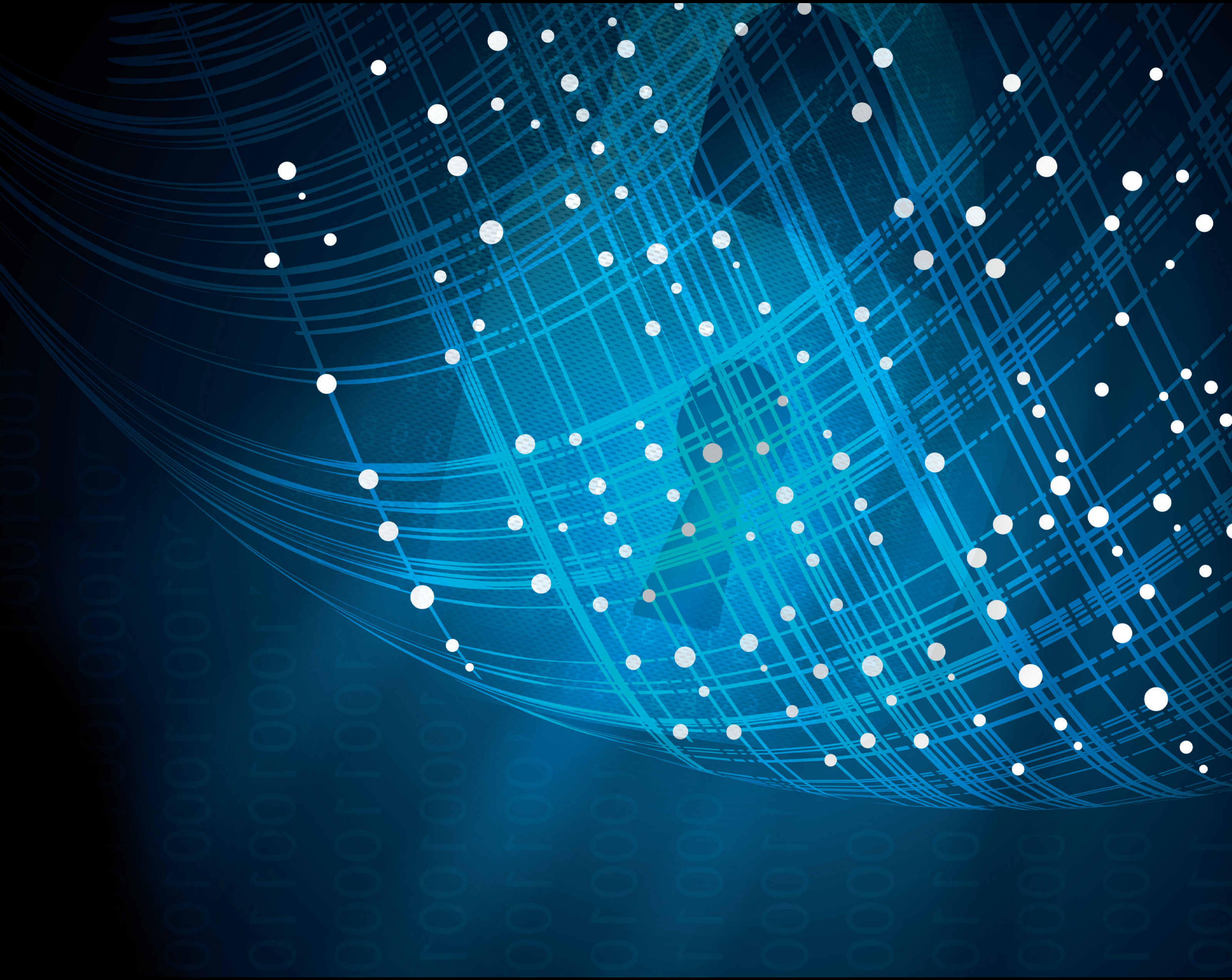
Security and Communication Networks

Security and Privacy Challenges in Internet of Things and Mobile Edge Computing 2021

Lead Guest Editor: Jinbo Xiong
Guest Editors: Qi Li, youliang tian, and Qing Yang



Security and Privacy Challenges in Internet of Things and Mobile Edge Computing 2021

Security and Communication Networks

Security and Privacy Challenges in Internet of Things and Mobile Edge Computing 2021

Lead Guest Editor: Jinbo Xiong
Guest Editors: Qi Li, youliang tian, and Qing Yang

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Security and Communication Networks
Volume 2022, Article ID 6754744, 13 pages
https://doi.org/10.1155/2022/6754744

Research Article

Privacy-Preserving Task Distribution Mechanism with Cloud-Edge IoT for the Mobile Crowdsensing

Liquan Jiang and Zhiguang Qin

School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Correspondence should be addressed to Liquan Jiang; helenjlq@hotmail.com

Received 21 February 2022; Revised 16 May 2022; Accepted 29 August 2022; Published 26 September 2022

Academic Editor: Jinbo Xiong

Mobile crowdsensing under big data provides an efficient, win-win, and low-budget data collection solution for IoT applications such as the smart city. However, its open and all access scenarios raise the threat of data security and user privacy during task distribution of mobile crowdsensing. To eliminate the above threat, this paper first designs a privacy-preserving task distribution scheme (Scheme 1), which realizes fine-grained access control and the practical keyword search, as well as protects the access policy. But it incurs expensive computational and communication consumptions for the task performer side. In this regard, we construct Scheme 2 to attain a lightweight trapdoor generation and keyword search mechanism, and it enables the crowdsensing platform to predecrypt a ciphertext without revealing any information about the task and the performer's privacy. Then, the resource-constrained device on the task performer side can recover the task with a few computational and communication overheads. The security of the scheme has been detailedly proved and analyzed, and theoretical comparisons and experiment demonstrate their practicability.

1. Introduction

The Internet of Things (IoT) [1, 2] paradigm realizes timely response to events and real-time collection and processing of huge amounts of data by connecting a large number of intelligent sensing devices with communication, storage, and computing capabilities through wireless sensor networks (WSN) [3]. Benefiting from the distributed network architecture and the potential raised by massive data, IoT is expected to promote innovation and development in many fields, improve user experience, and explore higher management levels. More specifically, as a public service blueprint supported by big data [5], many fields of smart city construction (city governance, smart transportation [4], smart medical care [6, 7], for instance) are expected to benefit greatly from the deployment of IoT. In addition to relying on the widely deployed sensing devices (including sensors, surveillance cameras, and GPS devices) in urban to monitor and collect massive amounts of data in real time, by introducing the mobile crowdsensing schema, residents are encouraged to actively participate in city governance and use their smart mobile devices (such as smartphone) to capture and upload events that are hard to detect, is considered to be a low-cost and emerging trend in the IoT-oriented smart city construction [8].

The mobile crowdsensing systems can be categorized as "participatory" and "opportunistic" according to task allocation strategy [8]. In a typical opportunistic mobile crowdsensing system [9], the system can adaptively assign optimal sensors to collect sensing data based on the operational scenario. This strategy guarantees the efficiency and accuracy of data collection, but at the cost of system flexibility and resident participation. As a flexible crowdsensing strategy emphasizes open participation, participatory mobile crowdsensing enables the urban administrator to publish some "tasks" in the crowdsensing platform, and then, any resident owns a fair opportunity to bid for these tasks. In a more professional view, as illustrated in Figure 1 [10], the

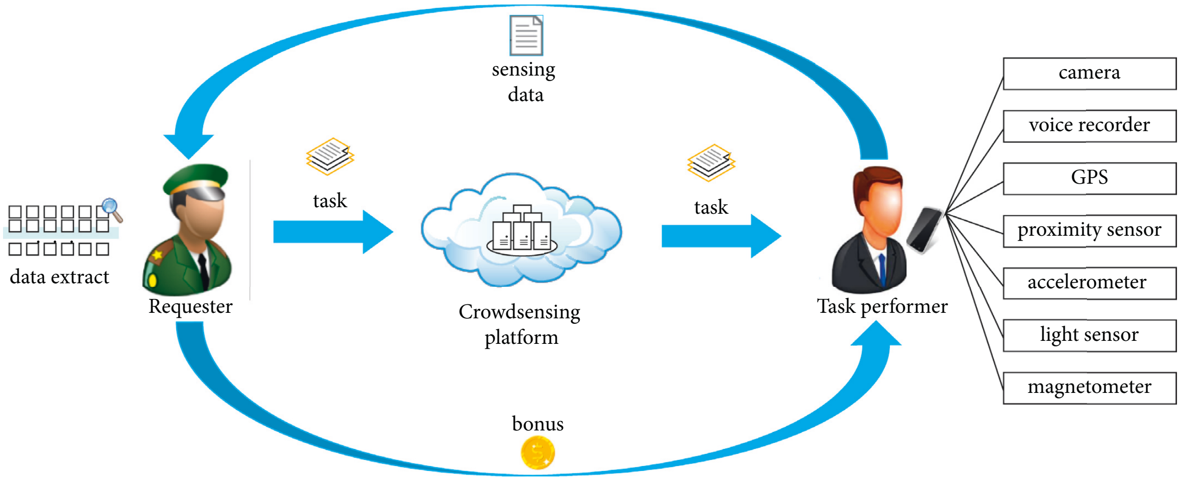


Figure 1: An overview of mobile crowdsensing system.

urban administrator acts as the requester, designs, and releases tasks to the crowdsensing platform. The task performer played by the resident scans the crowdsensing platform, chooses, and then subscribes to an available task. The task performer then executes his/her task and gets a reward by collecting via various embedded sensors and uploading sensing data within a specified time. Finally, the requester aggregates and filters the sensing dataset to obtain the optimal subset [11].

Featured with public access accessibility, low cost, and efficiency, mobile crowdsensing system wins widespread popularity and has been increasingly deployed in the public utilities IoT applications. On the other hand, however, privacy and data security issues in practical scenarios raise as a broad concern. Data circulating in the mobile crowdsensing system including task messages released by requesters and the sensing data collected by task performers, and it is possible for the attacker to conduct an attack by exploiting these two types of data. Specifically speaking, for instance, attackers may extract and analyze the weaknesses of urban facilities from the task message and then further break the vulnerable infrastructure such as the power grid system. On the other side, the attacker can also reveal the privacy information (for example, the home address, commutes route) of task performers from the sensing data [12]. For the sensing data, a public-key cryptosystem [13] is evaluated to be a feasible privacy protection strategy; that is, the task performer encrypts the sensing data with the public key previously released by the requester, and then, the requester can decrypt it to recover the plaintext-form sensing data [14]. However, the same cryptographic measure may not work for the task message, since the requester cannot predict which specific task performer would take over a task, and she/he is restricted to only encrypting and uploading a task message after confirming each task performer, which would reduce the efficiency of the mobile crowdsensing system. Besides, this measure also arouses worries about computational efficiency and the identity privacy of task performers. That is, for a task that requires multiple participants, the requester has to separately encrypt the same task message with the public key of each task performer, which incurs heavy computational overheads. And in this circumstance, the fact that the requester reveals which task performers subscribed to the task (in other words, breaks their identity privacy) is also self-evident.

Fortunately, attribute-based encryption (ABE) [15] provides the fine-grained, one-to-many, and privacy-preserving access control mechanism and is expected to break the above efficiency and privacy dilemma of the crypto-supported mobile crowdsensing system. In a typical ABE scheme, the task performer is labeled with a set of descriptive attributes, while a task message is encrypted with a specified attribute structure. A task performer can recover the task message if and only if his/her attribute set satisfies the access structure. This implies that the requester is just required to assign an [16–20] access structure and encrypt the task message for one time, and then, all task performers whose attribute sets satisfy the access structure are authorized to access the task message. In the process, the requester can reveal nothing about the task performer's identity, while only knowing s/he is an anonymous performer who holds a certain set of attributes.

Inspired by that, the latest research works put forward the solutions for the secure task distribution in mobile crowdsensing. However, there exist some gaps between these theoretically feasible solutions [10, 22, 24, 26, 36] and the practicality of mobile crowdsensing. Specifically, first, there is a practical issue that task performers have to locate their desired task among the stored ciphertext before downloading and decryption. Although there are some ABE-based solutions [26] that support keyword search, they are also hard to be practically deployed on mobile crowdsensing platforms for their cumbersome search procedures. Secondly, existing solutions (such as [10, 22]) are subject to heavy computational and storage overheads of the mobile terminal in the decryption side. This performance issue prevents them from further deployment in the mobile crowdsensing platform. Thirdlyin current ABE-based task distribution solutions for mobile crowdsensing, the sensitive information (such as the occupation and preference) about the task performer may be exposed from the public available access policy. Wang et al. provide a solution to this issue by hiding the access policy, but their solution also suffers from unworkable and cumbersome operations. To date, no work has systematically filled the above gaps by proposing a practical solution for task distribution in the mobile crowdsensing.

1.1. Contribution. In this paper, we put forward an efficient and privacy-preserving task distribution scheme (Scheme 1) and an edge-assisted scheme (Scheme 2) for IoT-oriented mobile crowdsensing, as the complete solution to fill the above gaps between current solutions and the practicality of mobile crowdsensing. Specifically, the contributions in our paper are described as follows.

(i) *Fine-Grained Access Control with Policy Hidden for the Task Distribution.* Inheriting the feature of one-to-many and fine-grained access from ABE, our solution enables the requester to share task messages with multiple task performers with encryption for only one time. During this process, the requester cannot and needs not to confirm the identity of each task performer. We then separate the attribute value from the attribute, thus hiding the specific attributes contained in the access structure, for preventing the sensitive information of the task performer from being leaked.

(ii) *Lightweight Keyword Search for the Encrypted Tasks.* It is obviously impractical for a task performer to search his/her desired ciphertext by downloading all ciphertext and seriatim decrypting them. To content the requirement that the task performer retrieves desired ciphertexts without decrypting, Scheme 1 designs the ciphertext keyword retrieval mechanism, in which the requester chooses a keyword that is associated with the task and then generates an “index,” and the task performer computes a “trapdoor” with his interested keyword. The crowdsensing platform can search those related ciphertexts by using the trapdoor, and during the process, the crowdsensing platform cannot learn any information about the task and the keyword. In Scheme 2, we further improve Scheme 1 by reducing the computational and communication cost of trapdoor generation and delivery for the performer side, as well as the cost of search on ciphertexts for the crowdsensing platform.

(iii) *Lightweight Decryption Operations.* To alleviate the computational overheads of the resource-constrained device on the task performer side, we delegate the decryption operation that should be assumed by the performer to the edge device in Scheme 2. Distinct from [10], in our solution, the edge device is considered to be semitrusted, which implies that we can prevent it from obtaining sensitive information including the user secret key.

1.2. Related Works. A sequence of solutions has been designed for the issues of data security and privacy protection of the task distribution phase in IoT-oriented mobile crowdsensing recently. Tao et al. [16] presented an anonymous bilateral authentication mechanism to guarantee the data authenticity while protecting the task performer’s identity privacy. Besides, they designed to solve the problem of large-scale key management in practical scenarios by using the pseudonym set. The requester usually requires the location of task performers to optimize task allocation, but it may reveal the location privacy of task performers. To protect the location privacy, Wang et al. [17] presented to fuzz the accurate location under the differential privacy constraint, thus protecting the location privacy. Karati and Biswas [18] proposed to simultaneously protect the data confidentiality and authenticity by inducing the identity-based encryption and designated verifier signature scheme. They also removed all pairing operations in their solution to improve the performance of cryptographic calculations. Ni et al. [19] proposed the SPOON scheme to attain the privacy protection of mobile users for task allocation. Specifically, it guarantees the confidentiality and authenticity of tasks with proxy reencryption and BBS+ [21] signature and uses an anonymous mechanism to protect the mobile user’s identity privacy.

Motivated by the feature of fine-grained and one-to-many access control of ABE, there are some works designed to integrate ABE into the data security and privacy protection strategy in mobile crowdsensing task distribution. Zhang et al. [36] proposed an ABE scheme with direct user revocation, which provides fine-grained access control on the encrypted time-sensitive task message for hierarchical task performers in a mobile crowdsensing system. Xue et al. [22] realized fine-grained and forward secure task access control in mobile crowdsensing by integrating ABE with Bloom filter encryption [23]. Besides, a “puncture” mechanism is imposed to the user secret key to prevent key reuse. Nkenyereye et al. [24] put forward a secure protocol based on ABE for mobile crowdsensing in the fog-based vehicular cloud [25], which supports policy updates for the fine-grained access control. Besides, they simultaneously protect the data authenticity and identity privacy with the pseudo-identity-based signature mechanism. To content the practical requirement that task performers search for some interested ciphertexts without decrypting them, Miao et al. [26] designed an ABE scheme with multikeyword search for mobile crowdsensing, and it realizes the flexible and comparable attribute access control by using the 1-encoding and 0-encoding technology [27]. Miao et al. [28] then presented a universal ABE scheme with ciphertext keyword search under the shared multiple data owners setting, and they also designed to hide the access policy to prevent the privacy information of data users from being revealed from their attributes. However, this scheme suffers from expensive computational consumption of the operations of ciphertext keyword search and decryption. Also aiming at attribute privacy, Zeng et al. [29] proposed a secure data sharing scheme for the medical IoT based on the partially policy-hidden ABE. In addition, it supports scalable flexibility and security: specifically, it is available for large attribute universe and user decryption key trace. Han et al. [30] proposed an ABE scheme with a similar partial policy-hiding mechanism, which also provides the user revocation and user decryption tracing to prevent the maliciously key leakage of a data user. Phuong et al. [44] put forward a fully policy-hidden ABE scheme, and as its name implies, it reveals nothing about the attributes in the access policy. But it is evaluated to be too inefficient to be practically deployed for its cumbersome algorithm structure. On this basis, Zhang et al. [31] alleviated the decryption overheads by applying the secure outsourced computing technology, but the system still cannot escape from the complicated algorithm structure. To address this problem, more recently, Ying et al. [32] constructed a novel fully policy-hidden ABE scheme with significant efficiency improvements by their designed security-enhanced Attribute Cuckoo Filter. This scheme also subtly integrates policy hiding into a policy update system.

Focus on the efficiency improvement, Tang et al. [33] indicated to alleviate the computational overheads of task encryption and decryption phase with online/offline encryption [34] and outsourced decryption [35] technology and designed to recommend the optimal task for task performers with the claimed "win-win" strategy. More recently, Wang et al. [10] presented a fine-grained access control protocol for the mobile crowdsensing platform, which enables the lightweight keyword generation and search and specifies the crowdsensing platform to predecrypt the ciphertext to reduce the computational overheads of the task performer side. However, their proposal is insecure unless they assumed the crowdsensing platform to be fully trusted, since they direct deliver the performer's user secret key to the crowdsensing platform.

1.3. Organizations. The remainder of this paper is organized as follows: Section 2 enumerates the preliminaries of our work, Section 3 presents the first scheme, Section 4 analyzes the security of the first scheme, Section 5 describes the second lightweight scheme, Section 6 evaluates the performance, and Section 7 concludes our work.

2. Preliminaries

2.1. Basic Concepts. **Bilinear Map**. Suppose G and G_T are two cyclic groups with prime order p, and g is the generator of G. A bilinear map e: $G \times G \longrightarrow G_T$ satisfies the following properties:

(i) Bilinearity: $e(g^a, g^b) = e(g, g)^{ab}$, where $g \in G$, and $a, b \in Z_p$

(ii) Non-degeneracy: $e(g, g) = 1$

(iii) Computaility: there exists an efficient algorithm to compute $e(g, g)$ for any $g \in G$

Hardness Assumption. Assume $a, b \in Z_p^*$, and $g \in G$ is selected as a generator of group G. The decisional Diffie-Hellman (DDH) assumption is described as follows: given a three tuple (g, g^a, g^b, R), there exists a probabilistic polynomial time (PPT) algorithm to determine whether $R = g^{ab}$ or R is a random element from group G.

2.2. System and Security Model. This paper designs an interactive system that involves four entities: the trusted authority (TA), the requester, the task performer, and the crowdsensing platform. TA is a fully trusted entity, which is responsible for initializing the system and distributing the user secret key according to the attribute set for each task performer. The requester is designed as a fully trusted entity that uploads the ciphertext (encrypted task message) to the crowdsensing platform. The task performer searches for his/her interested encrypted task message and then recovers the task message. The crowdsensing platform is used for storing the ciphertext and retrieving the keyword-related ciphertext for the task performer. It is also powerful enough to assist the resource-constrained task performer to decrypt the ciphertext. It is evaluated as a semitrusted entity; that is, it can honestly execute the cryptographic protocol but is curious about the sensitive information of its stored data and the user's privacy.

Figure 2 illustrates the workflow of the proposed system, where the specially designed algorithms for our Scheme 2 are denoted with blue dotted boxes. In logical order, the TA first runs the **Setup** algorithm to initialize the whole system, and then, it performs the **KeyGen** algorithm to distribute the user secret key for each registered task performer. The requester encrypts the task message with an access policy and ties the keyword index to the ciphertext by, respectively, running the **Encrypt** and the **Index** algorithms. Then, the requester uploads the ciphertext and the index to the crowdsensing platform. If a task performer wants to take on a task, s/he first specifies an interested queried keyword and then generates a trapdoor [42–46] with his/her user secret key and the queried keyword. The task performer forwards the trapdoor to the crowdsensing platform to request all ciphertexts related to the keyword. Subsequently, by running the **Search** algorithm, the crowdsensing platform estimates whether a ciphertext satisfies the keyword requirement of the trapdoor, and then, it returns the satisfied ciphertext to the task performer. Upon receiving the ciphertext from the crowdsensing platform, the task performer recovers the plaintext-form task message from the ciphertext with his/her user secret key by running the **Decrypt** algorithm. Considering the performance constraint on the task performer side, as well as the mass data stored in the crowdsensing platform, we deploy the more efficient Scheme 2 for the resource-constrained task performer device and the crowdsensing platform, and we deploy the more efficient Scheme 2 for the resource-constrained task performer device and the crowdsensing platform. Specifically, the **KeyGen** and **Search** algorithms are reconstructed in a lightweight manner. Besides, to attain efficient decryption for the task performer, we design the **Transform**, the **TranKeyGen** algorithms, and rebuild the **Decrypt** algorithm as described in Figure 2. To be specific, following the **TranKeyGen** algorithm, the task performer first generates a transformation key based on his/her user secret key and then forwards it to

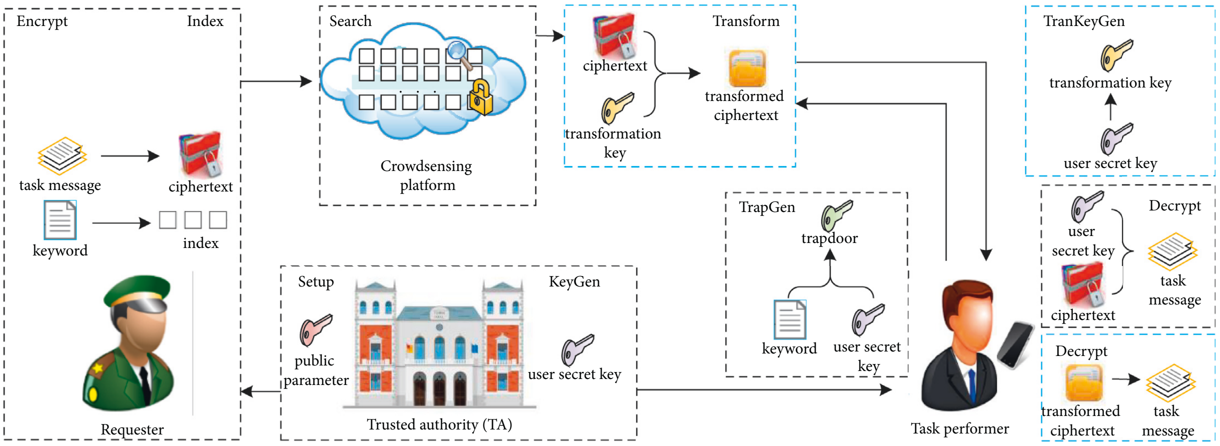


Figure 2: The system workflow.

the crowdsensing platform. By utilizing the transformation key, the crowdsensing platform runs the **Transform** algorithm to predecrypt the ciphertext and then returns the transformed ciphertext to the task performer. Finally, the task performer can execute the blue-marked **Decrypt** algorithm to recover the task message with lightweight operations.

