) Task type 1. (b) Task type 2. (c) Task type 3. (d) Task type 4. (e) Task type 5.

one subtask can be executed in parallel with two other serial subtasks. For these five types of tasks, five groups of tasks with the same amount of computing tasks but different task topologies are selected, and the execution results of these five groups of tasks are compared separately to reflect the execution effects of different methods on tasks with different topologies.

Figure 10 shows the comparison of the average task delays obtained after three methods are used to compute and offload a set of tasks with several 20 different directed acyclic graphs in the same network environment. From the figure, it can be clearly seen that the task execution effect of task type 3 is significantly better than other task types, and the task with the highest task execution delay is the task of task type 1. For the task of task category 1, the 5 subtasks can only perform serial work, so the execution delay is the highest. In task type 3, up to 3 subtasks can be processed in parallel at the same time. By migrating the parallel processing tasks to different edge computing servers, the computing tasks of the three subtasks can be processed at the same time. It can be seen from the experimental results that the higher the degree of parallelism of the task, the more obvious the optimization effect after computing offloading, and the lower the delay obtained.

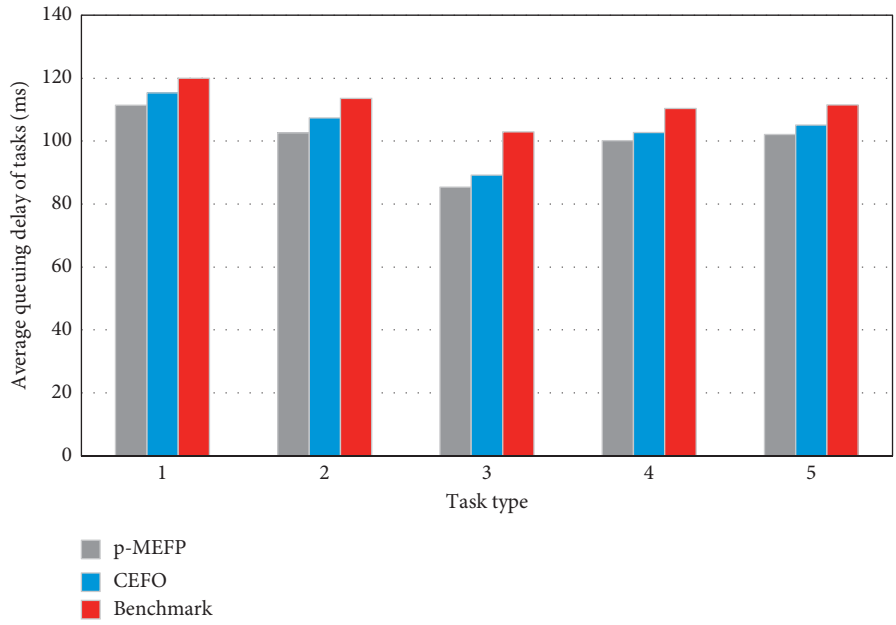


Figure 10: The average delay of computing offloading for different task types.

6. Conclusion

In this paper, the delay and energy consumption of edge computing offloading in the 5G network are analyzed firstly, according to which the goal of minimizing task delay has been proposed. A delay-aware offloading strategy, reducing the overall completion time of IoT applications by decomposing a computing task into several subtasks, is proposed which is expanded for multiuser situations. At the same time, the algorithm has been optimized for possible resource contention. To verify the performance of the proposed method, simulation experiments have been carried out. The results have shown that compared with the existing work, the proposed work can effectively reduce the overall task delay.

Data Availability

The basic data included in this study are provided in the supplementary information files.

Conflicts of Interest

The authors declare no conflicts of interest.

Supplementary Materials

The task requests data used in this study are stored in six files in the supplementary materials, which have the same format, and the number ranges from 50 to 300, respectively. In detail, the first to fifth columns in each piece of data are the task id, workflow id, start time, end time, and path length in sequence. (*Supplementary Materials*)

References

[1] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in IoT edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1133–1146, 2020.

[2] C. Shu, Z. Zhao, Y. Han, and M. Geyong, "Multi-user offloading for edge computing networks: a dependency-aware and latency-optimal approach," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1678–1689, 2019.

[3] Z. Chang, L. Liu, X. Guo, and S. Quan, "Dynamic resource allocation and computation offloading for IoT fog computing system," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3348–3357, 2020.

[4] A. Samanta and Z. Chang, "Adaptive service offloading for revenue maximization in mobile edge computing with delay-constraint," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3864–3872, 2019.

[5] J. Zheng, Y. Cai, Y. Wu, and X. S. Shen, "Dynamic computation offloading for mobile cloud computing: a stochastic game-theoretic approach," *IEEE Transactions on Mobile Computing*, vol. 18, no. 4, pp. 771–786, 2018.

[6] X. Xu, Z. Fang, J. Zhang et al., "Edge content caching with deep spatiotemporal residual network for IoV in smart city," *ACM Transactions on Sensor Networks*, vol. 17, no. 3, pp. 1–33, 2021.

[7] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[8] Z. Ning, K. Zhang, X. Wang et al., "Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, 2020.

[9] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, 2017.

[10] Z. Ning, P. Dong, X. Wang et al., "Mobile edge computing enabled 5G health monitoring for Internet of medical things: a decentralized game theoretic approach," *IEEE Journal on*

Selected Areas in Communications, vol. 39, no. 2, pp. 463–478, 2020.

[11] H. Yang, Y. Liang, J. Yuan, Q. Yao, A. Yu, and J. Zhang, "Distributed blockchain-based trusted multidomain collaboration for mobile edge computing in 5G and beyond," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 7094–7104, 2020.