2.3. Security Model. The basic scheme (Scheme 1) is indistinguishable under the chosen plaintext attack (IND-CPA) secure. The security model is parsed as an interactive game $\text{Game}_{\text{IND-CPA}}$ between a probabilistic polynomial time (PPT) adversary $\mathscr{A}$ and a challenger $\mathscr{C}$ as follows.

(i) **Initialize.** The adversary $\mathscr{A}$ specifies a challenged access structure $\mathbb{A}^*$

(ii) **Setup.** The challenger $\mathscr{C}$ runs the Setup algorithm to generate the public parameter MPK for $\mathscr{A}$

(iii) **Phase 1**. The adversary $\mathscr{A}$ queries on the user secret key of an attribute set, and then, the challenger $\mathscr{C}$ runs the **KeyGen** algorithm to generate a valid user secret key usk_W for $\mathscr{A}$

(iv) **Challenge**. The adversary $\mathscr{A}$ designates two equal-length task messages M_0 and M_1, then forwards them to $\mathscr{C}$, and $\mathscr{C}$ executes the Encrypt algorithm; that is, it randomly picks $b \in \{0, 1\}$ and encrypts M_b with an access policy $\mathbb{A}$ and then returns the ciphertext CT to $\mathscr{A}$

(v) **Phase 2.** This phase is the same as Phase 1

(vi) **Guess**. $\mathscr{A}$ outputs its guess $b' \in \{0, 1\}$ on b, and if $b' = b$, then we say $\mathscr{A}$ wins the game

Definition 1. If the basic scheme (Scheme 1) is indistinguishable against the chosen plaintext attack, then the probability for any PPT adversary$\mathscr{A}$ to win the above game $\text{Game}_{\text{IND-CPA}}$ is negligible.

Similarly, the security model of indistinguishability under the chosen keyword attack is parsed as an interactive game $\text{Game}_{\text{IND-CKA}}$ between a PPT adversary $\mathscr{A}$ and a challenger $\mathscr{C}$ as follows.

(i) **Initialize.** The adversary $\mathscr{A}$ specifies a challenged access structure $\mathbb{A}^*$

(ii) **Setup.** The challenger $\mathscr{C}$ runs the Setup algorithm to generate the public parameter MPK for $\mathscr{A}$

(iii) **Phase 1.** The adversary $\mathscr{A}$ answers the queries issued by $\mathscr{C}$

(iv) User secret key query. $\mathscr{A}$ queries on the user secret key of an attribute set W, and then, the challenger $\mathscr{C}$ runs the **KeyGen** algorithm to generate a valid user secret key usk_W for $\mathscr{A}$

(v) Trapdoor query. $\mathscr{A}$ queries on the trapdoor of a queried keyword q, and then, the challenger $\mathscr{C}$ runs the TrapGen algorithm to generate a valid trapdoor TD for $\mathscr{A}$

(vi) **Challenge**. The adversary $\mathscr{A}$ designates two equal-length task messages M_0 and M_1, then forwards them to $\mathscr{C}$, and $\mathscr{C}$ executes the Encrypt and Index algorithm; that is, it randomly picks $b \in \{0, 1\}$ and encrypts M_b with an access policy $\mathbb{A}$ and then returns the ciphertext CT to $\mathscr{A}$

(vii) **Phase 2**. This phase is the same as Phase 1

(viii) **Guess**. A outputs its guess $b' \in \{0, 1\}$ on b, and if $b' = b$, then we say $\mathscr{A}$ wins the game

Definition 2. If Scheme 2 is indistinguishable against the chosen keyword attack, then the probability for any PPT adversary $\mathscr{A}$ to win the above game $\text{Game}_{\text{IND-CKA}}$ is negligible.

3. Basic Scheme (Scheme 1)

This section detailedly describes the four interactive phases among those four kinds of entities in the basic scheme (Scheme 1).

3.1. System Initialization. TA runs the following Setup algorithm to establish the system and generate requisite system parameters.

Setup (λ): Taking the security parameter λ as input, TA selects two multiplicative cyclic groups $\mathbf{G}$, $\mathbf{G_T}$ with prime order $\mathbf{p}$, and $\mathbf{g}, \mathbf{u}, \mathbf{h}, \mathbf{w}, \mathbf{v}$ are five generators in group $\mathbf{G}$ and then define the bilinear pairing $\mathbf{e}$: $\mathbf{G} \times \mathbf{G} \longrightarrow \mathbf{G_T}$. Besides, we define a collision-resistant hash function $\mathbf{H}$: $\{0,1\}^* \longrightarrow \mathbf{Z_p}$. We set the attribute universe as $\mathbf{U} = \{\mathbf{A}_1, \ldots \mathbf{A_n}\}$. It also randomly selects $\alpha, \beta \in \mathbf{Z_p}$, then keeps secret the master secret key $\mathbf{MSK} = \alpha, \beta$, and makes the public parameter $\mathbf{MPK} = (\mathbf{p}, \mathbf{g}, \mathbf{G}, \ \mathbf{G_T}, \mathbf{H}, \mathbf{e}, \mathbf{u}, \mathbf{h}, \mathbf{w}, \mathbf{v}, \mathbf{Z} = \mathbf{e}(\mathbf{g}, \mathbf{g})^{\alpha}, \mathbf{Z}' = \mathbf{e}(\mathbf{g}, \mathbf{g})^{\alpha\beta}, \mathbf{U})$ to be publicly available.

3.2. User Registration. A newly added task performer with identity ID issues a registration request to TA. In response, by running the **KeyGen** algorithm, TA distributes the user private key usk_W for each registered task performer according to the identity ID and attribute set W.

KeyGen (MSK, MPK, W): this algorithm takes the public parameter MPK, the master secret key MSK, and the user attribute value set $W = \{W_1, \ldots, W_n\}$, where $\{W_i\} \in \{0,1\}$. For each attribute $A_i \in W$, TA picks $\{r_i\} \in Z_p$, where $i \in [1,n]$ randomly, and it also samples $r \in Z_p$. Under the above settings, TA computes $K_0 = g^{\alpha} w^{r}$, $K_0' = g^{\alpha\beta} w^{r}$, $K_1 = g^{r}$, $\{K_{i,2} = g^{r_i}\}_{i \in [1,n]}$, $\{K_{i,3} = (u^{A_i} h)^{r_i} v^{-r}\}_{i \in [1,n]}$, and TA then assembles the user secret key $usk_W = (W, K_0, K_0', K_1, \{K_{i,2}, K_{i,3}\}_{i \in [1,n]})$ and delivers usk_W to the task performer via secure channel.

3.3. Task Encryption and Distribution. To attain secure task distribution, the requester encrypts his/her tasks and uploads the encrypted task to the crowdsensing platform by running the following described Encrypt and Index algorithm. Notice that for each task message$M$, we specify a keywordδ to enable the encrypted task can be retrieved by any task performers without revealing to irrelevant entities the detailed information about the keyword.

(i) **Encrypt** $(MPK, M, \mathbb{A})$: the requester takes the public parameter MPK, the plaintext-form task message M, the revocation list RL, and the AND-Gate access policy $\mathbb{A}$, and then, s/he randomly chooses $s, s_1, \ldots, s_{n-1} \in Z_p$ and calculates $s_n = s - \sum_{i=1}^{n-1} s_i$. The access structure $\mathbb{A}$ is instantiated to $S = \{S_1, \ldots, S_n\}$, where $\{S_i\} \in \{0,1\}$. On this basis, the requester picks $t_1, \ldots, t_n \in Z_p$ and then calculates $C_0 = M \cdot e(g,g)^{\alpha s}, C_1 = g^{s}$, $\{C_{i,2} = w^{s_i} v^{t_i}$, $C_{i,3} = g^{t_i}\}_{i \in [1,n]}$. Besides, this algorithm requires the requester to compute $C_{i,4} = (u^{A_i} h)^{-t_i}$ for each attribute $A_i \in W \cap S$, while randomly selects $C_{i,4} \in G$ for each attribute $A_i \notin W \cap S$.

(ii) **Index**(MPK, usk_W, δ): the requester assigns the most appropriate keyword by referring to the keyword dictionary for a task message M. Specifically, s/he takes as input the public key MPK, the user secret key usk_W, and the keyword δ and then invokes the collision-resistant hash function $H(\cdot)$ and generates the index as $I = Z' H(\delta)^{s}$. Finally, the requester assembles the ciphertext $CT = (C_0, C_1, \{C_{i,2}, C_{i,3}, C_{i,4}\}_{i \in [1,n]}, I)$ and uploads CT to the crowdsensing platform.

3.4. Task Encryption and Distribution. This phase describes the workflow on the task performer side. Specifically, s/he first generates a trapdoor about his/her skilled fields with a keyword query by running the **TrapGen** algorithm. By using the trapdoor, the crowdsensing platform locates the target encrypted task message with the **Search** algorithm and forwards it to the task performer. Finally, the task performer recovers the plaintext-form task message by running the **Decryption** algorithm.

(i) **TrapGen** (MPK, q, usk_W): the task performer inputs the public parameter MPK, the queried keyword q, and his/her user secret key usk_W and then calculates $T_0 = K' \cdot H(q)$, $T_1 = K \cdot H(q)$, and $\{T_{i,2} = K_{i,2}^{H(q)}, T_{i,2} = K_{i,3}^{H(q)}\}_{i \in [1,n]}$. The trapdoor is assembled as $TD = (T_0, T_1, \{T_{i,2}, T_{i,3}\}_{i \in [1,n]})$ and is forwarded to the crowdsensing platform via secure channel when the task performer requests a task

(ii) **Search**(MPK, TD, CT): The crowdsensing platform inputs the public parameter MPK, the trapdoor TD, and the ciphertext CT and then checks whether the following equation holds:

$$I = \frac{e(C_1, T_0)}{\prod_{i=1}^{n} \left(e\left(C_{i,2}, T_1\right) e\left(C_{i,4}, T_{i,2}\right) e\left(C_{i,3}, T_{i,3}\right) \right).} \tag{1}$$

If it holds, the crowdsensing platform returns the ciphertext CT to the task performer via the public channel, and otherwise, it aborts and feedbacks $\perp$

(iii) **Decrypt** (CT, usk_W): upon obtaining the desired ciphertext CT, the task performer takes as input his/her user secret key usk_W and recovers the plaintext-form task message M by figuring up the following equation:

$$M = \frac{\prod_{i=1}^{n} \left(e\left(C_{i,2}, K_1\right) e\left(C_{i,4}, K_{i,2}\right) e\left(C_{i,3}, K_{i,3}\right) \right)}{e(C_1, K_0)}. \tag{2}$$

4. Security Analysis

4.1. System Initialization

Theorem 1. *(IND-CPA): if a probabilistic polynomial time (PPT) adversary $\mathscr{A}$ can breach the proposed system with nonnegligible probability under the chosen plaintext attack, then a challenger algorithm$\mathscr{C}$ can be constructed to solve the DDH problem with a nonnegligible advantage*

Proof. The proof is constructed on the basis of the proof of the ciphertext policy-hidden ABE scheme in [37]. Given the four-tuple (g, g^{a}, g^{b}, R) as the input of the DDH assumption, the challenger $\mathscr{C}$ aims to determine whether $R = g^{ab}$ or a random value in the group G_T.

(i) **Initialize**: the adversary $\mathscr{A}$ claims its target AND-Gate access structure $\mathbb{A}^*$ (it can be instantiated as the value set $S^* = (S_1^*, \ldots, S_n^*)$ to be challenged).

(ii) Setup: $\mathscr{C}$ defines two multiplicative cyclic groups G, G_T with prime order p and regulates the bilinear map $e\colon G \times G \longrightarrow G_T$, where g is selected as a generator of group G, $\mathscr{C}$ then samples $x, y, z, \alpha \in Z_p$, and computes $u = g^x$, $h = g^y$, $w = g^z, Z = e(g, g)^{\alpha}$, and sets $v = g^a$. Define the attribute universe $U = \{A_1, \ldots, A_n\}$. Finally, $\mathscr{C}$ returns to $\mathscr{A}$ the public parameter $MPK = (p, g, G, G_T, e, u, h, w, v, Z, U)$ and keeps secret the master secret key $MSK = \alpha$

(iii) Phase 1: The adversary $\mathscr{A}$ issues a sequence of queries to the challenger $\mathscr{C}$ as follows. Specifically, $\mathscr{A}$ forwards to $\mathscr{C}$ an attribute set W on the premise of that $W = \{W_1, \ldots, W_n\}$ does not content the challenged AND-Gate access structure S^*. As response, $\mathscr{C}$ randomly $r, \{r_i\} \in Z_{n+1}$ and then computes $K_0 = g^{\alpha} w^r$, $K_0' = g^{\alpha\beta} w^r$, $K_1 = g^r$, $\left\{K_{i,2} = g^{r_i}\right\}_{i \in [1,n]}$, and $\left\{K_{i,3} = (u^{A_i} h)^{r_i} v^{-r}\right\}_{i \in [1,n]}$, and C returns $usk_W = (W, K_0, K_0', K_1, \left\{K_{i,2}, K_{i,3}\right\}_{i \in [1,n]})$ to $\mathscr{A}$.

(iv) Challenge: the adversary $\mathscr{A}$ forwards two equal-length task messages M_0^* and M_1^*. As response, the challenger $\mathscr{C}$ randomly selects $\mu \in \{0, 1\}$ and implicitly sets $s = ab$ by regulating $C_0 = M \cdot e(g, g)^{\alpha s} = e(g, g)^{\alpha ab}, C_1 = R$. Assume that $A_j \notin S$, for each attribute A_i, where $i \in [1, n], i = j$, the challenger $\mathscr{C}$ randomly chooses $s_i, t_i \in Z_p$. Besides, $\mathscr{C}$ calculates $s_j = ab - \sum_{i=1, i \neq j}^{n} s_i$ and $t_j = -zb$ for the circumstance $i = j$, and $\mathscr{C}$ randomly selects $\mu \in \{0, 1\}$ and implicitly sets $s = ab$ by regulating $C_0 = M \cdot e(g, g)^{\alpha s} = e(g, g)^{\alpha ab} = e(g^a, g^b)^{\alpha}$ and $C_1 = R$. If $i = j$, then $\mathscr{C}$ calculates $C_{j,2} = w^{s_j} v^{t_j} = g^{z(ab - \sum_{i=1, i \neq j}^{n} s_i)} g^{-azb} = g^{-z \sum_{i=1, i \neq j}^{n} s_i}$ and $C_{j,3} = g^{t_j} = g^{-zb} = (g^b)^{-z}$ and randomly chooses $C_{j,4}$ from group G. If $i \neq j$, $\mathscr{C}$ directly calculates $\left\{C_{i,2} = w^{s_i} v^{t_i}, C_{i,3} = g^{t_i}\right\}_{i \in [1,n]}$ and $C_{i,4} = (u^{A_i} h)^{-t_i}$.

(v) Phase 2: This phase is the same as Phase 1.

(vi) Guess: The adversary A outputs its guess $\mu' \in \{0, 1\}$ on μ'. If $\mathscr{A}$ outputs $\mu' = \mu$, $\mathscr{C}$ returns 1 to guess $R = g^{ab}$. Otherwise, if $\mathscr{A}$ outputs $\mu' \neq \mu$, $\mathscr{C}$ returns 0 to guess R is a random element in group G. □

Theorem 2. *(IND-CKA): If a probabilistic polynomial time (PPT) adversary$\mathscr{A}$ can breach the proposed system with nonnegligible probability under the chosen keyword attack, then a challenger algorithm$\mathscr{C}$ can be constructed to solve the DDH problem with a nonnegligible advantage.*

Proof. Given the four-tuple (g, g^a, g^b, R) as the input of DDH assumption, the challenger $\mathscr{C}$ aims to determine whether $R = g^{ab}$ or a random value in the group G_T ,

(i) **Initialize**: The adversary $\mathscr{A}$ claims its target AND-Gate access structure $\mathbb{A}^*$ (it can be instantiated as the value set $S^* = (S_1^*, \ldots, S_n^*)$ to be challenged)

(ii) **Setup**: $\mathscr{C}$ defines two multiplicative cyclic groups G, G_T with prime order p and regulates the bilinear map $e\colon G \times G \longrightarrow G_T$, where g is selected as a generator of group G, $\mathscr{C}$ also regulates a collision-resistant hash function $H\colon \{0,1\}^* \longrightarrow Z_p$, and $\mathscr{C}$ then samples $x, y, z, \alpha \in Z_p$, computes $u = g^x$, $h = g^y, w = g^z, Z = e(g, g)^{\alpha\beta}$, and sets $v = g^a$. Define the attribute universe $U = \{A_1, \ldots, A_n\}$. Finally, $\mathscr{C}$ returns to $\mathscr{A}$ the public parameter $MPK = (p, g, G, G_T, H, e, u, h, w, v, Z', U)$ and keeps secret the master secret key $MSK = \alpha, \beta$.

(iii) **Phase 1**: The adversary $\mathscr{A}$ issues a sequence of queries to the challenger $\mathscr{C}$ as follows.

(iv) User secret key query: $\mathscr{A}$ forwards to $\mathscr{C}$ an attribute set W on the premise of that $W = \{W_1, \ldots, W_n\}$ does not content the challenged AND-Gate access structure S^*. As response, $\mathscr{C}$ randomly $r, \{r_i\} \in Z_{n+1}$ and then computes $K_0 = g^{\alpha} w^r$, $K_0' = g^{\alpha\beta} w^r$, $K_1 = g^r$, $\left\{K_{i,2} = g^{r_i}\right\}_{i \in [1,n]}$, and $\left\{K_{i,3} = (u^{A_i} h)^{r_i} v^{-r}\right\}_{i \in [1,n]}$, and $\mathscr{C}$ returns $usk_W = (W, K_0, K_0', K_1, \left\{K_{i,2}, K_{i,3}\right\}_{i \in [1,n]})$ to $\mathscr{A}$.

(v) Trapdoor query: $\mathscr{A}$ issues a query on the transformation key of $usk_W = (W, K_0,$ $K_0', K_1, \left\{K_{i,2}, K_{i,3}\right\}_{i \in [1,n]})$ to $\mathscr{C}$, and $\mathscr{C}$ assigns a desired keyword q, calculates $T_0 = K' H(q)$, $T_1 = KH(q)$, $\{$ $T_{i,2} = K_{i,2} H(q)$, $\left\{T_{i,3} = K_{i,3} H(q)\right\}_{\mathrm{i} \in [1,\mathrm{n}]}$, and returns the trapdoor $TD = (T_0, T_1,$ $\left\{T_{i,2}, T_{i,3}\right\}_{i \in [1,n]})$ to $\mathscr{A}$.

(vi) Challenge: The adversary $\mathscr{A}$ forwards two equal-length task messages M_0^* and M_1^*. As response, the challenger $\mathscr{C}$ randomly selects $\mu \in \{0, 1\}$ and implicitly sets $s = ab$ by regulating $I = Z' H(\delta)^s = e(g, g)^{\alpha\beta} H(\delta)^s = e(g, g)^{\alpha\beta} H(\delta)^{ab}$, $C_1 = R$. Assume that $A_j \notin S$, for each attribute A_i, where $i \in [1, n], i = j$, the challenger $\mathscr{C}$ randomly chooses $s_i, t_i \in Z_p$. Besides, $\mathscr{C}$ calculates $s_j = ab - \sum_{i=1, i \neq j}^{n} s_i$ and $t_j = -zb$ for the circumstance $i = j$. If $i = j$, the $C_{j,2} = w^{s_j} v^{t_j} = g^{z(ab - \sum_{i=1, i \neq j}^{n} s_i)} g^{-azb} = g^{-z \sum_{i=1, i \neq j}^{n} s_i}$, $C_{j,3} = g^{t_j} = g^{-zb} = (g^b)^{-z}$, and randomly chooses $C_{j,4}$ from group G. If $i \neq j$, $\mathscr{C}$ directly calculates $\left\{C_{i,2} = w^{s_i} v^{t_i}, C_{i,3} = g^{t_i}\right\}_{i \in [1,n]}$ and $C_{i,4} = (u^{A_i} h)^{-t_i}$.

(vii) Phase 2: This phase is the same as Phase 1.

(viii) Guess: The adversary $\mathscr{A}$ outputs its guess $\mu' \in \{0, 1\}$ on μ. If $\mathscr{A}$ outputs $\mu' = \mu$, $\mathscr{C}$ returns 1 to guess $R = g^{ab}$. Otherwise, if $\mathscr{A}$ outputs $\mu' \neq \mu$, $\mathscr{C}$ returns 0 to guess R is a random element in group G. □

4.2. Collusion Attack Resistance. Collusion attack indicates that multiple task performers whose attribute set does not satisfy the access structure may cheat the access authorization by combining their attributes-associated user secret keys. However, collusion attack is unavailing to our proposed scheme. Notice that the user secret key is parsed as $\mathbf{K}_0 = \mathbf{g}^{\alpha}\mathbf{w}^{\mathbf{r}}$, $\mathbf{K}_0' = \mathbf{g}^{\alpha\beta}\mathbf{w}^{\mathbf{r}}$, $\mathbf{K}_1 = \mathbf{g}^{\mathbf{r}}$, $\{\mathbf{K}_{\mathbf{i},2} = \mathbf{g}^{\mathbf{r}_\mathrm{i}}\}_{\mathbf{i}\in[1,\mathbf{n}]}$, $\{\mathbf{K}_{\mathbf{i},3} = (\mathbf{u}^{\mathbf{A}_\mathrm{i}}\mathbf{h})^{\mathbf{r}_\mathrm{i}}\mathbf{v}^{-\mathbf{r}}\}_{\mathbf{i}\in[1,\mathbf{n}]}$ for the component $\mathbf{K}_{\mathbf{i},3}$ that corresponds to the attribute $\mathbf{A}_\mathbf{i}$, and it is masked by the randomly selected $\mathbf{r} \in \mathbf{Z}_\mathbf{p}$, which is various for different task performers. Thus, multiple task performers cannot obtain a valid user secret key by just combining their individual user secret keys.

4.3. Attribute Privacy Protection and Policy Hidden. We instantiate the access structure with the mechanism in [38] to attain policy hidden. Specifically, the attribute universe $\mathbf{U} = \{\mathbf{A}_1, \ldots, \mathbf{A}_\mathbf{n}\}$ is available for each entity in the proposed system. The task performer issued the attribute value set $\mathbf{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_\mathbf{n}\}$, while the plaintext-form task message is encrypted with another attribute value set (access policy) $\mathbf{S} = \{\mathbf{S}_1, \ldots, \mathbf{S}_\mathbf{n}\}$. What is remarkable is that elements $\mathbf{W}_\mathbf{i}$ and $\mathbf{S}_\mathbf{i}$ are Boolean value or the wildcard $*$, and they just indicate whether the i-th attribute in the attribute universe U is contented for $\mathbf{W}$ or $\mathbf{S}$, or say “do not care” for the i-th attribute in $\mathbf{U}$ [38]. Therefore, (policy) attributes privacy cannot be revealed from the task performer’s attribute set $\mathbf{W}$ and access policy $\mathbf{S}$.

4.4. Keyword Privacy and Unlinkability. The keyword and the queried keyword are, respectively, embedded in the ciphertext and the trapdoor in the form of $I = Z' \cdot H(\delta)^s$ and $T_0 = K'H(q)$, $T_1 = KH(q)$, $\{T_{i,2} = K_{i,2}H(q)T_{i,3} = K_{i,3}H(q)\}_{\mathrm{i}\in[1,\mathrm{n}]}\{T_{i,3} = K_{i,3}H(q)\}_{\mathrm{i}\in[1,\mathrm{n}]}$. The crowdsensing platform is unable to reveal $H(\delta)$ from I since it is masked by the secret s. Similarly, it also cannot extract $H(q)$ from those trapdoor components for its unknown of the user secret key. Besides, we assert that nobody can reveal the equality of two trapdoors from different two task performers, despite they correspond to the same queried keyword q, since each task performer secretly holds his/her unique user secret key usk_W.

5. An Improved Scheme (Scheme 2)

Motivated by [40, 41], we design a more efficient scheme for the task performer and the crowdsensing platform. This scheme provides a lightweight trapdoor generation and search mechanism and delegates most decryption operations of the task performer to the edge device [39]. In comparison to Scheme 1, on the task performer side, we alleviate the computational and communication cost of the trapdoor generation and transmission and also significantly reduce the decryption cost while, in the edge side, we eliminate similar (or repeated) computations to lower the computational cost of ciphertext keyword search.

5.1. System Initialization. Setup (λ): Taking the security parameter λ as input, TA selects two multiplicative cyclic groups $\mathbf{G}$, $\mathbf{G_T}$ with prime order $\mathbf{p}$, and $\mathbf{g}, \mathbf{u}, \mathbf{h}, \mathbf{w}, \mathbf{v}$ are five generators in group $\mathbf{G}$, then defines the bilinear pairing $\mathbf{e}\colon \mathbf{G} \times \mathbf{G} \longrightarrow \mathbf{G_T}$. Besides, we define a collision-resistant hash function $\mathbf{H}\colon \{0,1\}^* \longrightarrow \mathbf{Z_p}$. We set the attribute universe as $\mathbf{U} = \{\mathbf{A}_1, \ldots, \mathbf{A}_\mathbf{n}\}$. It also randomly selects $\alpha, \beta \in \mathbf{Z_p}$, then keeps secret the master secret key $\mathbf{MSK} = (\alpha, \beta)$, and makes the public parameter $\mathbf{MPK} = (\mathbf{p}, \mathbf{g}, \mathbf{G}, \mathbf{G_T}, \mathbf{H}, \mathbf{e}, \mathbf{u}, \mathbf{h}, \mathbf{w}, \mathbf{v}, \mathbf{Z} = \mathbf{e}(\mathbf{g}, \mathbf{g})^{\alpha}, \mathbf{Z'} = \mathbf{e}(\mathbf{g}, \mathbf{g})^{\alpha\beta}, \mathbf{U})$ to be publicly available.

5.2. User Registration. **KeyGen** (MSK, MPK, W): This algorithm takes the public parameter MPK, the master secret key MSK, and the user attribute value set $W = \{W_1, \ldots, W_n\}$, where $\{W_i\} \in \{0,1\}$. For each attribute $A_i \in W$, TA picks $\{r_i\} \in Z_p$, where $i \in [1,n]$ randomly, it also samples $r \in Z_p$. Under the above settings, TA computes $K_0 = g^{\alpha}w^r$, $K_0' = g^{\alpha\beta}w^r$, $K_1 = g^r$, $\{K_{i,2} = g^{r_i}\}_{i\in[1,n]}$, $\{K_{i,3} = (u^{A_i}h)^{r_i}v^{-r}\}_{i\in[1,n]}$, and $K_4 = g^{\alpha\beta}$. TA then assembles the user secret key $usk_W = (W, K_0, K_0', K_1, \{K_{i,2}, K_{i,3}\}_{i\in[1,n]}, K_4)$ and delivers usk_W to the task performer via secure channel.

5.3. Task Encryption and Distribution

(i) Encrypt $(MPK, M, \mathbb{A})$: The requester takes the public parameter MPK, the plaintext-form task message M, the revocation list RL, and the AND-Gate access policy $\mathbb{A}$, and then, s/he randomly chooses $s, s_1, \ldots, s_{n-1} \in Z_p$ and calculates $s_n = s - \sum_{i=1}^{n-1} s_i$. The access structure $\mathbb{A}$ is instantiated to $S = \{S_1, \ldots, S_n\}$, where $\{S_i\} \in \{0,1\}$. On this basis, the requester picks $t_1, \ldots, t_n \in Z_p$ and then calculates $C_0 = M\cdot\ e(g,g)^{\alpha s}, C_1 = g^s$, $\{C_{i,2} = w^{s_i}v^{t_i}$, $C_{i,3} = g^{t_i}\}_{i\in[1,n]}$. Besides, this algorithm requires the requester to compute $C_{i,4} = (u^{A_i}h)^{-t_i}$ for each attribute $A_i \in W \cap S$, while randomly selects $C_{i,4} \in G$ for each attribute $A_i \notin W \cap S$.

(ii) **Index** (MPK, usk_W, δ): The requester assigns the most appropriate keyword by referring to the keyword dictionary for a task message M. Specifically, s/he takes as input the public key MPK, the user secret key usk_W, and the keyword δ and then invokes the collision-resistant hash function $H(\cdot)$ and generates the index as $I = Z' \cdot e(C_1, H(\delta))$. Finally, the requester assembles the ciphertext $CT = (C_0, C_1, \{C_{i,2}, C_{i,3}, C_{i,4}\}_{i\in[1,n]}, I)$ and uploads CT to the crowdsensing platform.

5.4. Task Search

(i) TrapGen (MPK, q, usk_W): The task performer inputs the public parameter MPK, the queried keyword q, and his/her user secret key usk_W and then calculates.

The task performer delivers TD to the crowdsensing platform via secure channel when the task performer requests a task.

(ii) **Search** (MPK, TD, CT): The crowdsensing platform inputs the public parameter MPK, the trapdoor TD, and the ciphertext CT and then checks whether the equation $I = e(TD, C_1)$ holds. If it holds, the crowdsensing platform returns the ciphertext CT to the task performer via public channel; otherwise, it aborts and feedbacks $\perp$.

5.5. Task Reveal. In this phase, we design to delegate the decryption operation to the edge device without directly handing over the user secret key. By following this idea, we blind the user secret key with a randomly selected $t \in Z_p$, and then, the edge device can transform (predecrypt) the ciphertext with the "blinded" key. Specifically, this phase performs by running the following algorithms.

(i) **TranKeyGen** (MPK, usk_W): The task performer inputs the public parameter MPK and his/her user secret key usk_W, and then, s/he picks $t \in Z_p$ and calculates $tk_0 = K_0^t$, $tk_1 = K_1^t$, $\{\ tk_{i,2} = K_{i,2}^t$, $\{tk_{i,3} = K_{i,3}^t\}_{i \in [1,n]}$, and s/he assembles the transformation key $TK = (tk_0, tk_1, \{tk_{i,2}, tk_{i,3}\}_{i \in [1,n]})$ and forwards TK to the edge device via a public channel. Notice that the task performer is required to keep secret the parameter t.

(ii) Transform (CT, TK): Upon receiving the transformation key TK, the edge device takes as input the desired ciphertext CT and generates the transformed ciphertext by figuring up the following equation:

$$CT' = \frac{\prod_{i=1}^{n} \left(e\left(C_{i,2}, tk_1\right) e\left(C_{i,4}, tk_{i,2}\right) e\left(C_{i,3}, tk_{i,3}\right) \right)}{e\left(C_1, tk_0\right).} \quad (3)$$

(iii) **Decrypt** (CT, CT'): The task performer takes as input the ciphertext CT and the transformed ciphertext CT', and then, s/he recovers the plaintext-form task message M by computing $M = C_0 \cdot CT'^{1/t}$.

Lemma 1. *(IND-CPA): The Scheme 2 is indistinguishable against the chosen plaintext attack if the Scheme 1 is IND-CPA secure.*

Proof. We omit the detail proof since it is similar with the proof of Theorem 1. What is different is that the "transformation key query" phase should be supplemented, which enables the challenger $\mathscr{C}$ to answer a sequence of queries on the transformation key from the adversary $\mathscr{A}$. □

Lemma 2. *(IND-CKA): The Scheme 2 is indistinguishable against the chosen keyword attack if the Scheme 1 is IND-CKA secure.*

Proof. We omit the detail proof since it is similar to the proof of Theorem 2. □

6. Performance Evaluation

6.1. Functionality and Complexity. Table 1 shows the comparisons on functionality among related schemes, including ABKS-SM [28], FGTAC [10] as well as Scheme 1 and Scheme 2 proposed in this paper. As illustrated, all of these schemes provide the security proof of IND-CPA and IND-CKA. In comparison with ABKS-SM [28] and Scheme 1, FGTAC [10] and our Scheme 2 enable the AND-Gate access control, fast ciphertext keyword search, lightweight decryption, and policy hidden. However, lightweight decryption in FGTAC [10] relies on a fully trusted crowdsensing platform, which impairs its practicality. Our Scheme 2 is proposed to attain lightweight decryption for task performers under the semitrusted crowdsensing platform assumption.

Table 2 describes the comparison of the above-mentioned schemes in terms of computational and storage complexity. In addition to the functional and practical advantages, our Scheme 2 is superior to ABKS-SM [28] and FGTAC [10] in storage cost. Our Scheme 2 is also well-performed in other indicators (including user secret key generation, trapdoor generation, search, and decryption) of computational cost except for encryption cost. Of course, we need not worry about the encryption cost since it is executed by the powerful task requester.

6.2. Experiment Results. We have experimented our proposed Scheme 1, Scheme 2 as well as related schemes such as ABKS-SM [28] and FGTAC [10] to evaluate and compare their practical performance. This experiment is conducted on a personal computer with an Intel (R) Core(TM) i7-7500U, 2.9 GHZ CPU, and 64 bit Windows 10 OS, and it is supported by the JPBC-2.0.0 library. To attain the 80 bit security, the elliptic curve is instantiated by a supersingular curve $y^2 = x^3 + x$ on the finite field F_p with the embedding degree of 2, where the prime degree of the field F_p is $p = 12qr - 1$, and the order of group G is the 160 bit Solinas prime $q = 2^{159} + 2^{17} + 1$, and then, there exists $|G| = |G_T| =$ 128 bytes and $|Z_p^*| = 20$ bytes. Besides, we designate SHA-256 to be the hash function in the experiment. We implement our proposed Scheme 1, Scheme 2 as well as ABKS-SM [28] and FGTAC [10] on the Enron e-mail Dataset [45], which is a widely used dataset that consists of 1,227,255 emails with 493,384 attachments covering 151 custodians.

The experimental results are pictorially described in Figure 3. When evaluating computing performance, we set the number of attributes to increase from 10 to 100 at the interval of 10, and the number of attributes is set to increase from 10 to 50 with the interval of 10 while evaluating storage performance. It is worth noting that since each attribute contains multiple "attribute values" in ABKS-SM [28], for a fair comparison, we only consider the number of attribute values. Figure 3(a) illustrates the time consumption for task encryption of these four schemes, and their computational time costs grow linearly with the size of involved attributes,

Table 1: Properties comparisons.

Schemes	F1	F2	F3	F4	F5	F6
ABKS-SM [28]	LSSS, AND-gate	CPA, CKA	Semi	×	×	√
FGTAC [10]	AND-gate	CPA, CKA	Fully	√	√	√
Scheme 1	AND-gate	CPA, CKA	Semi	×	×	√
Scheme 2	AND-gate	CPA, CKA	Semi	√	√	√

Notations: F1: access structure; F2: security level; F3: security requirement of the crowdsensing platform; F4: fast search; F5: lightweight decryption; F6: policy hidden.

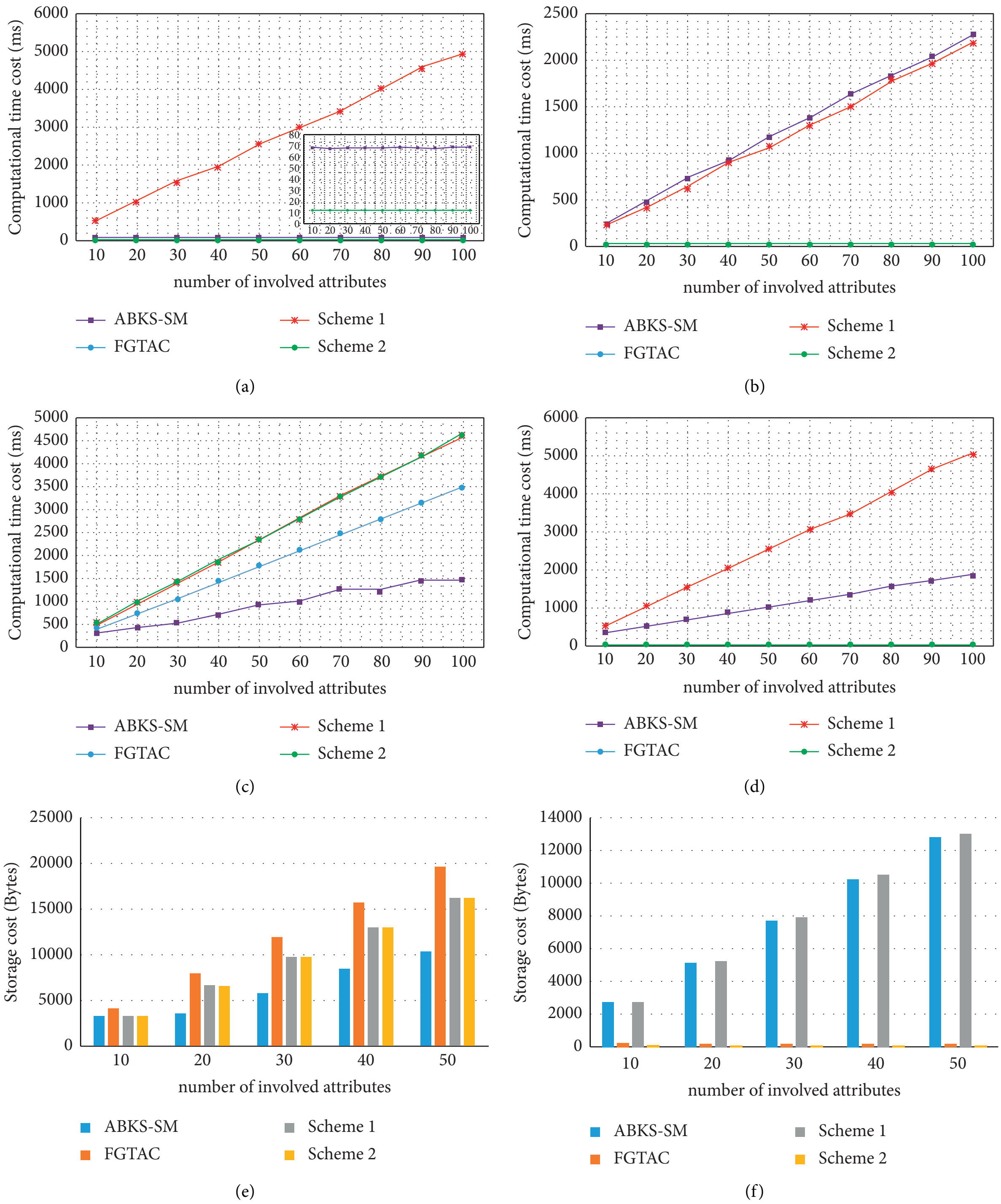
Table 2: Complexity comparisons.

Schemes	ABKS-SM [28]	FGTAC [10]	Scheme 1	Scheme 2
F1	$(2n+d+3)\lvert G\rvert$	$(4n+1)\lvert G\rvert$	$(2n+3)\lvert G\rvert$	$(2n+3)\lvert G\rvert$
F2	$(2n+d+4)e_G+e_{G_T}$	$(5n+2)e_G$	$(3n+5)e_G$	$(3n+5)e_G$
F3	$(\sum_{i=1}^{n} n_i+d+n+2)\lvert G\rvert+3\lvert G\rvert_T$	$(3n+2)\lvert G\rvert+\lvert G_T\rvert$	$(2n+m+1)\lvert G\rvert+\lvert G_T\rvert$	$(2n+m+1)\lvert G\rvert+\lvert G_T\rvert$
F4	$(\sum_{i=1}^{n} n_i+2\ d+n+2)e_G+3e_{G_T}$	$(3n+2)e_G+e_{G_T}$	$(3n+2m+1)e_G+e_{G_T}$	$P+(3n+2m+1)e_G+2e_{G_T}$
F5	$(2n+1)\lvert G\rvert$	$2\lvert G\rvert$	$(2n+2)\lvert G\rvert$	$\lvert G_T\rvert$
F6	$(2n+1)e_G$	$2\ e_G$	$(2n+2)e_G$	$P+e_{G_T}$
F7	$(2n+1)P+e_{G_T}$	$2P$	$(3n+1)P$	P
F8	$3P+de_G+de_{G_T}$	e_{G_T}	$(3n+1)P$	e_{G_T}

Notations: F1: size of the user secret key; F2: computational cost for user secret key generation; F3: size of the ciphertext; F4: computational cost for encryption; F5: size of the trapdoor; F6: computational cost for trapdoor generation; F7: computational cost for keyword search; F8: computational cost for decryption; **n**: number of attributes; $\mathbf{n_i}$: number of possible values for an attribute $\mathbf{A_i}$; **d**: number of data owners; **m**: number of user's attributes that satisfy the access policy; $\lvert\mathbf{G}\rvert$: an element in group **G**; $\lvert\mathbf{G_T}\rvert$: an element in group $\mathbf{G_T}$; **P**: a pairing operation; $\mathbf{e_G}$: an exponential operation over the group **G**; $\mathbf{e_{G_T}}$: an exponential operation over the group $\mathbf{G_T}$.

where our Scheme 1 and Scheme 2 show slight inferiority. However, they are acceptable since the requester is regarded as a powerful device, and even in our experiment platform, they generate a ciphertext within 5 seconds while the number of attributes reaches 100. This is because our schemes are constructed over the large-universe ABE scheme for attaining the scalable of attributes size in the mobile crowdsensing application; that is, it improves the usability at a few cost of efficiency. Figure 3(b) shows that for ciphertext keyword search, our Scheme 2 outperforms Scheme 1 and ABKS-SM [28] and is similar to FGTAC [10]; that is, the computational overhead is slight and constant. The time costs of FGTAC [10] and Scheme 2 are stable with the number of attributes, and those of the above four schemes, respectively, reach 1938.543 ms, 37.266 ms, 5045.127 ms, and 20.114 ms when the attributes number reaches 100. The excellent search performance of Scheme 2 is owed to our proposed lightweight search mechanism. For each ciphertext, we require the crowdsourcing platform to perform only one pairing operation involving the trapdoor, the index, and the key ciphertext component. We can observe from Figure 3(c) that the decryption time costs for the task performer in ABKS-SM [28], FGTAC [10], and Scheme 2 are constant even if the growth of the attributes number, but that of Scheme 2 is significantly less than ABKS-SM [28]. Specifically, the decryption time cost of Scheme 1 grows with the number of attributes (it attains 4972.268 ms when 100 attributes) while the remainders keep stable, which are within 70 ms and around 10 ms. This is due to the secure outsourcing and edge computing mechanism we implemented in Scheme 2 for the ciphertext decryption. In Figure 3(d), the trapdoor generation time consumption of Scheme 2 is slight and remains stable despite the attributes number increases, which is similar to that of FGTAC [10], and is far superior to ABKS-SM [28]. Specifically, the trapdoor generation time costs of both ABKS-SM [28] and Scheme 1 grow with the attributes number, and they are 2296.451 ms and 2207.195 ms, respectively, for 100 attributes setting. The time costs of FGTAC [10] and Scheme 2 are slight and nearly constant, and both of them are within 20 ms. This phenomenon also benefits from our lightweight keyword search mechanism that only requires a short and accessible trapdoor in Scheme 2 instead of embedding the queried trapdoor to each key component in Scheme 1. Figure 3(e) illustrates the comparison among these four schemes, and their ciphertext storage cost increases with the number of attributes. However, in fact, in Scheme 2, the task performer only needs to receive and store a constant size transformed ciphertext, which reduces the storage overhead of resource-constrained devices on the task performer side. In Figure 3(f), the trapdoor storage costs of FGTAC [10] and Scheme 2 are slight and constant size, which are friendly to the resource-constrained task performer side devices, despite that is growing with the number of attributes in ABKS-SM [28] and Scheme 1. This also benefits from our designed efficient ciphertext keyword search mechanism.

In a nutshell, our Scheme 1 uses ABE as the core to achieve task confidentiality and performer's identity privacy protection. Functionally, compared with other related works on mobile crowdsourcing security task distribution, Scheme 1 hides the access policy, thus preventing the performer's privacy leakage. And it allows the performer to flexibly



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3: Comparisons of time consumption in the publisher/subscriber side. (a) Encryption time cost. (b) Search time cost. (c) Decryption time cost. (d) Trapdoor generation time cost. (e) Ciphertext storage cost. (f) Trapdoor storage cost.

search the encrypted tasks it is interested in without revealing any preferences by designing the ciphertext keyword retrieval mechanism. In terms of performance, on the basis of Scheme 1, Scheme 2 implements an efficient ciphertext search mechanism, which allows the performer and the crowdsensing platform to generate a trapdoor and search ciphertexts with a small and fixed computational and storage overhead, respectively. On this basis, a large number of decryption operations that originally belonged to the task performer were transferred to the edge device. Compared with other related works on mobile crowdsourcing security task distribution, it improves the computational and storage performance on the performer side and crowdsensing platform side as shown in the experiment.

7. Conclusion

This paper designed the efficient and privacy-preserving task distribution mechanism for IoT-oriented mobile crowdsensing. We show our results by two practical cryptographic schemes. Scheme 1 realizes the fine-grained access control and access policy hidden by dividing the attribute into an attribute label and an attribute value, where the attribute value is publicly available, and the attribute label is hidden. We also design a keyword search mechanism over task ciphertexts that enables the task performer to conveniently generate the trapdoor. On this basis, Scheme 2 further improves the efficiency under the semitrusted crowdsensing platform assumption by delegating most operations to the crowdsensing platform and constructing a lightweight trapdoor. We then analyzed their security properties, provided the formalized security proof, and demonstrated their practicability and feasibility.

We note that although our work prevents the sensitive information of task performers from exposure, it still falls under the category of "partial policy hiding." The authors of [44] pointed out that some ABE schemes with partial policy hiding may still reveal the performer's attribute privacy. We notice that the latest representative work has transformed the primitive of full policy-hidden ABE from the cumbersome theoretical scheme to an efficient practical solution by optimizing the algorithm structure and extending the usability [32]. Therefore, in future work, we intend to further explore the more efficient and flexible ABE schemes with full policy hiding. In addition, although we profoundly reduce the trapdoor generation overhead on the performer side and the search burden of the crowdsensing platform, it may still suffer from the performance bottleneck in the crowdsensing platform with massive storage. In future work, we would like to explore an efficient ciphertext keyword search mechanism for the above practical setting.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Major Scientific Instrument and Equipment Development Project of National Natural Science Foundation of China (62027827).

References

[1] T. Li, Y. Tian, J. Xiong, and M. Z. Bhuiyan, "FVP-EOC: fair, verifiable and privacy-preserving edge outsourcing computing in 5G-enabled IIoT," *IEEE Transactions on Industrial Informatics*, p. 1, 2022.

[2] J. Sun, Y. Yuan, M. Tang, X. Cheng, X. Nie, and M. U. Aftab, "Privacy-preserving bilateral fine-grained access control for cloud-enabled industrial IoT healthcare," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6483–6493, 2022.

[3] D. Bd and F. Al-Turjman, "A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks," *Ad Hoc Networks*, vol. 97, Article ID 102022, 2020.

[4] Y. Bao, W. Qiu, X. Cheng, and J. Sun, "Fine-grained data sharing with enhanced privacy protection and dynamic users group service for the IoV," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2022.

[5] Y. D. Zhang, S. C. Satapathy, D. S. Guttery, J. M. Gorriz, and S. H. Wang, "Improved breast cancer classification through combining graph convolutional network and convolutional neural network," *Information Processing & Management*, vol. 58, no. 2, Article ID 102439, 2021.

[6] J. Sun, H. Xiong, X. Liu, Y. Zhang, X. Nie, and R. H. Deng, "Lightweight and privacy-aware fine-grained access control for IoT-oriented smart health," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6566–6575, 2020.

[7] Y. Bao, W. Qiu, and X. Cheng, "Secure and lightweight fine-grained searchable data sharing for IoT-oriented and cloud-assisted smart healthcare system," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2513–2526, 2022.

[8] C. Fiandrino, B. Kantarci, and F. Anjomshoa, "Sociability-driven user recruitment in mobile crowdsensing internet of things platforms," in *Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, Washington, DC, USA, December 2016.

[9] Q. Liang, X. Cheng, S. C. H. Huang, and D. Chen, "Opportunistic sensing in wireless sensor networks: theory and application," *IEEE Transactions on Computers*, vol. 63, no. 8, pp. 2002–2010, 2014.

[10] J. Wang, X. Yin, and J. Ning, "Fine-grained task access control system for mobile crowdsensing," *Security and Communication Networks*, vol. 2021, Article ID 6682456, 12 pages, 2021.

[11] B. Guo, Q. Han, H. Chen, L. Shangguan, Z. Zhou, and Z. Yu, "The emergence of visual crowdsensing: challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2526–2543, 2017.