[12] Y. Liu, M. Peng, G. Shou, Y. Chen, and S. Chen, "Toward edge intelligence: multiaccess edge computing for 5G and internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6722–6747, 2020.

[13] Y. Zhai, T. Bao, L. Zhu, M. Shen, X. Du, and M. Guizani, "Toward reinforcement-learning-based service deployment of 5G mobile edge computing with request-aware scheduling," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 84–91, 2020.

[14] X. Xu, Q. Huang, Y. Zhang, S. Li, L. Qi, and W. Dou, "An LSH-based offloading method for IoMT services in integrated cloud-edge environment," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 3s, pp. 1–19, 2021.

[15] Y. Liu, S. Wang, Q. Zhao et al., "Dependency-aware task scheduling in vehicular edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4961–4971, 2020.

[16] L. Chen, J. Wu, J. Zhang, H. N. Dai, X. Long, and M. Yao, "Dependency-aware computation offloading for mobile edge computing with edge-cloud cooperation," *IEEE Transactions on Cloud Computing*, p. 1. In press, 2020.

[17] M. Wang, T. Ma, T. Wu, C. Chang, F. Yang, and H. Wang, "Dependency-aware dynamic task scheduling in mobile-edge computing," in *Proceedings of 2020 16th international conference on mobility, sensing and networking (MSN)*, pp. 785–790, IEEE, Tokyo, Japan, December 2020.

[18] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive transmission optimization in SDN-based industrial internet of things with edge computing," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1351–1360, 2018.

[19] X. Xu, D. Zhu, X. Yang, S. Wang, L. Qi, and W. Dou, "Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–17, 2021.

[20] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: a survey, use cases, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2359–2391, 2017.

[21] J. Yan, S. Bi, Y. J. Zhang, and M. Tao, "Optimal task offloading and resource allocation in mobile-edge computing with inter-user task dependency," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 235–250, 2019.

[22] X. Liu, J. Yu, J. Wang, and Y. Gao, "Resource allocation with edge computing in IoT networks via machine learning," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3415–3426, 2020.

[23] B. Shen, X. Xu, L. Qi, X. Zhang, and G. Srivastava, "Dynamic server placement in edge computing toward internet of vehicles," *Computer Communications*, vol. 178, pp. 114–123, 2021.

[24] P. Zhou, K. Shen, N. Kumar, Y. Zhang, M. M. Hassan, and K. Hwang, "Communication-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Internet of Things Journal*, vol. 99, p. 1, 2020.

[25] R. S. Pereira, D. D. Lieira, M. A. C. D. Silva et al., "RELIABLE: resource allocation mechanism for 5G network using mobile edge computing," *Sensors*, vol. 20, no. 19, p. 5449, 2020.

[26] Y. Jararweh, "Enabling efficient and secure energy cloud using edge computing and 5G," *Journal of Parallel and Distributed Computing*, vol. 145, pp. 42–49, 2020.

[27] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, "Energy-efficient UAV-assisted mobile edge computing: resource allocation and trajectory optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3424–3438, 2020.

[28] M. Merluzzi, P. D. Lorenzo, S. Barbarossa, and V. Frascolla, "Dynamic computation offloading in multi-access edge computing via ultra-reliable and low-latency communications," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 342–356, 2020.

[29] L. Yang, H. Yao, J. Wang, C. Jiang, A. Benslimane, and Y. Liu, "Multi-UAV-Enabled load-balance mobile-edge computing for IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6898–6908, 2020.

[30] T. Cao, C. Xu, J. Du et al., "Reliable and efficient multimedia service optimization for edge computing-based 5G networks: game theoretic approaches," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1610–1625, 2020.

[31] S. Yang, "A joint optimization scheme for task offloading and resource allocation based on edge computing in 5G communication networks," *Computer Communications*, vol. 160, pp. 759–768, 2020.

[32] Z. Zhu, G. Han, G. Jia, and L. Shu, "Modified DenseNet for automatic fabric defect detection with edge computing for minimizing latency," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9623–9636, 2020.

[33] H. Tian, X. Xu, T. Lin et al., "DIMA: distributed cooperative microservice caching for internet of things in edge computing by deep reinforcement learning," *World Wide Web*, pp. 1–24, 2021.

[34] X. Xia, F. Chen, Q. He et al., "Data, user and power allocations for caching in multi-access edge computing," *IEEE Transactions on Parallel and Distributed Systems*, p. 1, 2021.

[35] D. Harris, J. Naor, and D. Raz, "Latency aware placement in multi-access edge computing," in *Proceedings of 2018 4th IEEE conference on network softwarization and workshops (NetSoft)*, pp. 132–140, IEEE, Montreal, Canada, June 2018.

[36] V. D. Nguyen, T. T. Khanh, T. Z. Oo, N. H. Tran, E. N. Huh, and C. S. Hong, "Latency minimization in a fuzzy-based mobile edge orchestrator for IoT applications," *IEEE Communications Letters*, vol. 25, no. 1, pp. 84–88, 2020.

[37] Y. Han, Z. Zhao, J. Mo, C. Shu, and G. Min, "Efficient task offloading with dependency guarantees in ultra-dense edge networks," in *Proceedings of 2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, Waikoloa, HI, USA, December 2019.

[38] C. Shu, Z. Zhao, Y. Han, and G. Min, "Dependency-aware and latency-optimal computation offloading for multi-user edge computing networks," in *Proceedings of 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1–9, IEEE, Boston, MA, USA, June 2019.