[12] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[13] Y. Sun, P. Chatterjee, Y. Chen, and Y. Zhang, "Efficient identity-based encryption with revocation for data privacy in internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2734–2743, 2022.

[14] D. Wu, Z. Yang, B. Yang, R. Wang, and P. Zhang, "From centralized management to edge collaboration: a privacy-preserving task assignment framework for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4579–4589, 2021.

[15] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE symposium on security and privacy (SP'07)*, pp. 321–334, IEEE, Berkeley, CA, USA, May 2007.

[16] D. Tao, P. Ma, and M. S. Obaidat, "Anonymous identity authentication mechanism for hybrid architecture in mobile crowd sensing networks," *International Journal of Communication Systems*, vol. 32, no. 14, Article ID e4099, 2019.

[17] L. Wang, D. Yang, and X. Han, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 627–636, Geneva, Switzerland, April 2017.

[18] A. Karati and G. P. Biswas, "Provably secure and authenticated data sharing protocol for IoT-based crowdsensing network," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 4, Article ID e3315, 2019.

[19] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. Shen, "Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1317–1331, 2020.

[20] O. A. Khashan, "Hybrid lightweight proxy re-encryption scheme for secure Fog-to-Things environment," *IEEE Access*, vol. 8, pp. 66878–66887, 2020.

[21] M. H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic k-TAA," *International conference on security and cryptography for networks*, vol. 4116, pp. 111–125, 2006.

[22] L. Xue, J. Ni, and C. Huang, "Forward secure and fine-grained data sharing for mobile crowdsensing," in *Proceedings of the 2019 17th International Conference on Privacy, Security and Trust (PST)*, pp. 1–9, IEEE, Fredericton, Canada, August 2019.

[23] L. L. Gremillion, "Designing a Bloom filter for differential file access," *Communications of the ACM*, vol. 25, no. 9, pp. 600–604, 1982.

[24] L. Nkenyereye, S. R. Islam, M. Bilal, M. Abdullah-Al-Wadud, A. Alamri, and A. Nayyar, "Secure crowd-sensing protocol for fog-based vehicular cloud," *Future Generation Computer Systems*, vol. 120, pp. 61–75, 2021.

[25] J. Sun, G. Xu, T. Zhang, H. Xiong, H. Li, and R. Deng, "Share your data carefree: an efficient, scalable and privacy-preserving data sharing service in cloud computing," *IEEE Transactions on Cloud Computing*, p. 1, 2021.

[26] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attribute-based multi-keyword search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3008–3018, 2018.

[27] H. Y. Lin and W. G. Tzeng, "An efficient solution to the millionairesąŕ problem based on homomorphic encryption," *International Conference on Applied Cryptography and Network Security*, vol. 3531, pp. 456–466, 2005.

[28] Y. Miao, X. Liu, K. K. R. Choo et al., "Privacy-preserving attribute-based keyword search in shared multi-owner setting," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1080–1094, 2021.

[29] P. Zeng, Z. Zhang, R. Lu, and K. K. R. Choo, "Efficient policy-hiding and large universe attribute-based encryption with public traceability for internet of medical things," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10963–10972, 2021.

[30] D. Han, N. Pan, and K. C. Li, "A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 316–327, 2022.

[31] L. Zhang, W. You, and Y. Mu, "Secure outsourced attribute-based sharing framework for lightweight devices in smart health systems," *IEEE Transactions on Services Computing*, p. 1, 2021.

[32] Z. Ying, W. Jiang, X. Liu, S. Xu, and R. Deng, "Reliable policy updating under efficient policy hidden fine-grained access control framework for cloud data sharing," *IEEE Transactions on Services Computing*, p. 1, 2021.

[33] W. Tang, K. Zhang, J. Ren, Y. Zhang, and X. Sherman Shen, "Privacy-preserving task recommendation with win-win incentives for mobile crowdsourcing," *Information Sciences*, vol. 527, pp. 477–492, 2020.

[34] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption,"vol. 8383, pp. 293–310, in *Proceedings of the International Workshop on Public Key Cryptography*, vol. 8383, pp. 293–310, Springer, Berlin, Germany, March 2014.

[35] B. Qin, R. H. Deng, and S. Liu, "Attribute-based encryption with efficient verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1384–1393, 2015.

[36] J. Zhang, J. Ma, and T. Li, "Efficient hierarchical and time-sensitive data sharing with user revocation in mobile crowdsensing," *Security and Communication Networks*, vol. 57, pp. 34–56, 2021.

[37] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang, "Attribute based encryption with privacy protection and accountability for CloudIoT," *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 762–773, 2022.

[38] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures,"vol. 5037, pp. 111–129, in *Proccedings of the International Conference on Applied Cryptography and Network Security*, vol. 5037, pp. 111–129, Springer, Berlin, Germany, 2008.

[39] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, "A blockchain-based machine learning framework for edge services in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, 2022.

[40] J. Cui, H. Zhou, Y. Xu, and H. Zhong, "OOABKS: online/offline attribute-based encryption for keyword search in mobile cloud," *Information Sciences*, vol. 489, pp. 63–77, 2019.

[41] Y. Bao, W. Qiu, P. Tang, and X. Cheng, "Efficient, revocable and privacy-preserving fine-grained data sharing with keyword search for the cloud-assisted medical IoT system," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 2041–2051, 2022.

[42] J. Hao, C. Huang, and G. Chen, "Privacy-preserving interestability based task allocation in crowdsourcing," in *Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Shanghai, China, May 2019.

[43] A. De Caro and V. Iovino, "jPBC: java pairing based cryptography," in *Proceedings of the 2011 IEEE symposium on computers and communications (ISCC)*, pp. 850–855, IEEE, Kerkyra, Greece, July 2011.

[44] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 35–45, 2016.

[45] Enron Email Data, "Enron Email Data," 2016, https://aws.amazon.com/de/datasets/enron-email-data.

[46] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, pp. 463–474, Berlin, Germany, November 2013.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 9638683, 12 pages
https://doi.org/10.1155/2022/9638683

Research Article

Cross-Modal Discrimination Hashing Retrieval Using Variable Length

Chao He,[1] **Dalin Wang**,[2] **Zefu Tan**,[1] **Liming Xu**,[3] **and Nina Dai**[1]

[1]*School of Electronic and Information Engineering, Chongqing Three Gorges University, Chongqing 404100, China*
[2]*Chongqing Preschool Education College, Chongqing 404047, China*
[3]*School of Computer Science, China West Normal University, Nanchong 637002, Sichuan, China*

Correspondence should be addressed to Nina Dai; daininа83@163.com

Received 20 April 2022; Accepted 10 August 2022; Published 9 September 2022

Academic Editor: Qing Yang

Fast cross-modal retrieval technology based on hash coding has become a hot topic for the rich multimodal data (text, image, audio, etc.), especially security and privacy challenges in the Internet of Things and mobile edge computing. However, most methods based on hash coding are only mapped to the common hash coding space, and it relaxes the two value constraints of hash coding. Therefore, the learning of the multimodal hash coding may not be sufficient and effective to express the original multimodal data and cause the hash encoding category to be less discriminatory. For the sake of solving these problems, this paper proposes a method of mapping each modal data to the optimal length of hash coding space, respectively, and then the hash encoding of each modal data is solved by the discrete cross-modal hash algorithm of two value constraints. Finally, the similarity of multimodal data is compared in the potential space. The experimental results of the cross-model retrieval based on variable hash coding are better than that of the relative comparison methods in the WIKI data set, NUS-WIDE data set, as well as MIRFlickr data set, and the method we proposed is proved to be feasible and effective.

1. Introduction

With the advent of the big data era, the different types of modal data, e.g., text, image, and audio for the Internet of Things and Mobile Edge Computing, are dramatically increasing [1]. The traditional single-mode data retrieval methods, e.g., text retrieval text, image retrieval image, and audio retrieval audio, are gradual shift to cross-modal retrieval, e.g., text retrieval image, text retrieval audio, image retrieval text, which makes the retrieval return with the characteristics of diverse information and rich content [2]. Over the last few years, the cross-modal retrieval algorithms have been recently receiving significant attention and progress due to the application research of guaranteed data privacy and privacy-preserving cooperative object classification [3, 4].

There are two main categories in these research methods. One is the potential subspace learning-based method [5–8], among which the canonical correlation analysis (CCA) is the most commonly used model [5]. The CCA mapped the two-modal data into a potential subspace to achieve the correlation maximization of the associated data pairs, and then directly retrieves the similarity query in the subspace. Given the paramount idea of the correlation maximization of relevant data in subspace, some experts have proposed other deformation model algorithms similar to the CCA model. Fu et al. proposed the generalized Multiview analysis (GMA) to maximize the subspace correlation of multimodal data and achieve the class-discriminant via adding label information, which is conducive to further boosting the accuracy of the cross-modal retrieval [6]. Costa Pereira et al. first projected the original feature data of each mode into their respective semantic feature space, and then mapped the semantic features of multimodes into a unified subspace via applying CCA or kernel CCA. The proposed model utilized the label information of the data to improve the classification area analysis, meanwhile avoiding the direct mapping of the original multimodal features into the unified subspace so that the cross-modal retrieval performance is notably improved [7]. Mandal and Biswas proposed the generalized dictionary pair algorithm and achieved good results via learning unified sparse coding subspace [8]. Although some progress has been made in unified subspace learning-based cross-modal retrieval algorithms, there are still some problems in cross-modal retrieval of large-scale multimodal data scenarios, e.g., high computational cost, high data storage resource consumption, and weak stationarity. Therefore, another kind of cross-modal retrieval algorithm based on hashing coding has stimulated a lot of interest in the research community.

With the characteristics of storage consumption and efficient retrieval speed, the Hash coding technology is very suitable for large-scale data trans-modal and trans-media tasks, e.g., real-time multimodal data personalized recommendation [9], hot topic detection, and trans-media retrieval. In the Hash coding-based cross-modal retrieval method [10–13], for maintaining the connection between multimodal data, the multimodal data was projected into low-dimensional Hamming space through linear mapping, and then an XOR operation was performed to measure the similarity distance. Thus, the speed problem of large-scale data retrieval was solved effectively. However, most of the prior arts are only suitable for scenarios of the single label and paired training data. Therefore, Mandal et al. first proposed a hashing cross-modal retrieval model for multiple training scenarios [14]. However, this model is similar to the method presented in Refs. [15, 16] that maps multimode data into equal-length hash coding, so that the data of various modes may not be well represented. In addition, the solution of binary hash coding is an NP-hard problem, which relaxes the binary constraint of hash coding, so that the learned hash coding is not accurate enough. For analytical simplicity, this paper first proposed a cross-modal retrieval model based on variable-length hash coding and added binary constraints in the process of solving hash coding. Therefore, the learned variable-length hash coding can better represent the original multimodal data and achieve higher accuracy. The main highlights of this paper are organized as follows.

(1) To combat the issue caused by the same length, we propose a variable-length hash coding-based cross-modal retrieval model in this paper, i.e., all modal data are projected into the hash coding space of the optimal lengths. Therefore, compared with the hash coding space of the fixed length, the original multimodal data can be represented more easily, and the model in this paper is more flexible in debugging experiments.

(2) We propose a more generalized multiscene cross-modal retrieval. The great majority of the existing cross-modal retrieval models, based on single label and pairwise multimodal dataset scenarios, cannot be applied to multilabel and unpaired multimodal dataset scenarios. In addition, the cross-model retrieval in this paper has good adaptability to single label or multilabel, paired, or unpaired multimodal dataset scenarios.

(3) Based on the single-modal data hash method, we propose a variable-length discrete hash coding-based cross-modal retrieval algorithm, and the validity of the algorithm is verified on several public data sets.

2. Related Works

This section mainly introduces several related hash coding cross-modal retrieval algorithms, which are also served as benchmark algorithms in the experimental process. Any reader who has a great interest in other cross-modal retrieval models, such as incorporating feedback technology and deep learning, can refer to Ref. [17].

2.1. Hashing Cross-Modal Retrieval Based on Semantic Correlation Maximization. Taherkhani et al. proposed a Semantic Correlation Maximization (SCM)-based cross-modal hash retrieval model. Meanwhile, compared with other supervised hash cross-modal retrieval models, this model has the advantages of lower training time complexity, better adaptability, and more stability for large-scale data sets [10]. The main highlights are as follows. (1) The calculation of the complex pin-to-pair similarity matrix can be avoided directly via applying label information of the training data set to calculate the similarity matrix, thus only small linear time complexity can be achieved, which also makes the model more stable. (2) The serialization solution method of hash coding is proposed via the computation code of bit by bit on the closed interval. Therefore, there is no need to set hyperparameters and stop conditions. To use label semantic information, cosine similarity between label vectors is used to construct the similarity matrix, and the similarity between the data object i and the data object j is defined as follows.

$$S_{ij} = \frac{\langle l_i, l_j \rangle}{\|l_i\|_2 \|l_j\|_2}, \tag{1}$$

where $\langle l_i, l_j \rangle$ represents the inner product of the corresponding label vector and $\|l\|_2$ describes the binary norm of the label vector. To achieve a cross-modal similarity query, the hash function should maintain the semantic similarity of multimodal data. More specifically, the hash coding of each modal data can reconstruct the semantic similarity matrix. The specific objective function of the SCM model is defined as follows:

$$\min_{W_x, W_y} \left\| \operatorname{sign}(XW_x)\operatorname{sign}(YW_y) - cS \right\|_F^2, \tag{2}$$

where X and Y represent the data of the two modes, W defines the linear transformation matrix, c describes the equilibrium parameter, and S defines the similarity measurement between two data among different modalities. There is a symbolic function in (2), so it is obvious that the optimization solution is an NP-hard problem, which relaxes the constraints of the symbolic function and adds the constraints between the bits of the hash coding. Finally, the transformation matrixes W_x, W_y of each modal data can be calculated, so that the hash coding of new data can be resolved.

2.2. Hashing Cross-Modal Retrieval Based on Semantic Preserving. Chen et al. proposed a Semantic Preserving Hash cross-modal retrieval (SEPH) model, which converts the similar association information of data into the form of the probability distribution and then approximates hash coding via minimizing the Kullback–Leibler (KL) divergence distance [11]. The whole objective function model is effectively guaranteed in mathematical theory. As with the SCM model, the similarity matrix is first constructed to provide supervisory information for the learned hash coding. This model mainly includes two steps, i.e., hash coding solution and learning of kernel logic Sti regression function. When it comes to the process of solving the hash coding, the similarity matrix is first transformed into the form of probability P, and the semantic probability distribution Q on the unified hash coding is calculated, then the KL distance between the two distributions is minimized, and the semantic preserving hash coding is resolved.

$$P_{ij} = \frac{S_{ij}}{\sum_{i \neq j} S_{ij}},$$
$$Q_{ij} = \frac{\left(1 + h\left(B_i, B_j\right)\right)^{-1}}{\sum_{t=1} \left(1 + h\left(B_i, B_t\right)\right)^{-1}}, \tag{3}$$

where $h(,)$ represents the Hamming distance function of hash coding; learning the best hash coding B aims to make the distribution between P and Q as similar as possible. The KL distance between the distributions is measured as follows:

$$D_{KL}(PQ) = \sum_{i \neq j} P_{ij} \log\left(\frac{P_{ij}}{Q_{ij}}\right). \tag{4}$$

In all, a better unified semantically preserving hash coding can be calculated according to the solution steps, and then the logistic regression mapping function of each modal data mapped to the unified hash coding is learned. The representation of learning the $k(1 \leq k \leq K)$-th Logistic regression function for X mode data is defined as follows:

$$\min_{w^k} \sum_{i=1}^{n} \log\left(1 + e^{-b_i^k x_i w^k}\right) + \lambda \left\| w^k \right\|_2^2, \tag{5}$$

where $b_i^k \in \{-1, +1\}^{n \times 1}$ defines the column vector on the k-th bit attribute of the common binary code, and the transformation matrix w^k can be solved. Then, the probability that the value b belongs to −1 and +1 at the k-th bit of the binary code of the new sample x^q data in X mode can be calculated as follows:

$$P\left(c = b | x^q\right) = \left(1 + e^{-b x^q w^k}\right)^{-1}. \tag{6}$$

Therefore, the value at the k-th bit of data binary coding is selected as the value corresponding to the high probability, which is defined as follows:

$$c^k = \operatorname{sign}\left(P\left(c = 1 | x^q\right) - P\left(c = -1 | x^q\right)\right). \tag{7}$$

Finally, the k-th logistic regression function on the X mode data can be learned, and then the new sample x^q is mapped into the binary coding with the growing degree of K. The final hash coding can be achieved by changing the element with the value of −1 into 0.

2.3. Hashing Cross-Modal Retrieval Based on Generalized Semantic Preserving. Because most of the existing cross-modal retrieval methods require multimodal data to appear in pairs, i.e., another modal data corresponding to text or image exists in training set data, Mandal et al. proposed a Generalized Semantic Preserving Hashing model (GSPH) for N-label cross-modal retrieval, which is suitable for a single label or multilabel, paired or unpaired multimodal data application scenarios [14]. The GSPH model first learns the optimal hash coding of each modal data, meanwhile the hash coding preserves the semantic similarity between the multimodal data and then learns the hash function of multimodal data mapped to the hash coding space. The main highlights are as follows. (1) A hash model that can deal with single-label paired data and single-label unpaired data is proposed for the first time. (2) The generalized hash cross-modal retrieval model is proposed, which can be applied to the scenarios of single-label paired data, single-label unpaired data, multilabel paired data, as well as single-label unpaired data. Meanwhile, the semantic similarity of data is maintained by the common hash coding. As with SCM and SEPH methods, the GSPH algorithm also needs to define the similarity matrix $S \in R^{N_1 \times N_2}$ between multimodal data, where N_1 and N_2 are the sample numbers of X and Y modal data, respectively, so the objective function of the GSPH model is defined as follows:

$$\min \left\| S - \left(\frac{1}{q}\right) B_x B_y^T \right\|_F^2 s.t B_x \in \{-1, +1\}^{N_1 \times q}, B_y \in \{-1, +1\}^{N_2 \times q}. \tag{8}$$

The binary coding B_x and B_y of the X and Y modal data can be calculated by the GSPH method, and then the mapping function of the original data for each modal into hash coding needs to be learned. Just like the SEPH method, the logistic regression function is selected as the mapping function. Therefore, readers can refer to Section 2.2 for learning the mapping hash function and generating the hash coding of new samples.

3. Cross-Modal Retrieval Based on Variable-Length Hash Coding

In this section, the cross-modal retrieval algorithm of variable-length hash coding is presented, and the optimization process of the objective function and time complexity of the algorithm is analyzed. To facilitate the analytical simplicity and reduce the experimental operation, this paper mainly studies the case of two-modal data and gives the algorithm model extended to three or more modal data in Section 3.5.

3.1. Algorithmic Model. The variables presented in this paper are defined as follows. $X \in R^{d_1 \times n_1}$ and $X \in R^{d_2 \times n_2}$ represent the original feature data sets of the two modes, respectively, $B_X \in R^{q_1 \times n_1}$ and $B_Y \in R^{q_2 \times n_2}$ are the corresponding variable-length hash coding, where each column represents a sample and each row represents attribute features. In addition, P_X and P_Y are the projection matrixes, and W is the association matrix of two modes. The similarity matrix $S \in R^{n_1 \times n_2}$ between multimode data is constructed as follows:

$$S_{ij} = \begin{cases} \langle l_x^i, l_y^j \rangle & I, \\ e^{-\left\| l_x^i - l_y^j \right\|^2 / \delta} & II, \\ 1, \text{if} l_x^i = l_y^j; 0, if l_x^i \neq l_y^j & III, \end{cases} \tag{9}$$

where l defines the label vector of the sample, and each element S_{ij} of the similarity matrix represents the similarity between X modal data i and Y modal data j. The next goal of this paper is to learn the compact hash coding of the optimal length for each model, so that these hash coding can perfectly represent the original multimode data and maintain the semantic similarity of multimode data sets. This paper calculates the similarity of different modal data in potential space by referring to Ref. [7] and assumes that there is a common potential abstract semantic space V between multimodal data, in which multimodal data can be queried and retrieved directly. And, each modal hash coding is projected into the potential abstract semantic space in the following form:

$$M_1 : B_X \xrightarrow{W_1} V_X M_2 : B_Y \xrightarrow{W_2} V_Y. \tag{10}$$

In the space V, the similarity between data can be calculated according to the relation of the inner product, which is defined as follows:

$$\widetilde{S} = V_X^T V_Y = \left(W_1 B_X \right)^T \left(W_2 B_Y \right) = B_X^T W_1^T W_2 B_Y. \tag{11}$$

Remembering $W = W_1^T W_2$, we do not need to explicitly solve the existing form of each mode data in the potential abstract semantic space V, but only calculate the similarity W between the varied-length hash coding of each mode. The cross-modal retrieval objective function of the specific variable-length hash coding is defined as follows:

$$\begin{aligned} \min_{B_X, B_Y, W, P_X, P_Y} & \left\| B_X - P_X X \right\|_F^2 + \left\| B_Y - P_Y Y \right\|_F^2 \\ & + \left\| S - B_X^T W B_Y \right\|_F^2 s.t. B_X \in [-1, +1]^{q_1 \times n_1}, B_Y \in [-1, +1]^{q_2 \times n_2}. \end{aligned} \tag{12}$$

The first two terms of (12) are applied to, respectively, project the two-modal data into the hash coding space of the optimal lengths, and the last term indicates that the variable-length hash coding in the potential space still maintains the semantic similarity relation of the original multimodal data. The corresponding projection matrixes P_X, P_Y, hash coding B_X, B_Y, and correlation matrix W can be solved simultaneously through optimization.

3.2. Model Solution Procedure. To simplify the difficulty of solving hash coding, the prior art converts binary constraint conditions of hash coding into solving continuous real-valued problems and then obtains approximate hash coding through symbolic functions [10–12]. However, the solved hash coding has essential defects and cannot represent the original multimodal data effectively. The binary constraint condition of hash coding is always maintained in the solving process of this subsection. When the objective function is solved, the variables B_X, B_Y, W, P_X, P_Y of simultaneous solution are nonconvex and difficult to solve. Therefore, this paper first solves one of the variables and fixed the remaining variables, and then solves the other variables in this way. All variables are solved by iteration until the objective function tends to converge.

(a) Fix other variables and resolve P_X, P_Y. Therefore, the objective function can be simplified in the following form:

$$\min_{P_X} \left\| B_X - P_X X \right\|_F^2 \min_{P_Y} \left\| B_Y - P_Y Y \right\|_F^2. \tag{13}$$

Therefore, the analytical formulae can be calculated by regression formula, respectively,

$$\begin{aligned} P_X &= B_X X^T \left(X X^T \right)^{-1}. \\ P_Y &= B_Y Y^T \left(Y Y^T \right)^{-1}. \end{aligned} \tag{14}$$

(b) Fix other variables and resolve W. The objective function can be simplified in the following form:

$$\min_W \left\| S - B_X^T W B_Y \right\|_F^2. \tag{15}$$

It is obvious that (15) is a bilinear regression model, and the analytical formula is as follows:

$$W = \left(B_X B_X^T \right)^{-1} B_X S B_Y^T \left(B_Y B_Y^T \right)^{-1}. \tag{16}$$

(c) Fix other variables and resolve B_X. The objective function can be simplified in the following form:

$$\begin{aligned} \min_{B_X} & \left\| B_X - P_X X \right\|_F^2 + \left\| S - B_X^T W B_Y \right\|_F^2 s.t. B_X \\ & \in [-1, +1]^{q_1 \times n_1}. \end{aligned} \tag{17}$$

Because of the two-value constraint, it is complicated to resolve directly. Therefore, in this paper, the variable B_X is solved successively, i.e., when solving a row vector of B_X, the remaining row vectors are fixed first, and then the other row vectors are solved iteratively. (17) can be further transformed into (18).

$$\begin{aligned} \min_{B_X} & \left\| B_X \right\|_F^2 - 2Tr\left(B_X^T P_X X \right) + \left\| P_X X \right\|_F^2 + \| S \|_F^2 \\ & - 2Tr\left(B_X^T W B_Y S^T \right) + \left\| B_X^T W B_Y \right\|_F^2 s.t. B_X \\ & \in [-1, +1]^{q_1 \times n_1}. \end{aligned} \tag{18}$$

Because of the binary constraint, it is obvious that the first term is a constant, i.e., $\| B_X \|_F^2 = q_1 * n_1$. If constant terms and irrelevant variables B_X are removed, (18) can be rewritten into a more concise form.

$$\min_{B_X} \left\| D B_X \right\|_F^2 - 2Tr\left(B_X^T Q \right) s.t. B_X \in [-1, +1]^{q_1 \times n_1}, \tag{19}$$

where $D = B_Y^T W^T$, $Q = (W B_Y S^T + P_X X)$ and $Tr(\ldots)$ are the trace of the solution matrix. After deformation, the solution of (19) has a relationship with the solution of the objective function in Ref. [16], so this paper refers to its solution process. When solving the i-th row vector z^T of B_X, let B_X' be the matrix B_X after row vector deletion z^T, p^T defines the i-th row vector of Q, Q' represents the matrix Q after row vector deletion p^T, d defines the i-th column vector of D, and D' represents the matrix D after column vector deletion d, and then refer to the solution results in Ref. [16].

$$z = \mathrm{sign}\left(p - B_X' D'^T d\right). \tag{20}$$

The i-th row vector of B_X can be resolved, and then the remaining row vectors can be solved via a similar procedure.

(d) Fix other variables and resolve B_Y.

In the process of solving B_Y, it is similar to solving B_X, so readers can refer to the solution method of B_X for a detailed solution of B_Y.

3.3. Algorithm Description. To project hash coding into the optimal space for comparison, measurement, and retrieval, the associated transformation matrix W is introduced into the cross-modal retrieval model of variable-length hash coding on the base of the GSPH model, and then the similarity between data can be compared in the potential space through W. Subsection 2.2 provides the solution process of each variable in the model, and the overall training steps for the model are shown in Algorithm 1.

According to the proposed training process, the projection matrix of each mode can be calculated separately, and then the corresponding hash coding can be solved by a symbolic function. For query sample x' or y', the corresponding hash coding generation method is $b' = \mathrm{sign}\,(P_X x')$ or $b' = \mathrm{sign}\,(P_Y y')$. To improve the accuracy of generating corresponding hash coding, the query sample pair information (x', y') of these two modes can be used to generate hash coding simultaneously. If the final hash coding is expected to exist in the hash coding space of the X mode, then $b' = \mathrm{sign}\,(P_X x' + \theta W P_Y y')$. If the final hash code is desired to exist in the hash coding space of the Y mode, then $b' = \mathrm{sign}\,(P_Y y' + \theta W^T P_X x')$, where θ is a non-negative equilibrium parameter. The overall testing steps for the model are summarized in Algorithm 2.

3.4. Time Complexity. The time complexity of the cross-modal retrieval algorithm in this section is mainly composed of computation-related variables. In the training phase, the time of each iteration is consumed in updating the projection matrixes P_X, P_Y, transformation matrix W, and corresponding hash coding matrixes B_X, B_Y, in which these variables are calculated by (14) and (16), and (17), respectively, and the corresponding calculation time complexity is $\mathrm{O}\,(d^2qn), \mathrm{O}\,(q^2n^2), \mathrm{O}\,(dq^2n)$. Therefore, the total time complexity of the proposed model is $\mathrm{O}\,((d^2 + qn + dq)qnT)$, where T represents the total number of iterations, where $d = \max\,(d_1, d_2), q = \max\,(q_1, q_2), n = \max\,(n_1, n_2)$. More specially, d_1, q_1, and n_1 are the original dimension, hash length, and the total number of samples of X mode data, respectively, and d_2, q_2, and n_2 are the original dimension, hash length, and the total number of samples of Y mode data, respectively. Once the training process is end, the time and space complexity for generating a new sample is $\mathrm{O}\,(dq)$.

3.5. Application Scenario. The cross-modal retrieval model can be easily extended to the scenarios of three or more modal data, assuming that $m\,(m > 2)$ modal data, then the cross-modal retrieval model of variable-length hash coding for m modal data is defined as follows:

$$\min_{B_i, W^{(i,j)}, P_i} \sum_{i=1}^{m} \left\| B_i - P_i X_i \right\|_F^2 + \sum_{i,j}^{m} \left\| S^{(i,j)} - B_i^T W^{(i,j)} B_j \right\|_F^2 \tag{21}$$
$$s.t. B_i \in [-1, +1]^{q_i \times n_i}.$$

The first item in (21) represents the hash code mapping of all modal data into the optimal length, and the second item represents the semantic relationship preservation between the hash coding of each mode and another modal hash coding. The process of model optimization and query sample hash coding generation can follow the way of two-modal data scenarios.

4. Results and Discussion

4.1. Data Sets and Performance Metrics. To verify the validity of the model, the commonly used WIKI data set, NUS-WIDE data set, and MIRFlickr data set are selected for the cross-modal retrieval. In addition, the precision-recall and Mean Average Precision (MAP) index are used to measure model performance as shown in Refs. [11–13].

WIKI data set is collated from Wikipedia page [7], and each image has the corresponding description text, in which each text contains no less than 70 words. The data sets belong to a single-label data, and there are 10 categories, each image or text belongs to one of these categories, and images or texts belonging to the same category are considered to have similar semantic information. There exist 2866 samples (2173 training sets and 693 test sets), in which image data is represented by 128-dimensional Scale Invariant Feature Transform (SIFT) features and text data by 10-dimensional Latent Dirichlet Allocation (LDA) features.

NUS-WIDE data set is collected and sorted from the Internet by the National University of Singapore [18], which regulates 269,648 images and explanatory annotations accomplished by about 5,000 people. Each sample belongs to multilabel data, which is eventually divided into 81 categories. Due to the sample numbers of some categories differ greatly in this paper, just as Refs. [10, 11], the top 10 categories with many samples are firstly selected, and finally 186,577 text-image pairs have been achieved. Text and image are considered similar, if there is at least one of the same category attributes. Subsequently, 1% of the data (about 1866) are randomly selected as the test set and 5000 samples as the training set. The images of the NUS-WIDE data set are

```
    Input: Training datasets $X/Y$ and label matrix $L_X/L_Y$; Initialized association matrix $W$; Initialized variable-length hash $B_X, B_Y$;
    Initialized iteration control parameter $T$
    Output: Variables $B_X, P_X, B_Y, P_Y, W$
    Procedure:
(0) Applying label matrix $L_X, L_Y$ and (9) to construct a semantic similarity matrix $S$
(1) $iter = 0$;
(2) while $iter < T$ do
(3) According to (14), update the dictionary projection matrix $P_X, P_Y$;
(4) According to (16), update the association matrix $W$;
(5) According to equation (18) and the detailed solving process in Ref. [14], the hash code of variable length is updated one line at a
    time and finally updated as a whole $B_X, B_Y$;
(6) If the objective function (12) tends to converge, and stop the iteration; otherwise, skip to step (2);
(7) End while
```

Algorithm 1: Training produce of proposed method.

```
     Input: Testing datasets $X'/Y'$; trained $f(\cdot)$, $g(\cdot)$ and $\mathbf{W}$.
     Output: The top $n$ cross-modal data matching the samples to be retrieved.
     Procedure:
(1)  if input independent $x'$ or $y'$ then
(2)    compute the corresponding hash code by $b' = \mathrm{sign}(f(x'))$ or $b' = \mathrm{sign}(g(y'))$;
(3)  end if
(4)  if input paired $(x', y')x'$ then:
(5)    if hash code exists in space of Y data:
         $b' = \mathrm{sign}(g(y')) + W^{\mathrm{T}} f(x')$;
(6)    else:
         $b' = \mathrm{sign}(f(x')) + W^{\mathrm{T}} g(y')$;
(7)    end if
(8)  end if
(9)  Calculate the Hamming distance between the hash code b' and the hash codes of all samples in the retrieval database
(10) Sort the distances calculated in ascending order, and return the first $n$ samples.
```

Algorithm 2: Testing produce of the proposed method.

represented by 500-dimensional SIFT features and the text data by the word frequency of 1000 dimensions.

MIRFlickr data set originated from the Flickr website, which contains 25000 images and corresponding manually annotated text information [19]. Just as Ref. [11], we have deleted some data without labels or with less than 20 times of labeled words, and finally 16,738 samples are divided into 24 categories. Each image text pair belongs to multicategory data, which contains at least one category label. This paper selects 5% data as a test set and 5000 samples as the training set. Images in the data set are represented by 150-dimensional edge histograms and text by 500-dimensional vectors. The evaluation criteria are defined as follows:

$$\text{Accuracy: } P(N) = \frac{n}{N} \times 100\%,$$
$$\text{Recall: } R(N) = \frac{n}{N_r} \times 100\%, \tag{22}$$

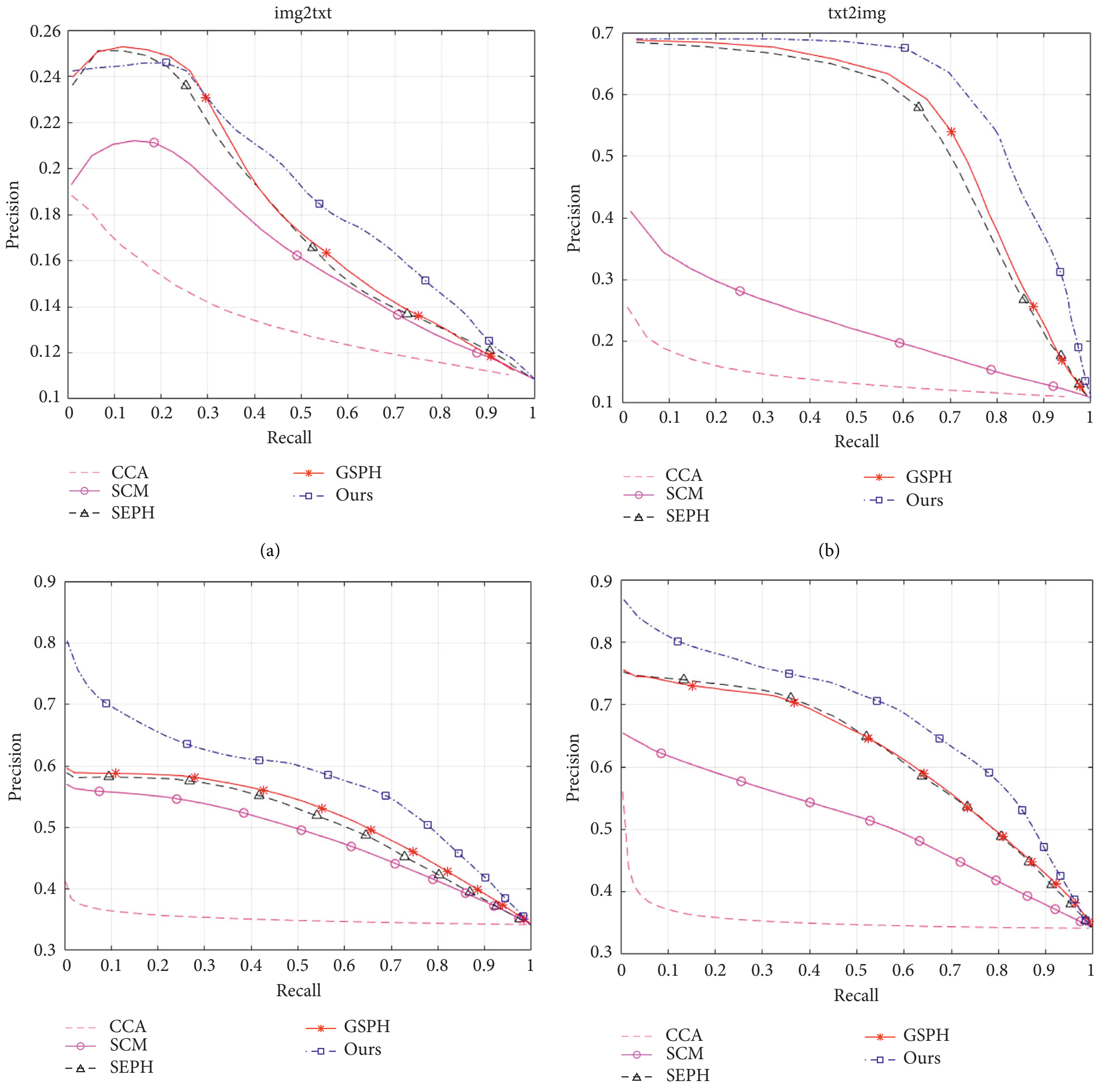
where n represents the number of relevant samples among N results stemming from the retrieval and N_r defines the number of samples related to query samples in the whole database.

Average Precision (AP) indicator calculation: Given a query sample and the first R returned results, the AP calculation equation of this sample is defined as follows:

$$AP = \frac{1}{K} \sum_{r=1}^{R} P(r)\delta(r), \tag{23}$$

where K represents the number of retrievably returned results related to query samples, and $P(r)$ defines the accuracy of the returned first r retrieval results. If the r-th retrieval result is related to the query sample, $\delta(r)$ is 1; otherwise, $\delta(r)$ is 0. Finally, the AP average value of all query samples is solved, which is the MAP index to evaluate the overall search performance.

4.2. Benchmark Algorithm. In this subsection, the various multimodal data are preprocessed according to the method represented in Ref. [16], i.e., the distance between sample points and randomly selected reference points is calculated. Then the discrete supervised hash model is used to initialize the hashing coding of each mode. To highlight the importance of the label matrix in the process of optimization, the



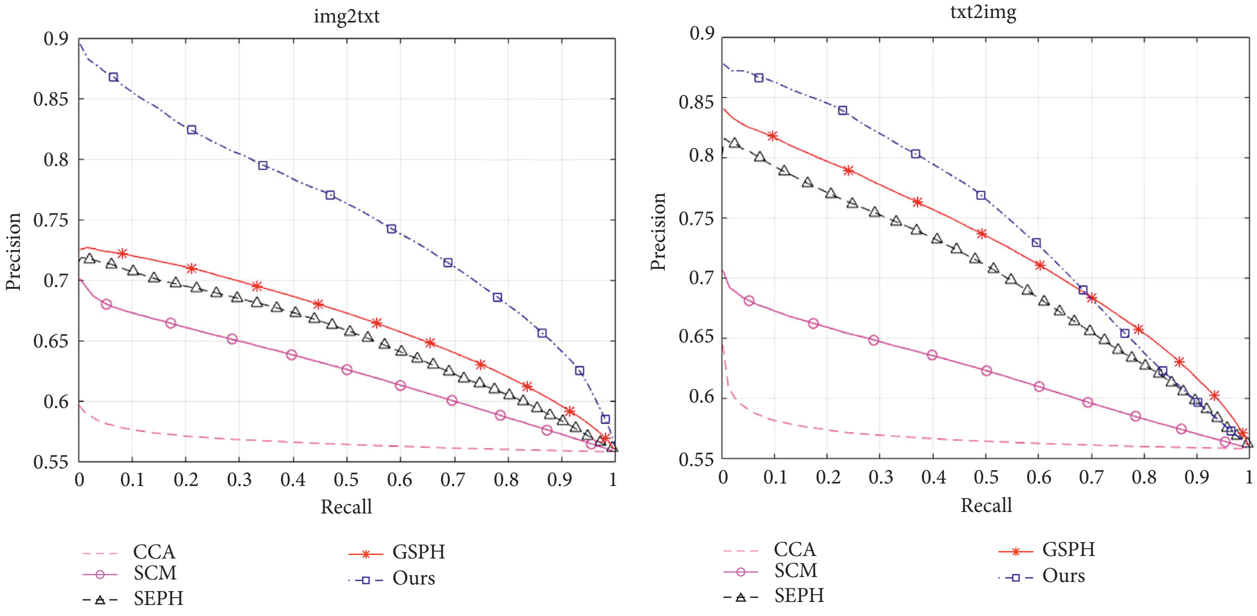
(a)

(b)

(c)

(d)

Figure 1: Continued.



(e)

(f)

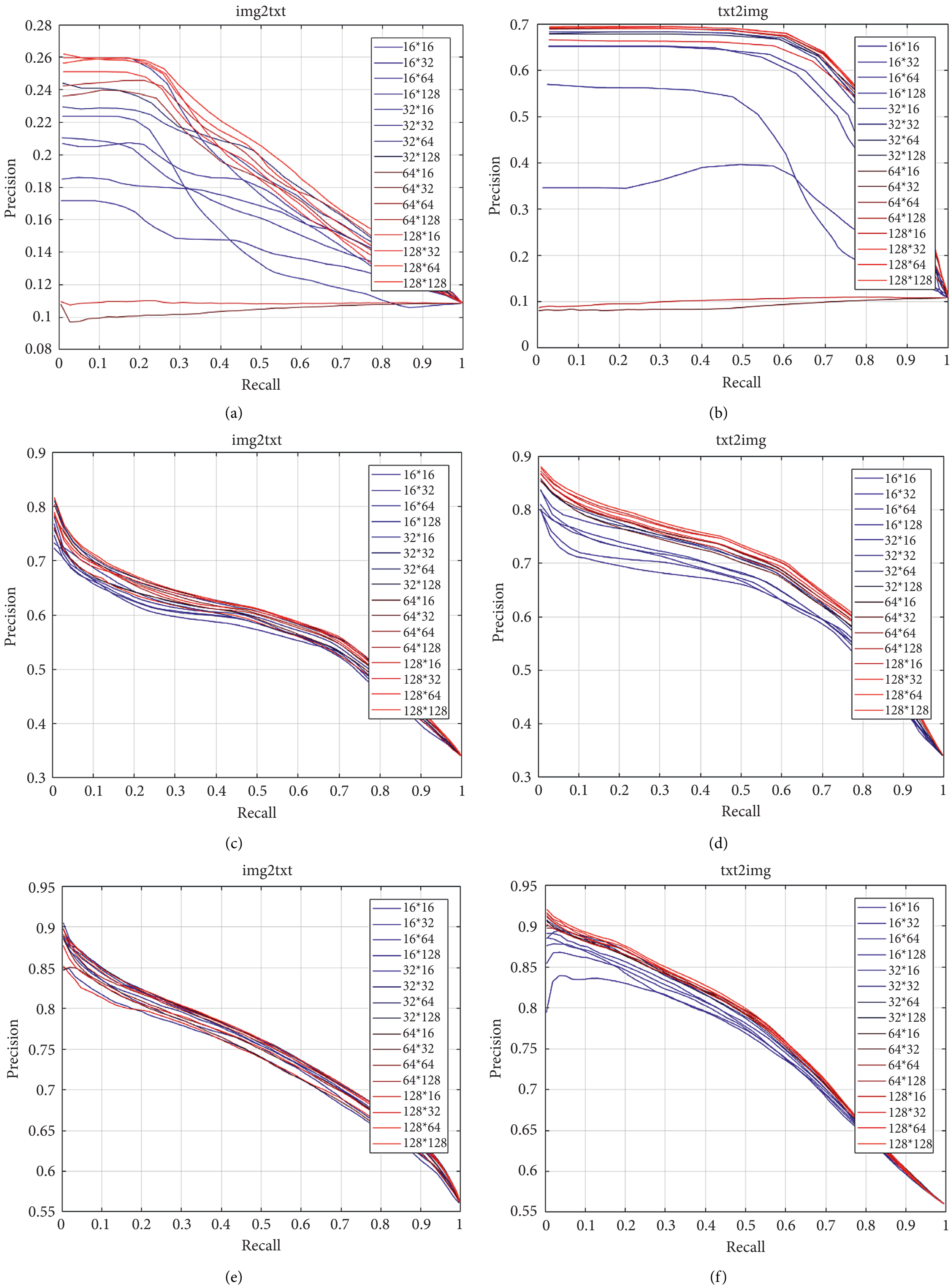
Figure 1: Precision rate and recall rate of different methods for the different data sets: (a) WIKI (img2txt), (b) WIKI (txt2img), (c) NUS-WIDE (img2txt), (d) NUS-WIDE (img2txt), (e) MIRFlickr (txt2img), and (f) MIRFlickr (txt2img).

label matrix of all data is enlarged by 10 times. In addition, CCA, a typical correlation analysis method commonly used in the field of cross-modal retrieval, and the cross-modal retrieval algorithm based on semantic correlation hash coding in recent years are selected as a comparative experiment. These hashing cross-modal retrieval models are SCM, SEPH, and GSPH, respectively, and the comparison experiments proposed in this paper are implemented in MATLAB with the help of the parameters set in the original text. Both SEPH and GSPH models include two methods to learn hash functions: (1) training hash functions SEPH_rnd and GSPH_rnd based on randomly selected samples; (2) training hash functions SEPH_knn and GSPH_knn based on selecting samples through clustering. The experiment shows that the performance of the hash function obtained by these two training methods is the same. Therefore, the first method, randomly selected samples, is selected to train the hash functions of both SEPH and GSPH models in the comparative experiment. Moreover, the two different methods in the SCM model are SCM_seq and SCM_orth, and the experiment results show that the former is generally superior to the latter; therefore the former is used as a comparative experiment [10].

4.3. Experimental Results. This subsection presents the experimental results of cross-modal retrieval on the WIKI dataset, NUS-WIDE dataset, and MIRFlickr dataset. The following cross-modal retrieval tasks include image retrieval text and text retrieval image, and these two retrieval tasks are analyzed in detail. Figure 1 shows the curves of retrieval accuracy rate and recall rate on three kinds of data sets. To facilitate the comparison with the benchmark algorithm, both image and text are projected into equal-length hash coding space (64 bits). It can be seen from Figure 1 that the performance of the method proposed in this paper is generally superior to that of the comparison method, although the front part of the curve (subgraph (a) of Figure 1) in the image retrieval text task on the WIKI dataset is slightly lower than that of SEPH and GSPH methods. However, it can be seen from the subgraph (a) of Figure 2 that the effect of the optimal hash coding combination length in this paper is slightly higher than that of SEPH and GSPH methods. It can also be seen from Figure 1 that for the other two groups of multilabel data, the effect of this paper has been improved more than that of the comparison methods, due to the model in this paper being more suitable for multilabel data sets than the CCA, SCM, SEPH, and GSPH models.

The MAP index of image retrieval text and text retrieval image of each method is presented in detail in Tables 1 and 2, respectively, and the highest MAP value of each column is marked black. To compare the effects of CCA and other methods, this paper projected data into subspaces of different dimensions to observe the influence of CCA methods. Tables 1 and 2 show that the MAP value of the proposed method and other hash coding methods increases slightly as the length of hash coding increases. As can be seen from the numerical part marked black in the table, the MAP value of the proposed method is superior to that of the comparison method, no matter in the image retrieval text task or the text retrieval image task. Given that the hash coding length is 64 bits, this paper improves about 15%, 10%, and 13% in the image retrieval text task on WIKI, NUS-WIDE, and MIR-Flickr data sets, and about 12%, 11%, and 5% in the text retrieval image task compared with the GSPH method.

Figure 2 shows the experimental results of different length combinations for the hash coding proposed in this



(a)

(b)

(c)

(d)

(e)

(f)

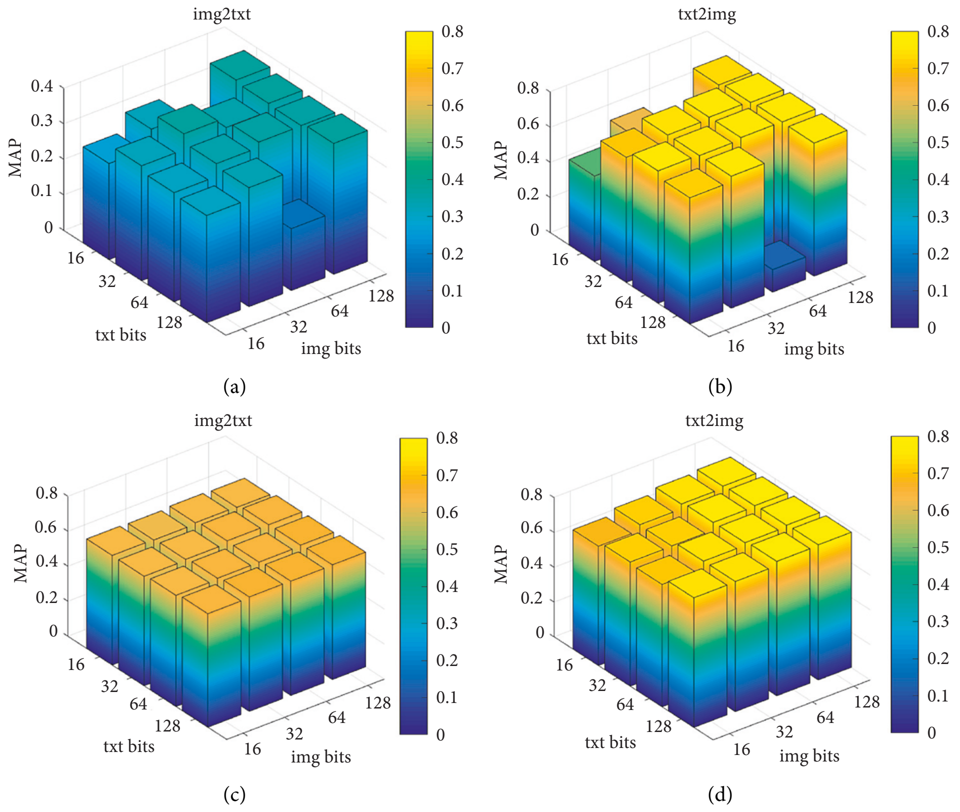
Figure 2: Precision rate and recall rate of different hash coding length combinations: (a) WIKI (img2txt), (b) WIKI (txt2img), (c) NUS-WIDE (img2txt), (d) NUS-WIDE (img2txt), (e) MIRFlickr (txt2img), and (f) MIRFlickr (txt2img).

Table 1: MAP image retrieval text img2txt.

	WIKI data set				NUS-WIDE data set				MIRFlickr data set			
	16	32	64	128	16	32	64	128	16	32	64	128
CCA	0.184	0.170	0.150	0.140	0.373	0.366	0.361	0.358	0.579	0.574	0.571	0.568
SCM	0.234	0.241	0.246	0.257	0.501	0.542	0.553	0.551	0.610	0.631	0.647	0.641
SEPH	0.276	0.296	0.300	0.313	0.560	0.578	0.582	0.581	0.671	0.652	0.681	0.648
GSPH	0.272	0.290	0.305	0.307	0.571	0.582	0.585	0.593	0.665	0.676	0.687	0.692
VHC	0.271	0.368	0.351	0.369	0.627	0.632	0.644	0.656	0.766	0.772	0.778	0.779

Table 2: MAP text retrieval image txt2img.

	WIKI data set				NUS-WIDE data set				MIRFlickr data set			
	16	32	64	128	16	32	64	128	16	32	64	128
CCA	0.168	0.159	0.154	0.150	0.371	0.365	0.362	0.360	0.579	0.574	0.572	0.570
SCM	0.226	0.246	0.249	0.253	0.535	0.540	0.542	0.539	0.615	0.624	0.628	0.631
SEPH	0.631	0.658	0.659	0.669	0.683	0.695	0.693	0.708	0.710	0.744	0.727	0.744
GSPH	0.645	0.663	0.671	0.674	0.681	0.697	0.686	0.714	0.726	0.742	0.748	0.764
VHC	0.487	0.748	0.751	0.757	0.686	0.715	0.761	0.776	0.766	0.780	0787	0.791



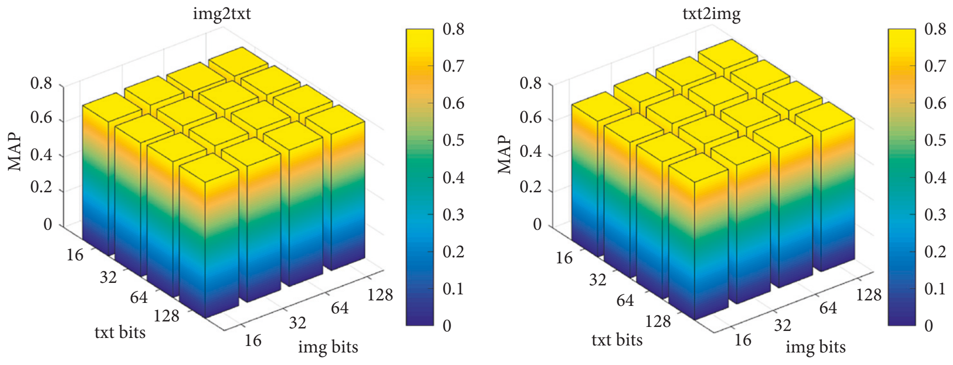
(a)

(b)

(c)

(d)

Figure 3: Continued.



(e)

(f)

Figure 3: 3-D histogram of MAP index of different hash coding length combinations: (a) WIKI (img2txt), (b) WIKI (txt2img), (c) NUS-WIDE (img2txt), (d) NUS-WIDE (img2txt), (e) MIRFlickr (txt2img), and (f) MIRFlickr (txt2img).

paper (image hash coding length ∗ text hash coding length). To show the variation tendency of different hash length combinations, the curve colors of hash coding length combinations from 16 ∗ 16 to 128 ∗ 128 gradually change from dark blue, light blue, light red, and then dark red as shown in Figure 2. Generally speaking, with the growth of image hash coding, the cross-modal retrieval effect also becomes better, especially for the subgraphs (d) and (*f*) of Figure 2. In addition, Figure 2 also shows that the cross-modal retrieval model of variable-length hash coding in this paper has a more significant impact on WIKI data sets.

From the MAP three-dimensional histogram in Figure 3, it can be seen that the same and fixed hash code length cannot be set for all datasets. To be special, the optimal hash code combination is 48 ∗ 64(text ∗ image) for the img2txt task on the NUS-WIDE dataset. But the optimal hash code length combination is 32 ∗ 64 (text ∗ image) for the img2txt task on the MIRFlickr dataset to implement the img2txt task. The reason is that the text information of NUS-WIDE is richer and more hash codes are needed to represent text features. From another point of view, for some retrieval tasks, using a shorter hash code length can also achieve a comparable retrieval effect. Thus, we can conclude that using a variable-length hash code can balance the data redundancy and retrieval accuracy.

5. Conclusion

In this paper, a variable-length hash coding-based cross-modal retrieval algorithm is first proposed, which projects multimodal data into the optimal hash length space of each modal data. The similarity matrix of multimodal data is constructed according to the label matrix of each mode, and the semantic similarity relationship of the original data is still guaranteed after the multimodal hash coding is projected into the potential abstract semantic space. Then the binary constraint condition of the hash coding is always maintained in the process of optimizing the model, so that the learned multimode hash coding can better represent the original multimode data. A wide variety of experiments on WIKI datasets, NUS-WIDE datasets, and MIRFlickr datasets show that the performance of the proposed method is generally superior to that of the correlation benchmark algorithms. Therefore, the method in this paper is feasible and effective. Compared with the deep learning-based hashing methods, the retrieval performance is relatively low. Thus, in our future work, we will embed the proposed similarity matrix into the deep learning-based method to further improve the retrieved accuracy and effectively measure the relationship among multiple source data.

Data Availability

The datasets used and/or analyzed during the current study are available from the author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant 61871062), Science and Technology Research Program of Chongqing Municipal Education Commission (Grants KJQN201800615 and KJQN201900609), Natural Science Foundation of Chongqing, China (Grants cstc2020jcyj-zdxmX0024 and cstc2021jcyj-msxm2025), Key Project of Science and Technology of Chongqing Municipal Education Commission (Grant KJZDK201901203), and University Innovation Research Group of Chongqing (Grants CXQT20017 and CXQT20024).

References

[1] J. Xiong, J. Ren, L. Chen et al., “Enhancing privacy and availability for data clustering in intelligent electrical service of IoT,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2019.

[2] Y. Peng, X. Huang, and Y. Zhao, “An overview of cross-media retrieval: concepts, methodologies, benchmarks, and

challenges," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2372–2385, 2018.

[3] J. Xiong, R. Bi, Y. Tian, X. Liu, and D. Wu, "Toward lightweight, privacy-preserving cooperative object classification for connected autonomous vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2787–2801, 2022.

[4] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2021.

[5] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: an overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, Dec, 2004.

[6] X. Fu, K. Huang, E. E. Papalexakis et al., "Efficient and distributed generalized canonical correlation analysis for big Multiview data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2304–2318, 2019.

[7] J. Costa Pereira, E. Coviello, G. Doyle et al., "On the role of correlation and abstraction in cross-modal multimedia retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 521–535, Mar. 2014.

[8] D. Mandal and S. Biswas, "Generalized coupled dictionary learning approach with applications to cross-modal matching," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3826–3837, 2016.

[9] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[10] F. Taherkhani, V. Talreja, M. C. Valenti, and N. M. Nasrabadi, "Error-corrected margin-based deep cross-modal hashing for facial image retrieval," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, no. 3, pp. 279–293, Jul. 2020.

[11] Z. D. Chen, C. X. Li, X. Luo, L. Nie, W. Zhang, and X. S. Xu, "SCRATCH: a scalable discrete matrix factorization hashing framework for cross-modal retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2262–2275, Jul, 2020.

[12] D. Wang, X. Gao, X. Wang, L. He, and B. Yuan, "Multimodal discriminative binary embedding for large-scale cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4540–4554, 2016.

[13] X. Xu, F. Shen, Y. Yang, H. T. Shen, and X. Li, "Learning discriminative binary codes for large-scale cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2494–2507, May 2017.

[14] D. Mandal, K. N. Chaudhury, and S. Biswas, "Generalized semantic preserving hashing for N-label cross-modal retrieval," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2633–2641, Honolulu, Hi, USA, July 2017.

[15] F. Shen, X. Zhou, Y. Yang, J. Song, H. T. Shen, and D. Tao, "A fast optimization method for general binary code learning," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5610–5621, Dec, 2016.

[16] J. Gui, T. Liu, Z. Sun, D. Tao, and T. Tan, "Fast supervised discrete hashing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 490–496, 2018.

[17] J. Xiong, X. Chen, Q. Yang, L. Chen, and Z. Yao, "A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2347–2360, 2020.

[18] T. S. Chua, J. Tang, and R. Hong, "NUS-WIDE: a real-world web image database from National University of Singapore," in *Proceedings of the 2009 Proceedings of the ACM International Conference on Image and Video Retrieval*, pp. 1–9, ACM, 2009.

[19] M. J. Huiskes and M. S. Lew, "The MIR Flickr retrieval evaluation," in *Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*, pp. 39–43, ACM, Santorini Island, Greece, July 2008.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 8951961, 41 pages
https://doi.org/10.1155/2022/8951961

Review Article

Machine and Deep Learning for IoT Security and Privacy: Applications, Challenges, and Future Directions

Subrato Bharati and Prajoy Podder

Institute of Information and Communication Technology (IICT), Bangladesh University of Engineering and Technology (BUET), Dhaka 1205, Bangladesh

Correspondence should be addressed to Subrato Bharati; subratobharati1@gmail.com

Received 19 March 2022; Revised 19 June 2022; Accepted 26 July 2022; Published 27 August 2022

Academic Editor: Jinbo Xiong

The integration of the Internet of Things (IoT) connects a number of intelligent devices with minimum human interference that can interact with one another. IoT is rapidly emerging in the areas of computer science. However, new security problems are posed by the cross-cutting design of the multidisciplinary elements and IoT systems involved in deploying such schemes. Ineffective is the implementation of security protocols, i.e., authentication, encryption, application security, and access network for IoT systems and their essential weaknesses in security. Current security approaches can also be improved to protect the IoT environment effectively. In recent years, deep learning (DL)/machine learning (ML) has progressed significantly in various critical implementations. Therefore, DL/ML methods are essential to turn IoT system protection from simply enabling safe contact between IoT systems to intelligence systems in security. This review aims to include an extensive analysis of ML systems and state-of-the-art developments in DL methods to improve enhanced IoT device protection methods. On the other hand, various new insights in machine and deep learning for IoT securities illustrate how it could help future research. IoT protection risks relating to emerging or essential threats are identified, as well as future IoT device attacks and possible threats associated with each surface. We then carefully analyze DL and ML IoT protection approaches and present each approach's benefits, possibilities, and weaknesses. This review discusses a number of potential challenges and limitations. The future works, recommendations, and suggestions of DL/ML in IoT security are also included.

1. Introduction

Internet of Things (IoT) considers the interconnection between several devices, i.e., industrial systems, intelligent sensors, autonomous vehicles, mechanisms and terminals, mechanical systems, and so on [1, 2]. Alternatively, it can be termed as a network of physical things or objects that are connected with limited communication, computation, and storage capabilities along with embedded electronics (i.e., sensors and actuators), connectivity of network, and software that enables these things to exchange, analyze, and collect data [3]. IoT relates to our everyday life, extending from smart devices in the household, i.e., smart meters, IP cameras, smoke detectors, smart adapters, smart refrigerators, smart bulbs, AC, smart ovens, and temperature sensors, to more advanced devices, for example, heartbeat detectors, radio-frequency identification (RFID) devices, accelerometers, IoT in automobiles, sensors in rooms, and so on [4]. Several services and applications referred to by the IoT are emerging in personal healthcare, home appliances, critical agricultural infrastructure, and the military [1].

The massive scale of IoT networks introduces latest issues, including the management of these devices, the complete volume of data, communication, storage, processing, and security and privacy concerns, among others. There has been substantial research into various components of the IoT, such as architecture, communication, applications, protocols, security, and privacy, to name a few. The guarantee of security and privacy and user satisfaction are the cornerstones of the commercialization of IoT technology. The fact that the IoT makes use of empowering technologies including cloud computing (CC), software-defined networking (SDN), and edge computing enhances the number of dangers that attackers can encounter. As a result, monitoring security in the development of IoT infrastructure has become challenging and complex. Solutions must consist of wide-ranging considerations to fulfill the security challenges [5]. On the other hand, IoT systems are frequently put to use in an unprepared state. As a result, a fraudster can use wireless networks to connect to IoT devices and gain physical access to confidential data. Complexity and integrative arrangements characterize IoT systems. In light of the proliferation of connected devices, it might be difficult to meet the ever-evolving security standards for the IoT. In order to provide the necessary level of security, solutions need to consider the system as a whole. However, most IoT devices can function independently of human input. Someone without permission may thus acquire physical access to these devices [6–8].

Furthermore, the IoT system introduces novel attack surfaces. The interconnected and interdependent systems cause these types of attacks to surface. Accordingly, the security of IoT systems is faced with a higher risk than the security of other traditional computing devices. The outdated computing systems will be fruitless for these IoT schemes [9–11].

IoT systems ought to instantaneously consider security, energy efficiency, IoT software applications, and data analytics at the time of related tasks as a sign of the wide-ranging application [7]. This expansion offers an innovative scope for scholars from the interdisciplinary research program to consider recent challenges in the IoT schemes from various perceptions. However, the large-scale as well as cross-cutting nature of IoT devices and the many components engaged in their implementation have created new security issues. The IoT devices' characteristic presents various security issues. Additionally, the stages of IoT provide a massive number of useful information. If this information is not analyzed and transmitted securely, a crucial privacy gap may occur. Applying related security mechanisms, such as authentication, encryption, application security, network security, and access control, is inadequate and challenging for enormous schemes with numerous associated schemes. Every portion of the IoT platform contains intrinsic vulnerabilities. For example, a special kind of botnet like "Mirai" has newly affected extensively distributed denial of service (DDoS) attacks by using IoT systems [9, 12].

For extensive methods with multiple connected devices and each module of the method having inherent vulnerabilities, it is difficult and inadequate to apply existing security protection mechanisms like encryption, identity verification, application security, access control, and computer security [9]. For example, the "Mirai" botnet has lately been responsible for large-scale DDoS attacks by abusing IoT devices. For the IoT ecosystem, existing security methods need to be improved. However, the deployment of cryptographic functions against a particular security issue is rapidly overtaken by new categories of attack developed by the attackers in order to bypass current remedies. Address-spoofed source IPs are commonly applied in magnified DDoS attacks to hide the location of attacks from the targeted organization's security teams. As a result of the vulnerabilities in IoT systems, more sophisticated and catastrophic attacks such as Mirai might be predicted. On account of the wide range of IoT scenarios and applications, knowing which security solutions are best for IoT systems is not easy. As a result, the focus of the study should be on devising appropriate IoT security methods [12, 13].

While security and privacy are interconnected, security may exist without privacy, but privacy cannot exist without security. Security safeguards the availability of information, integrity, and confidentiality, while privacy is more detailed about privacy rights in relation to personal information. Regarding the processing of personal data, privacy takes precedence, but information security entails preventing illegal access to information assets. Personal data may relate to any information about a person, including names, credentials, addresses, social security numbers, bank account data, and so on.

A number of ways have been suggested to address the boundary between security and privacy concerns in DL and ML. Homomorphic encryption, differential privacy, trusted execution, and secure multiparty computing environment are the four most often used DL and ML privacy technologies. This technique uses differential privacy to prevent the adversary from figuring out which instances were utilized to build the target model. Training and testing data are protected by safe multiparty computing and homomorphic encryption. For sensitive data security and training code, trusted execution environments leverage hardware-based security and isolation. These approaches, on the other hand, greatly increase the computing burden and need a tailored approach for each type of neural network. DL or ML privacy concerns are yet to be addressed in a way that is accepted worldwide. To protect against adversarial attacks, a wide variety of security measures have been suggested, which may be divided into three categories: input preprocessing, strengthening the model's resilience, and malware detection. Preprocessing's goal is to lessen the model's reliance on immunity by doing operations such as picture transformation, randomization, and denoising that do not often need model update or retraining. Introducing regulation, feature denoising, and adversarial training as well as other techniques to strengthen the model's robustness via model retraining and change falls under the second group. Adaptive denoising and image transformation detection are the examples of third-category detection mechanisms that may be implemented before the first layer of the model. To the best of our knowledge, no defense strategy exists that can entirely protect against adversarial cases despite the many defensive mechanisms that have been offered. To counter hostile instances, adversarial training is currently the most effective technique. For poisoning attacks, there are two basic means of defense. The first is an outlier identification technique, which eliminates outliers from the relevant set. The second step is to enhance the neural network's ability to withstand contamination from poisoned samples.

1.1. Motivation and Scope.

Deep learning (DL) and machine learning (ML) are effective methods of data analytics as well as investigation to realize "abnormal" and "normal" behavior following how IoT devices and components interrelate with each other within the environment of IoT [14]. The IoT systems' input data can be investigated and collected to find out standard patterns of the interface, thus detecting malicious manners at initial stages. Furthermore, DL/ML techniques can be significant in identifying new threats, which are regular modifications of existing threats, as they can highly detect upcoming unknown threats by learning from previous attacks. As a result, IoT systems need to be able to move from secure communication between security-based intelligence and devices via ML/DL techniques for safe and efficient systems.

Several unique properties of IoT networks will be discussed in the following paragraphs.

Heterogeneity. Each item in an IoT network has unique features, communication protocols, and capabilities that all function together. Different communication paradigms and protocols (such as Ethernet or cellular), as well as varied hardware resource limits, might be used by the devices. On the one hand, this diversity allows devices to communicate across platforms, but on the other hand, it introduces additional obstacles to the network of IoT.

Proximity Communication. Additionally, IoT devices may communicate with one another without trusting on a central authority like base stations, which is an important feature. Dedicated short range communication (DSRC) and other point-to-point communication technologies are used in device-to-device (D2D) communication. Decoupling services and networks allows device-centric and content-centric communication, broadening the IoT service spectrum, whereas the conventional internet's design is more network-centric.

Massive Deployment. Massive deployment predicted that the existing internet's capabilities would be exceeded by the billions of devices linked to it and the internet. Massive IoT deployments are not without their own set of difficulties. Storage and architecture networks for intelligent devices, efficient protocols for data transfer, and proactive detection and protection of IoT-based devices from malicious attacks are only some of the difficulties that need to be addressed. A worldwide information and communication infrastructure that can be retrieved from everywhere as well as at any time is envisaged for IoT devices. How much is connection reliant on the kind of IoT service and application provided? For example, a swarm of sensors or a connected automobile may have a local connection, whereas critical infrastructure management and smart home access through mobile infrastructure may have global connectivity.

Low-Cost and Low-Power Communication. For optimal network operations, low-cost as well as ultra-low-power solutions are needed for the massive networking of IoT devices. For modern and critical IoT connectivity, self-healing and self-organization features are needed. Self-organizing networks ought to be implemented in these cases since relying on the network structure is not an option.

Low Latency and Ultra-Reliable Communication (LLURC). Remote surgical procedures, intelligent transportation, and industrial process automation systems all rely on the ability of IoT networks to reliably and quickly respond to real-time demands.

Safety. As a result of the enormous number of IoT devices linked to the internet, the private data exchanged via these systems may be at risk. Privacy and device security are also vital considerations. One of the most exciting aspects of IoT is its ability to make timely and intelligent choices based on the data it processes.

Dynamic Changing Network. An enormous number of IoT devices need proper management. These devices may behave dynamically. For example, when a device goes to sleep or wakes up, it is determined by various factors, including the software it is running.

The commercialization of IoT services and applications is heavily dependent on security and privacy. Various sectors, such as healthcare and business, have been impacted by security breaches that range from basic hacking to well-coordinated intrusions at the corporate level. Due to their restrictions and the environment in which they operate, IoT devices and apps face significant security issues. IoT security and privacy concerns have been thoroughly considered from different viewpoints, including communication privacy and security, architecture data security, and identity management as well as malware analysis [4]. Sections 3 and 5 explore more into the issues of security and the threat model.

According to Fernandes et al. [15], security challenges in IoT and conventional information technology (IT) devices are comparable and different. In addition, they also addressed privacy concerns. Software, hardware, networks, and applications are some of the most often cited points of comparison and contrast in this debate. IoT and conventional IT have a lot of things in common when it comes to security concerns. Despite this, the real concern of the IoT is the lack of available resources, which makes it challenging to implement advanced security measures in IoT networks. Improved algorithms and cross-layer architecture are also needed to address IoT privacy and security concerns. As part of an overall privacy and security method for IoT, current security solutions will be nominated for consideration, as well as new intelligent, resilient, scalable, and evolutionary methods to handle IoT security concerns.

ML implies intelligent procedures that utilize previous experiences or example data to understand how to maximize performance criteria. Algorithms that use machine learning to develop behavioral models on massive datasets are known as ML algorithms. Because of machine learning, computers can learn independently, even if no instructions are provided. The newly included data are fed into these models, which serve as a foundation for generating predictions about the future. AI, optimization, information theory, and cognitive science all have origins in ML, so it is a multidisciplinary field of study [16]. ML is no exception. Robotics, voice recognition, and other areas where people are unable to apply their skills, such as hostile environments, need the use of machine learning [17]. It may also be used when the answer to a particular issue evolves over time. To put this into context, Google utilizes ML to identify risks to mobile devices and apps running on the Android platform. Infected mobile devices may be scanned and cleaned using this tool. Macie, an Amazon tool that applies ML to organize as well as categorize data stored in Amazon's cloud storage, was also released recently. False positives and true negatives may occur even when ML methods are used correctly. As a result, if an incorrect prediction is produced, ML approaches need direction and change of the model.

Contrary to popular belief, with DL, a new kind of ML, the model is able to establish its accuracy of prediction. Prediction and classification tasks in novel applications of IoT with customized and contextual support might benefit from the self-service character of DL models. Moreover, the complete volume of data produced by IoT networks necessitates the use of DL and ML methods to offer intelligence to the systems. In addition, the IoT data created by DL and ML algorithms can be effectively exploited to make educated and intelligent choices by the IoT systems. The analyses of privacy, security, malware, and attack detection are just a few of the many applications for DL and ML. DL methods can also be employed in IoT systems to conduct identification tasks and complicated sensing to develop new apps and services that take into account real-time interactions among people, the physical environment, and smart devices.

Real-world uses of ML in security include the following:

(i) Different handwriting styles are used for character recognition in security encryption.

(ii) Recognition of faces in forensics: lighting, pose, occlusion (beard and glasses), hairstyle, makeup, etc.

(iii) Software and apps that contain malicious code need to be identified.

(iv) Behavior analysis is used to identify DDoS attacks on infrastructure. On the other hand, there are several difficulties associated with applying DL and ML in IoT applications. For example, designing an appropriate model for processing data from many IoT applications is challenging. In the same way, correctly classifying input data is likewise a complex undertaking. The use of little marked data in the learning process is also tricky. Using these models on IoT devices with limited processing and storage resources presents further difficulties [18]. Like essential infrastructure and real-time applications, DL and ML algorithms produce anomalies. IoT security solutions that use DL and ML must be thoroughly analyzed in this context.

1.2. Contributions. The main influences of this work are presented below:

(i) A review of various types of attacks with its example is discussed.

(ii) Comprehensive analysis of ML and latest developments in IoT defense DL methods: the most promising DL and ML algorithms are examined for IoT protection schemes, and their benefits, drawbacks, and implementations are addressed in the security of IoT systems. In addition, comparison and description tables are provided for DL and ML approaches for learning lessons.

(iii) A number of state-of-the-art applications of DL and ML in IoT security and privacy are illustrated.

(iv) We offer a taxonomy of the most recent IoT privacy and security solutions based on deep learning and machine learning techniques. Moreover, new insights of ML and DL in IoT securities are illustrated.

(v) Potential limitations, challenges, future directions, and suggestions of DL and ML are given to help the recent and future research.

The work is presented as follows. Literature reviews with their limitations and why this work is needed are illustrated in Section 2. Next, several IoT threats are illustrated in Section 3. After that applications of IoT security are described in Section 4. Section 5 describes the four levels of an IoT application. Moreover, Section 6 gives the DL and ML models, where we can find how to work with each ML and DL model in IoT security, and solutions are also described in Section 7. In Section 8, we can see a number of new insights into deep and machine learning for IoT security that can help future research works. Section 9 discusses challenges, limitations, and future directions. A number of suggestions and recommendations are presented in Section 10, and Section 11 gives the conclusion.

2. Literature Reviews

A number of surveys or reviews have covered IoT security to offer some guidelines for future challenges. Though several studies have looked at IoT security, none have focused on DL or ML applications for IoT security. Several works [19–25] have been reviewed for motivating and organizing the challenges in access control, authentication, application security, encryption, and network security in IoT environments. The survey in [26] provided a survey of IoT communication on security issues with its solutions. Another paper [27] emphasized IoT systems for intrusion detection.

Moreover, IoT frameworks for regulatory approaches and legal issues can determine security and privacy requirements [28]. The context of distributed IoT has also covered privacy and security in [29]. These works were also concerned with various challenges. Several issues must be found out, and the researchers assert that the distributed IoT method offers numerous advantages in terms of privacy and security. The survey in [30] described evolving threats and vulnerabilities in IoT devices, for example, threats of ransomware as well as security concerns. The authors in [31] concisely indicated the context of IoT using ML techniques concerning data security and privacy protection. This survey also described three challenges with respect to ML application in IoT environments (i.e., communication and computation overhead, partial state consideration, and backup security justifications). Numerous survey studies, including [31, 32], have examined the use of data mining and ML methods in cybersecurity to assist intrusion detection. Above all, they reviewed anomaly detections and misuse in cyberspace [32]. The methodology was based on several classes of AI (artificial intelligence) methods from the point of view of an IoT context, and the opportunities of applying those approaches in IoT environments were observed. The review in [3] also provided ML techniques in IoT security where they offered future challenges and current solutions. Another survey of ML methods for wireless sensor networks (WSNs) appeared in [33].

The motivation of that study was to review ML methods in real-life WSN applications, i.e., clustering, localization, routing and unrealistic aspects of quality of service (QoS), and security. The framework of WSNs in DL methods was described in the work of Oussous et al. [34]. On the other hand, this work emphasizes network configuration. Besides, it differs from the proposed survey that focuses on DL/ML methods for ensuring IoT security. Some traditional ML techniques [16] were considered with advanced methods, including DL methods [35] for processing big data. Above all, the relationship of several ML techniques for signal processing approaches was focused on investigating and processing relevant big data. An overview of DL is offered on state-of-the-art approaches [36]. The survey proposed the opportunities and challenges of various existing solutions with their uses and evolution. The essential principles of several DL classifiers were evaluated with their procedures in addition to developments of DL methods in several uses [37, 38], for example, speech processing, pattern recognition, and computer vision. In mobile advertising, a review of improvement in DL methods was used for recommendation systems, which show a crucial role [39]. Various effective ML applications [40] were similarly conducted in self-organizing networks. The survey focused on the merits and demerits of various methods and offered future opportunities and challenges in expanding artificial intelligence and future network design [41]. The significance of 5G in artificial intelligence was highlighted. Intrusion detection using data mining was covered in [42]. Application of multimedia mobile was surveyed conducting DL methods as well [43]. Recent DL techniques in mobile security, speech recognition, mobile healthcare, language translation, and ambient intelligence were focused. Similar research was conducted on the most advanced state-of-the-art deep learning approaches used in a variety of IoT data analytics applications [44]. On the other hand, our survey covers a complete review of recent progress in deep learning approaches and cutting-edge machine learning approaches for ensuring security in IoT. This review compares and identifies the advantages, prospects, and weaknesses of different DL/ML approaches for security in IoT. This paper also discusses numerous future directions and challenges and discusses the realized problems and future prospects based on a study of possible DL/ML applications in the context of IoT security, thus offering an effective guideline for researchers or scholars to modify the security of IoT environment from simply empowering a secure communication between IoT modules to providing IoT security on the basis of intelligence methods.

3. IoT Threat

Several heterogeneous sensing systems communicating with one another through a local area network (LAN) are referred to as the IoT [45]. The risks in IoT are distinct from those posed by traditional networks, owing in large part to the capabilities accessible to end devices [46]. The traditional Internet relies on powerful computers and servers with plenty of resources, while the IoT relies on equipment with low memory and computing power. That is why an IoT device in the real world cannot continue employing multifactor authentication and dynamic protocols like a regular network. Wireless protocols applied by IoT devices, for example, Zigbee and LoRa are less secure than those used by traditional networks. A lack of standard operating system and particular features inside IoT applications has resulted in various data contents and formats in the systems, making the creation of a uniform security protocol complicated [47]. There are several security and privacy problems associated with the IoT because of these flaws. As a network grows in size, the risk of an attack rises. Since the IoT has no firewalls, its network is more vulnerable than a traditional office or company network. IoT systems that exchange data with one another are frequently multivendor systems, adhering to a wide range of spectra and protocols from different manufacturers. The connection between such devices is difficult, necessitating the use of a trustworthy third party as a bridge [48]. Additionally, many reports have posed concerns about how billions of smart devices receive app updates [49, 50]. Since an IoT device has small computing resources, its ability to cope with advanced threats is harmed. To conclude, IoT weaknesses may be classified as either essential or widespread. For example, although vulnerabilities such as battery drain attacks, insufficient standardization, and insufficient confidence are exclusive to IoT systems, vulnerabilities in internet-inherited systems may be considered general. Numerous IoT risks have been identified and classified in the past [32, 51–54]. We address the most often identified challenges to the IoT over the last ten years and try to categorize them into privacy and protection classifications. Privacy and security are basic principles that turn around network availability [55–57]. On the Internet of Things, data may take several forms, including a user's identification records, an order issued to a car by a key fob, or a graphical chat between two people. Unauthorized data disclosure can constitute a breach of data security, integrity, or availability. If a threat compromises secrecy, it is classified as a privacy

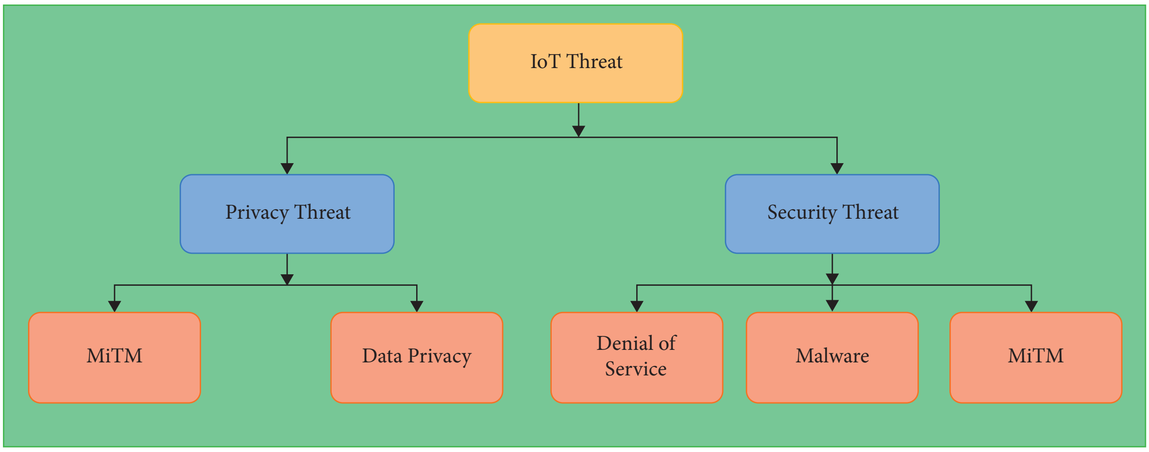


Figure 1: Types of IoT threat.

threat. Both data confidentiality and network stability are jeopardized by security attacks. Figure 1 depicts the various types of threats that exist in IoT domains.

3.1. Privacy Threats. Along with protection risks, there are common privacy threats against IoT data and their users, i.e., de-anonymization, inference, and sniffing. In any scenario, the effect is on data secrecy, regardless of whether the data are at rest or in motion. This segment discusses different types of privacy threats.

3.1.1. Man-in-the-Middle (MiTM). Passive MiTM attacks (PMAs) and active MiTM attacks (AMAs) are two types of MiTM attacks. The PMA passively monitors data flow between two systems. The data are not altered if the PMA violates anonymity. An intruder with access to a computer will observe passively for months before launching an attack. With the proliferation of cameras in IoT devices such as dolls, wristwatches, and tablets, the impact of passive MiTM attacks such as sniffing and eavesdropping is immense. In comparison, the AMA is actively engaged in data misuse, dealing with an operator pretending to be someone else, e.g., impersonation to retrieve a profile or an authorization attack.

3.1.2. Privacy in Data. As with MiTM attacks, attacks in data privacy are categorized as passive data privacy attacks (PDPAs) or active data privacy attacks (ADPAs). Data tampering, data outflow [58], re-identification, and identity stealing are all issues relating to data protection [59]. Re-identification attacks often referred to as hypothesis attacks are focused on position recognition, de-anonymization, and data aggregation [59]. The primary objective of these attacks is to collect data from various outlets to discover the targets' characters. Certain attackers can conduct the data gathered to mimic a specific goal [53]. Each attack that modifies records, such as data tampering, falls under the definition of ADPA, while data leakage and re-identification fall under the definition of PDPA.

3.2. Security Threats

3.2.1. Denial of Service. While compared to other types of security threats, denial of service (DoS) has the simplest application. In addition, as several IoT devices with poor security features continue to increase, DoS attacks are an attacker's favorite tool. The primary goal of a DoS attack is to overwhelm the IoT network with illegal requirements and to deplete network resources, including bandwidth. As a result, legal consumers cannot access the services. DDoS is a more complex form of DoS attack where a particular objective is attacked from several origins, which makes the attack more difficult to detect and avoid [60–65]. Though DDoS attacks have different flavors, they all have a similar objective. A variety of attacks in DDoS include SYN floods [66], in which a hacker dispatches a number of SYN appeals to a remarkable target; attacks in internet control message protocol (ICMP) [67] (in which several ICMP packets are being transmitted via a spoof-IP); crossfire attacks [68] in which an attacker is attacking a complex, massive botnet; and user data logs (User Datagram Protocol). Botnet attacks [69] are a form of DDoS attack occurring in IoT networks. A botnet is a group of IoT devices hacked to start an attack on a particular item, such as a bank server. Botnet attacks can be carried out by various protocols, including Message Queuing Telemetry Transportation (MQTT), a Domain Name Server (DNS), and a Hypertext Transfer Protocol (HTP), as outlined in [69]. Some ways to detect DoS attacks in an IoT environment are proposed. The authors in [61] showed how an attack in a fog-to-things environment is identified by applying DL techniques. In a second paper, the work of Abeshu and Chilamkurti [60] suggested that the usage of distributed DL on fog computing could ease DDoS attacks. The IDS is a sequence of development exercises to lessen attacks in DDoS with sophisticated computer learning and deep learning algorithms [63, 64]. The software-defined network flood issue has been emphasized by the authors in [62, 65]. The study showed that the top layer of the SDN is susceptible to a brute force attack because of the insufficient protection in the TCP channel of plain text.

3.2.2. Malware. One of the most well-known attack domains is the execution and injection of malicious code into IoT systems via developing existing vulnerabilities in IoT systems. Vulnerabilities in application security, authentication, and authorization may be exploited for malware injection. Without these approaches, physically tampering with IoT devices to modify the software and misconfiguring security parameters may also enable attackers to introduce malicious code. Malware is a persistent threat that is executed via various methods due to the vulnerabilities mentioned above. Malware comes in a variety of forms, including spyware, bot, adware, ransomware, virus, and trojan, to mention a few [47, 70]. Moreover, Azmoodeh et al. [71] conducted research on malware distributed through the Internet of Battlefield Things (IoBT). These hackers are often well trained, well funded, and state sponsored. The authors in [72–74] used various supervised machine learning algorithms to attempt to protect resource-constrained Android devices against malware attacks. The studies [50, 75, 76] examined malware detection in-depth and identified many security flaws in the Android framework, specifically at the application layer. It contains applications with a variety of component forms.

3.2.3. Man-in-the-Middle. Man-in-the-middle (MiTM) attacks were among the first cyber threat types [77]. Spoofing and impersonation include MiTM attacks. For example, the MiTM attacker could communicate with a node "*X*," which communicates with destination "*A*." Similarly, a hacker can use this kind of attack to link to a server with an HTTPS connection in SSL stripping. Recently, several researchers have been motivated to develop security against MiTM attacks [78–81]. A work of Ahmad et al. [78] illustrated the clinical condition in which a patient is given an insulin injection instantly. This form of program is subject to a fatal MiTM attack. Likewise, the authors in [80] addressed new safety methods for wireless mobile devices, including the usage of a concealed key, in the face of impersonation attacks. Using non-volatile memory and cryptography using hash values, this key has been safeguarded against loss or theft. This method was not only insecure but also wasteful of resources, like OAuth 2.0, which is the most extensively used IoT protocol and is vulnerable to cross-site request forgery attacks (CSRF). The OAuth protocol takes a considerable time to authenticate computers physically. The researchers in [81] listed a physical layer security defect in the authentication of a wireless system. They discussed the current hypothesis test, which compares some information in radio channels to the channel record of Alice to identify Eve spotter in wireless networks. It is sometimes inaccessible, mainly in active networks.

3.3. Another Threat to Privacy and Security. There are two types of security threats: physical and cyber. Active and passive cyber risks are further subclassified. The following part provides an outline of some of these risks and threats.

3.3.1. Physical Threats. Physical destruction is one kind of threat that may be posed. A cyberattack is not usually possible in these cases since the attacker lacks the necessary technical know-how. To put it another way, the attacker can only impact the IoT devices that can be physically accessed by the hacker. If IoT systems are used widely, these sorts of attacks may become more widespread since cameras and sensors are projected to be widely available and accessible [29, 82]. Natural catastrophes, such as earthquakes or floods, or human-caused disasters, such as wars, may also create physical threats [83, 84].

3.3.2. Cyber Threats

Active Threats. As part of an active threat, the attacker is not only skilled at listening in on communication channels but also at changing IoT devices to change settings and regulate communication, refuse services, and many more. A series of interventions, interruptions, and alterations may be used in an attack. There are several ways to attack an IoT system, including impersonation (such as spoofing), data tampering, malicious inputs, and DoS. IoT devices or authorized users may be impersonated in a cyberattack called an impersonation attack. If an attack vector is available, active intruders may try to mimic an IoT entity in part or its whole. An IoT system is attacked using a malicious input attack in order to introduce malicious software into the system. Code injection attacks will be carried out using this program. Malicious software injected into IoT systems has a dynamic character, and new attack types are continually being produced to breach the systems' IoT components in remarkable ways [30, 85]. Data tampering, on the other hand, is the act of purposely altering (deleting, altering, modifying, or manipulating) data via illegal actions. Transmitting and storing data are commonplace. IoT systems may be compromised in both cases, which might have substantial consequences, i.e., altering the IoT-based billing price of the smart grid. IoT may be subjected to a wide range of denial of service attacks. A wide variety of DoS attacks may be found, from those that target internet traffic to those that target cellular connectivity. It is more difficult to distinguish a DDoS attack from regular traffic and devices than a DoS attack with a small number of devices or a large signal, which is more straightforward to distinguish from regular devices and traffic than DDoS attacks. A typical goal of DoS attacks is to disrupt the availability of IoT services [27]. Many IoT systems are vulnerable to devastating DDoS attacks, such as Mirai, since they include billions of linked devices. Using IoT devices, the Mirai botnet has lately been utilized to launch large-scale DDoS attacks.

Passive Threats. Eavesdropping on the communication network or the channels is all that is required to carry out a passive threat. An eavesdropper may obtain sensor data, monitor the sensor bearers, or do both by listening in on their conversations. The illegal market for important personal information, such as health data, has exploded recently [86]. In comparison to credit card information, which sells for $1.50, and social security numbers, which sell for $3, personal health information is worth $50 on the black market. Furthermore, an attacker may determine the location of an IoT device's owner by eavesdropping on the owner's communications if they are in range [87, 88].

4. Applications of IoT Security

Almost all IoT applications, whether currently in use or under development, have security as a top priority. IoT applications are growing at a fast pace and have already penetrated most of the current sectors. A few IoT applications required stricter support of security from IoT-based technologies they employ, despite operators supporting these apps with current networking technology. There are several crucial IoT applications that are covered in this section.

4.1. Home Automation. IoT has a broad range of applications, including home automation. This category includes applications for remotely controlling electrical appliances to save energy, devices put on doors and windows to find intruders, and more. Energy use and water use are being tracked via the use of monitoring devices, and customers are being counseled on ways to conserve resources and money. The authors in [89] have recommended the application of security techniques that is based on logic to improve home security. Users' activities in crucial areas of the house are being compared to their typical behavior in order to identify intrusions. Attackers may, however, get access to IoT devices in the house without the owner's permission and use that access to do damage to the owner. For example, the number of burglaries has skyrocketed after different home automation systems were installed [89]. Internet traffic to and from a smart home has been used by opponents before to determine a person's activities or even their presence at the residence.

4.2. Smart Cities. Smart cities make full use of newly available computing and communication technologies in order to develop the lives of its residents [90]. Smart cities, smart transportation, smart disaster response, and other smart services are all included. Governments throughout the globe are promoting the creation of smart cities via different incentives [91]. Even though smart apps are meant to enhance the quality of life for individuals, they pose a risk to their privacy. Citizens' credit card purchasing habits and information might be at danger while using smart card services. Smart mobility apps can expose where its users are located. Parents can keep an eye on their children with the use of mobile apps. The child's security may be jeopardized, though, if these applications were hacked.

4.3. Smart Retail. Applications of IoT are widely employed in retail. Many applications have been developed to track the temperature and humidity of inventory as it moves through the supply chain. Additionally, IoT may be utilized to optimize warehouse refilling by monitoring goods movement. Intelligent shopping apps are also being generated to help consumers based on preferences, habits, and sensitivities to certain components, for example. These applications are being developed. An augmented reality system that allows physical shops to experience internet buying has also been created. IoT applications deployed and used by retail firms have been plagued by security concerns. They include Home Depot, Apple, Sony, and JPMorgan Chase. Attackers may attempt to breach IoT apps related to the storage conditions of goods and communicate incorrect information about the items to customers in an effort to promote sales. Consumers' credit and debit card information, e-mail addresses, phone numbers, and other personal information can be stolen if the elements of security are not comprised of smart retail. This can result in monetary losses for both the businesses and the customers.

4.4. Animal Farming and Smart Agriculture. Soil moisture monitoring, maintaining selective watering in dry zones and micro-climate conditions, and controlling temperature and humidity are only a few of the smart farming practices. The use of sophisticated features in agriculture may lead to higher yields and assist farmers avoid monetary losses. Moreover, fungus and other microbiological pollutants may be prevented by carefully monitoring and controlling the humidity and temperature levels in different vegetable and grain production processes. The quality and quantity of vegetables and crops may also be improved by controlling the climate. As with crop monitoring, apps in IoT are available to track the activity and health status of farm animals using sensors attached to the animals. Compromised agricultural apps might lead to animal theft and crop harm.

4.5. Smart Grids and Smart Metering. Management, monitoring, and measurements may all be done with smart meters. It is the most prevalent use of smart meters in smart grids, where power use is tracked and measured. A smart metering system might potentially be used to combat power theft. Storage tank monitoring and cistern level monitoring are two further uses for smart meters. By dynamically adjusting the position of solar panels in the sky, smart meters may also be used to monitor and enhance the performance of solar energy facilities. Other applications of IoT include smart meters to monitor water pressure to measure the weight of items or in water transportation systems. Smart meters, nevertheless, are susceptible to both cyberattacks and physical attacks compared to traditional meters, which can only be interfered with by physical means. Smart meters or advanced metering infrastructure (AMI) are designed to carry out additional tasks beyond the simple tracking of energy use. A smart home area network (HAN) connects all of a household's electrical appliances to smart meters, which may be used to monitor use and costs. Consumer or adversary intrusions into such systems may alter the acquired data, resulting in financial losses for users or service providers [92].

4.6. Smart Environment. IoT may be applied to identify forest fires, monitor snow levels at high altitudes, prevent landslides, detect earthquakes early, monitor pollution, and many other things. There is a strong connection between the lives of humans and animals in these regions and the use of IoT applications. The information from these applications of IoT will also be used by government entities working in these domains. The repercussions of a security breach or vulnerability in any IoT application area might be dire. False negatives and false positives may have severe effects for IoT applications in this situation. For example, if the app begins incorrectly identifying earthquakes, the government and companies may suffer financial damages. If, on the other hand, the software fails to forecast the earthquake, both property and lives will be lost. As a result, security flaws and data manipulation must be avoided in smart environment applications.

4.7. Security and Emergencies. The deployment of numerous IoT applications in the field of security and emergencies is another key development. It covers applications such as restricting access to restricted areas to only those with proper credentials. Hazardous gas leak detection in industrial regions and near chemical companies is another use for this technology. There are a variety of buildings where sensitive information is stored on computers or where sensitive commodities are stored. Protecting sensitive information and items is possible with the use of security apps. Buildings with high levels of sensitivity, such as nuclear power plants, may benefit from the usage of IoT apps that monitor liquids. The repercussions of a security compromise in these apps might be dire. Criminals may, for example, attempt to get access to restricted regions by exploiting programs' security flaws. The immediate and long-term consequences of erroneous radiation level warnings may also be severe. Long-term radiation exposure in babies, for example, may cause significant disorders that are life threatening.

5. IoT Applications of Security Threat for Each Layer

This section describes the four levels of an IoT application: the first is the sensing layer, the second is the network layer, the third is the middleware layer, and the fourth is the application layer. In an IoT application, each layer employs a variety of technologies that introduce a variety of challenges and security risks with them.

Security risks in IoT applications are discussed in this section for the four tiers. This section also discusses the particular security concerns of the gateways that link these levels.

5.1. Sensing Layer and Its Security Issues. This layer focuses on physical actuators and sensors for the IoT. Sensors pick up on the physical activity taking place around them [93]. While sensors collect information about their surroundings, actuators take action on the physical world depending on that information. There are a variety of sensors that may be used to gather information, including video sensors, ultrasonic sensors, humidity and temperature sensors, and more. Chemical, electrical, electronic, and mechanical sensors may all be employed to gather data about the physical world around us. IoT applications employ a variety of sensing layer technologies, including GPS, RFID, RSNs, WSNs, and so on. Sensing layer security risks include the following:

(i) *Injection Attack Using Malicious Code.* A malicious code is injected into the node's memory by the attacker. The firmware or software of IoT nodes is often changed over the air, which provides an entry point for malicious malware injection by attackers. Using malicious code, attackers may cause the nodes to execute certain attempts to get access or even undesired operations to the whole IoT system.

(ii) *Node Capturing.* Actuators and sensors are only two examples of low-power nodes in IoT applications. The opponents may launch a wide range of attacks against these nodes. A malicious node might be used to replace or capture the legitimate node in the IoT system. In reality, the attacker has complete control over the new node. This might lead to the full IoT application being compromised [94].

(iii) *Side-Channel Attacks (SCAs).* There are a variety of side-channel attacks that may expose sensitive data besides those that target nodes directly. An adversary may get access to sensitive information via the microarchitectures of CPUs, electromagnetic emission, and their power consumption. Power consumption, laser-based, timing, and electromagnetic side-channel attacks may all be used to launch attacks. To avoid side-channel attacks when implementing cryptography modules in modern circuits, there are many countermeasures.

(iv) *Booting Attacks.* Devices on the edge are susceptible to a variety of threats during starting up. A lack of built-in safety measures explains why this happens. When the node devices are rebooted, attackers may use this vulnerability to attack them. Due to their low power consumption and sleep-wake cycles, edge devices need to have a secure startup mechanism.

(v) *Sleep Deprivation Attacks.* Attackers' goal in these sorts of attacks is to drain the power supply of relatively powerless IoT edge devices. Due to a dead battery, the IoT nodes are unable to provide any service. For this to happen, malicious malware or increased power consumption on the edge devices is used to execute endless loops in those devices' CPUs and RAMs.

(vi) *False Data Injection Attack*. An attacker may then utilize the node to feed false data into the IoT system after it has been taken over. Inaccurate findings might lead to the IoT application malfunctioning. DDoS attacks may be launched using this strategy as well.

(vii) *Eavesdropping and Interference*. Open environments are widely used to launch IoT apps. This means that IoT applications are vulnerable to eavesdroppers as a consequence. It is possible for the attackers to eavesdrop and steal data throughout various stages of authentication or transmission.

5.2. Network Layer and Its Security Issues. The primary role of the network layer is to transfer data from the sensor layer to the computing unit for processing. The following are the most common network security problems.

(i) *DoS/DDoS Attack*. In this kind of attack, unwanted requests are issued in mass quantities to the attacked servers. As a result, legitimate users will be unable to access the targeted server's resources. DDoS attack occurs when several sources are exploited by the attacker to overwhelm the target server and cause it to become unusable. As a result of the complexity and variety of IoT networks, such attacks are more likely to take place because of this. Due to improper configuration, many IoT devices used in IoT applications are vulnerable to DDoS attacks. This vulnerability was used by the Mirai botnet attack to block multiple services by sending requests to IoT devices that were poorly configured [12].

(ii) *Routing Attacks*. Nodes in an application of IoT that are attempting to conduct malicious activity may attempt to reroute traffic as it passes through the system. In a sinkhole attack, an adversary broadcasts a fake shortest route and actively encourages nodes to use it. Wormhole attacks, when coupled with sinkhole attacks, may pose a major danger to computer security. Fast packet transmission may be achieved using a wormhole connection between two nodes. An attacker may exploit a vulnerability in an IoT application by creating a "wormhole" between a hacked node and a device on the internet.

(iii) *Access Attack*. The term "advanced persistent threat (APT)" may also apply to an access attack. In this form of attack, an unauthorized individual or an adversary obtains access to the IoT network without permission. For a long time, the attacker may remain unnoticed in the network. This kind of attack is aimed at stealing important information or data, rather than causing harm to the network. Applications of IoT are particularly vulnerable to attacks because they constantly collect and send crucial data.

(iv) *Data Transit Attacks*. The storage and sharing of data is a major concern for IoT applications. Because data are so valuable, they are constantly a target for cybercriminals and other bad guys. Whether data are kept locally or in the cloud, they are subject to cyber attacks while they are in transit or are traveling between locations. Data travel a long way in IoT applications between actuators, sensors, and the cloud, among other places. Data transmissions using the IoT may be compromised since a variety of connecting mechanisms are in use.

(v) *Phishing Site Attack*. Phishing attacks are those in which a single attacker may target a large number of IoT devices with little to no effort. The attackers believe that at least some of the devices will succumb to the onslaught. There is a chance that individuals may encounter phishing sites when browsing the internet. Any IoT device that a user has access to becomes a target for cyberattacks as soon as their login credentials are stolen. Phishing attempts on the network layer of IoT are quite common [95].

5.3. Middleware Layer and Its Security Issues. The middleware in IoT is responsible for creating an abstraction layer between the application and network layers. It is also possible for middleware to offer substantial compute and storage capabilities [96]. The APIs provided by this layer are used to meet the needs of the application layer. The middleware layer contains machine learning, permanent data storage, brokers, queuing systems, and so forth. Middleware is important for providing a strong and dependable application of IoT, but it is also vulnerable to a variety of attacks. As a result of these attacks, the whole IoT application may be hijacked. Besides database and cloud security, middleware security is a major concern. These attacks on the middleware layer are described in more detail below.

(i) *SQL Injection Attack*. Middleware may be attacked using SQL injection (SQLi), which is another attack vector. A malicious SQL query may be inserted into a program by an attacker in such attacks [97, 98]. This allows the attackers to access the private information of any user as well as to change database entries themselves [99]. According to the Open Online Application Security Project (OWASP) top 10 2018 paper, SQLi is a top threat to web security.

(ii) *Flooding Attack in Cloud*. Cloud-based denial of service attacks use a similar methodology and impact QoS in the same way. The attackers use a continual stream of queries to a service to deplete cloud resources. By increasing the strain on the cloud servers, these attacks may have an important effect on cloud systems.

(iii) *Man-in-the-Middle Attack*. Subscribers and clients communicate with each other through the MQTT broker, which operates as a proxy for the MQTT protocol. Messages may be transmitted to several recipients without knowing where they are going thanks to this method's ability to disconnect the publishing server from the subscribers. As long as the attacker can take control of the broker, he or she will be able to take over all communication without the awareness of the clients.

(iv) *Signature Wrapping Attack*. XML signatures are utilized in the middleware's web services. An attacker may use weaknesses in Simple Object Access Protocol (SOAP) to break the signature scheme and perform operations or change intercepted messages in a signature wrapping attack.

5.4. Gateways and Their Security. Connecting devices, objects, people, and cloud services is a key function of a gateway. In addition, gateways aid in the provision of IoT device hardware and software. The decryption and encryption of IoT data, as well as the translation of protocols across various levels, are handled by gateways. Zigbee, LoRaWAN, TCP/IP, and Z-Wave stacks are among the several IoT systems that are currently in use today. In the following, we will look at some of the security issues that IoT gateways are facing.

(i) *End-to-End Encryption*. End-to-end application layer protection must be executed in order to guarantee data confidentiality [37]. Only the intended receiver may decode encrypted communications using this program. Z-Wave and Zigbee protocols offer encryption; however, the gateways are necessary to decrypt and re-encrypt the messages in order to convert the information from one protocol to another. At the gateway level, this decryption exposes the data to security vulnerabilities.

(ii) *Firmware Updates*. In order to obtain and install firmware upgrades, most IoT devices lack a user interface or the computational capacity. Typically, gateways are used to obtain and apply firmware upgrades. Verify signature validity and the existing and new firmware versions before implementing any changes.

(iii) *Extra Interfaces*. Installing IoT devices while keeping an eye on the attack surface is a critical technique [100]. IoT gateway manufacturers should only implement the protocols and interfaces that are strictly essential. A backdoor authentication or data leak should be prevented by restricting certain services and functionality for end users only.

(iv) *Secure On-Boarding*. Whenever a new IoT sensor or device is added, encryption keys must be protected. All keys flow via gateways, which operate as a go-between for management services and new devices. During the on-boarding process, the gateways are vulnerable to eavesdropping attempts and MiMA aims at stealing the encryption keys.

5.5. Application Layer and Its Security. The application layer is responsible for interacting with and serving customers directly. Smart cities, smart homes, smart grids, and other IoT applications all fall under this umbrella. This layer contains unique security concerns, such as data theft and privacy concerns, that are not present in other levels. Various apps have different security concerns at this tier. A sublayer between the network layer and application layer, known as a middleware layer or application support layer, is used in many IoT systems. There is a layer of support that enables different business services and aids in the allocation and calculation of resources. The application layer's most pressing security concerns are outlined here.

(i) *Reprogram Attacks*. If the process of programming the IoT devices is not secure, the devices might be reprogrammed remotely. This might eventually lead to complete control of the IoT [101].

(ii) *Malicious Code Injection Attacks*. To get access to a network or system, attackers often use the quickest or most straightforward technique. If the system is susceptible to misdirection and malicious scripts owing to poor code checks, then an attacker would select it as the first access point. XSS (cross-site scripting) is often used by attackers to introduce malicious code into a supposedly trustworthy website that is otherwise safe. For example, if an IoT account is hacked, it might cause the whole system to be rendered inoperable.

(iii) *Access Control Attacks*. The term "access control" is used to describe the practice of restricting access to a resource (such as an account or data) to just those who are allowed to use it. When a user's credentials are stolen, the whole IoT program is vulnerable.

(iv) *Service Interruption Attacks*. In the current literature, these attacks are also known as DDoS attacks or unlawful interruption attacks. IoT applications have been the target of a number of similar attacks in the past. By intentionally overloading the network or servers, these attacks prevent genuine users from accessing IoT apps.

(v) *Data Thefts*. Critical and confidential data are handled by IoT apps. In applications of IoT, there is a great deal of data mobility that makes it even more susceptible to attacks than data at rest. Unless the IoT apps are secure, consumers will be unwilling to provide their personal information. Data encryption, data isolation, privacy network, and management and user authentication and other methods and protocols are being employed to protect the applications of IoT against data theft.

6. Overview of DL and ML for IoT

Security and privacy are interdependent. It is possible to conceive of a setting that is safe yet does not provide individual confidentiality. One may envisage a dwelling that is private due to the presence of windows, yet this would not always provide protection from intruders. While privacy is impossible to get without sacrificing some level of security, the opposite is not true. Privacy is always compromised when security is inadequate or exposed. In this section, we will illustrate the most prominent ML and DL models for classifying security aspects in IoT. ML techniques refer to unsupervised and supervised methods. The supervised methods are classified into Naïve Bayes (NB), support vector machine (SVM), random forest (RF), K-nearest neighbor (KNN), decision tree (DT), ensemble learning (EL), and association rule (AR). Additionally, the unsupervised approaches only refer to two approaches including principal component analysis (PCA) and K-means. DL techniques are similarly classified into unsupervised, supervised, and hybrid methods.

6.1. ML in IoT Security.

6.1. ML in IoT Security. This section discusses the traditional ML algorithms. Figure 2 depicts various ML classifiers for IoT security.

6.1.1. Supervised Machine Learning. We consider some traditional supervised ML techniques and merits, demerits, and applications in IoT for security enhancement.

(1) Bayesian Theorem-Based Systems. Bayes' theorem illustrates the possibility of an event based on previous data associated with the event [102]. This is exemplified by the fact that DoS attacks are linked to network traffic information. Accordingly, Bayes' theorem is likely to evaluate the attack on network traffic by applying previous traffic facts aside from the above. As a typical ML process, Naïve Bayes is a Bayes' theorem. As a result of its ease of use, it is sometimes called a supervised classifier.

NB estimates subsequent possibilities and applies Bayes' theorem. For specific feature sets, this theorem can forecast the likelihood. For example, Naïve Bayes can be applied to categorize the traffic, including abnormal or normal. These features may be employed for the classification of traffic, for example, connection position flag, connection protocol (e.g., User Datagram Protocol (UDP) and Transmission Control Protocol (TCP)), and connection duration are implemented or computed independently by this classifier. Despite that, features depend on each other. Each feature predicts the probability in this classification method where the traffic is abnormal or normal. So, the Naïve Bayes was modified by hyperparameter tuning. This optimized model was applied for the detection of anomalies [103, 104] and network intrusion [105, 106]. The main benefits of these classifiers are ease of use, simplicity of implementation, the requirement of a small training sample, applicability to multiclass and binary classification [107], and toughness to inappropriate features [108].

The paper of Zhang et al. [109] offered a method of intrusion detection to develop Naïve Bayes and PCA. It depicted that the Bayes classifier was more efficient than other classifier algorithms for detecting intrusion due to its rapid speed in the classification process. The intrusion of PCA can highly decrease the detection period. Next, the weight coefficient has been described to develop the PCA so that it can reduce the complexity of input data. The comparison between the detection time and rate using the traditional Bayesian method depicts that the technique introduced in this task is the best in intrusion detection. This work provided good accuracy. Besides, it also timely solved the requirement for detecting the intrusion of the network [109].

(2) Support Vector Machines (SVMs). SVMs are mainly applied to evaluate data so that they are employed for classification and regression analysis. In the attributes of data, SVMs generate a separating hyperplane between the classes (two or more). The main target of the hyperplane minimizes the maximum adjacent sample features and distance between the hyperplane of each class. Each class has a maximum margin with minimum error [31, 110, 111]. Non-linearity will occur if the research inputs cause the hyperplane to become confusable, necessitating the use of a kernel function to reform it. It is also challenging to use the appropriate kernel function in SVMs. With its high degree of precision, SVM is ideal for implementing network protection for IoT devices such as smart grid [112], ransomware [113], and intrusion [114].

SVMs have been utilized for classifying data by constructing a scattering hyperplane between two or more groups of data attributes that maximize the gap between the hyperplane and the class nearest sampling points [115, 116]. SVMs are well known for their broad spectrum of practical properties, but they are remarkably well suited to datasets with a limited number of sampling points [31, 110]. Theoretically, statistical learning [111] is designed for SVMs. Initially, they were designed to partition into a plane of two-dimensional composed of points of linearly independent data in various groups (i.e., abnormal or normal). This model will benefit from a good hyperplane to maximize distance by calculating the discrepancy between the closest points and the hyperplane in every class. It has benefitted from its ability and scalability to track intrusions in real time and automatically change training tendencies. It has been commonly applied in a variety of security applications, including intrusion detection [117–120], and is memory effective since they break data points using hyperplanes with O ($N2$) time complexity, where N is the sampling number [31, 110]. Research in [113] has created an Android malware identification tool to help protect IoT networks, as well as a linear SVM for its device in the context of the IoT. SVM's identification efficiency is superior to those of other computer algorithms such as Naïve Bayes, RF, and DT. SVM, on the other hand, outperformed the other ML algorithms. These findings support the robustness of SVM-based malware identification. However, more research is required to examine the efficiency of SVMs with enriched and attack scenarios of datasets generated in a variety of environments. In this case, it might be helpful to compare the efficiency of the SVM with that of deep learning algorithms like CNN. An SVM was previously used to protect an intelligent device, and an observational smart grid attack detector was tested [121]. This study found that ML algorithms including SVM,

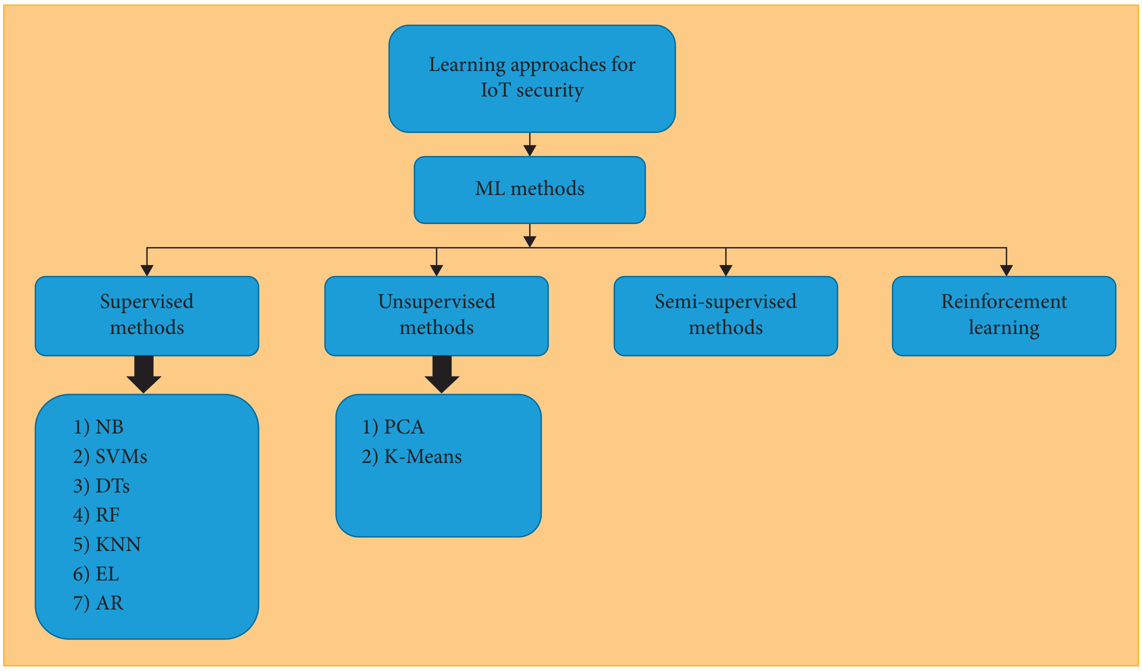


Figure 2: Various ML methods for IoT.

KNN, sparse logistic regression, and ensemble learning successfully identify unknown and known threats, outperforming traditional approaches used in intelligent grid applications. In another line of investigation, SVM was recently used to crack data encryption. The findings in [122, 123] demonstrate that ML techniques can be applied to hack cryptographic devices, and SVM outperforms conventional methods (such as template attacks).

(3) Decision Trees (DTs). The majority of DT-related classification methods are carried out by labeling samples based on their values of the attribute. Every vertex in a decision tree denotes one point, and every edge in the classification analysis represents a possible attribute for the vertex. Samples are categorized based on their attribute values and are classified from the root vertex [124, 125]. The feature that separates the training data optimally is referred to as the tree's initial vertex [126, 127]. Many approaches, such as the Gini index [128] and information benefit [129], are applied to deviate from training samples in search of the optimal function. The majority of DT approaches are split into two stages: classification (inference) and construction (induction) [130, 131]. DT is typically built by starting with an unoccupied tree and adding nodes and branches during the building (induction) phase. The feature that essentially divides the training samples is then called the tree's origin vertex. This role is chosen for a variety of reasons, including the value of experience. The idea is to delegate root nodes to reduce the intersection of groups in a training range, thus enhancing the discrimination efficiency of the classifier. Each sub-DT goes through the same process before all the leaves and associated groups are obtained. Following the development, new species are categorized with a collection of characteristics and an undefined class, beginning with the root nodes of the tree and progressing in the direction of the position values on the tree's inner nodes. This process is repeated before a leaf is collected. Lastly, the latest associated samples (such as expected classes) are determined [130]. Researchers summarized the crucial points for simplifying DT development in [130]. To begin, post or pretaking is used to reduce the tree's height. The state search space is then redimensioned. Third, the search algorithm has been updated. Following that, data attributes are minimized by disregarding or deleting unwanted features via the search procedure. To conclude, the tree's architecture is transformed into a data form, such as a law list. The table below summarizes the major drawbacks of DT-based approaches [130]. First, because of the house's design, they need a lot of room. Second, learning DT-based strategies is only simple when only a few DTs are involved. Certain structures, on the other hand, have a large number of trees and judgment nodes. The computational complexity of these applications, as well as the model underlying sample classification, is large. In defense applications such as intrusion prevention, DTs are used as the primary classifier or in combination with other master classifiers [132, 133]. In a previous study, for example, IoT devices were protected using a fog-based call system [134]. The thesis employed DT to examine network traffic in order to identify suspect traffic origins and, as a result, DDoS activity.

(4) Random Forest (RF). The abbreviation "RF" refers to a supervised learning algorithm. Several DTs are built and combined in an RF to provide an accurate and reliable prediction model, resulting in improved overall results [135–139]. An RF is then made up of several trees that have been built at random and conditioned to vote for a certain class. The classification's final performance is determined by

the most well-known class [135]. Since RF classification is mainly composed of DTs, these classification algorithms are very different. To begin, as the training set is fed into the network, the DTs generate a set of recommendations for classifying new data. RF constructs subsets of class voting rules that employ DTs; as a result, the designation contribution is the average vote, and RF is resistant to overfitting. Furthermore, RF eliminates the need to choose functions and permits a small range of input parameters [31]. However, in many real-time applications where the training dataset is high, RF may be inefficient since it requires the creation of several DTs. Radio frequency techniques have been used to track network anomalies and intrusion detection [137, 140, 141]. In [142], when narrow feature sets are applied to develop the device's application and reduce computational overhead to real-time classifications, ANN, KNN, SVM, and RF were learned to identify DDoS attacks on IoT systems, with RF being slightly stronger than other classificatory algorithms. RF was trained to recognize IoT interface groups from a white list of network traffic capabilities. The authors manually label and retrieve data from the network's seventeen IoT modules. These systems were divided into nine IoT device groups and applied to train a multiclass ranking using RF algorithms. According to the findings, ML algorithms are generally useful for correctly recognizing unauthorized IoT devices, especially RF [13, 54, 143]. Figure 3 depicts the basic architecture of RF.

(5) K-Nearest Neighbor (KNN). KNN is an ad hoc nonparametric approach. In KNN classifiers, the Euclidean distance is often used as a distance metric [144–146]. Figure 4 shows how the KNN classification is used to characterize new input materials. The orange circles in the diagram represent destructive behavior, while the blue circles represent machine actions. The most recent sample (blue circle) must be labeled as benign or malicious. The classifier of KNN categorizes the novel example by voting a fixed number of times, i.e., the class of unknown samples is determined by KNN using a plurality vote of the closest neighbors. For example, if the classification of KNN is created on the nearest neighbor (where $k = 1$), Figure 4 would classify the hidden sample as having normal behavior (as the closest cycle is orange.) Since the two nearest circles are orange, the unknown specimen would be identified as having natural activity if the classification of KNN is trained on the two closest neighbors (where $k = 2$ for normal behavior). If the classification of KNN has been trained on the four and three neighboring countries ($k = 4$, $k > 3$), the unknown sample class would be labeled as aggressive since the three and four circles closest to the unknown sample class are orange circles (malicious behavior). Cross-validation is essential for evaluating the optimum rate of k for a particular dataset. The KNN algorithm is an easy, high-performing classification algorithm on broad training datasets [147, 148], but the optimal k value is still determined by the context. Choosing the optimum value of k can also be a time-consuming and difficult operation. KNN classifiers have been used in network attack identification and detection of abnormalities [149–157]. In the area of the IoT, researchers in [158] recommended a paradigm for detecting R2L and U2R attacks. The algorithm decreased the dimensionality of the feature, allowing for two degrees of feature reduction to improve precision before introducing a model for a 2-tier classification based on KNN and NB classifiers. The suggested model performed admirably in detecting both attacks. Another study [159] developed a KNN method focused on an intrusion detection method. The invention was intended to be used for node classification in a wireless sensor network (WSN). The proposed program was accurate and precise in detecting intrusions.

(6) Ensemble Learning (EL). Ensemble learning (EL) is one of the most suitable methods of ML. EL integrates the outputs of various fundamental classification methods to achieve a single performance, increasing classification accuracy. This method attempts to integrate several multiclassifiers (such as heterogeneously or homogeneously) in order to arrive at a final output [160]. At the early stages of machine learning growth, each technique has advantages and accomplishments in particular implementations or datasets. Experiment comparisons in [161] revealed that the optimal style of learning varies depending on the application. The simple learning principle used to construct a classifier is determined by the data. Since the quality of data varies based on implementation, the best learning technique cannot be suitable for all applications. As a result, several classifiers have begun to be integrated to improve precision. EL employs a variety of learning techniques to eliminate inconsistency and is immune to overfitting. Combining several classifiers yields findings that go beyond the new range of theories; therefore, EL can respond well to a problem [162]. Due to the fact that EL is composed of several classifiers, an EL-based architecture has a higher time complexity than an EL-based scheme [163–166]. EL successfully detected intrusions, anomalies, and malware [167–170].

(7) Association Rule (AR) Algorithms. By analyzing the relationships between variables in a training data collection, AR algorithms [171] were used to characterize an unknown variable. Consider the variables X, Y, and Z in the P dataset. In order to analyze their links and to construct a model, an AR algorithm is intended to investigate the connections between these variables. The model then calculates the current sample class. AR algorithms describe regular sets of variables [31], which often coexist with vector collections in an attack. For instance, connections between IP/TCP variables and the attachment style using an AR were studied in a previous study [172]. The frequencies of various variables such as goal port, source IP, source port, and service name were analyzed in order to evaluate the attack form. The AR intrusion detection algorithm was successful, according to [173]. The researchers created a fuzzy rule-based model for intrusion detection, which resulted in a low false-positive rate and a high detection rate [173]. Nevertheless, compared to previous learning techniques, the increased reality is not widely utilized in IoT settings; more research is still recommended to determine if an approach to augmented

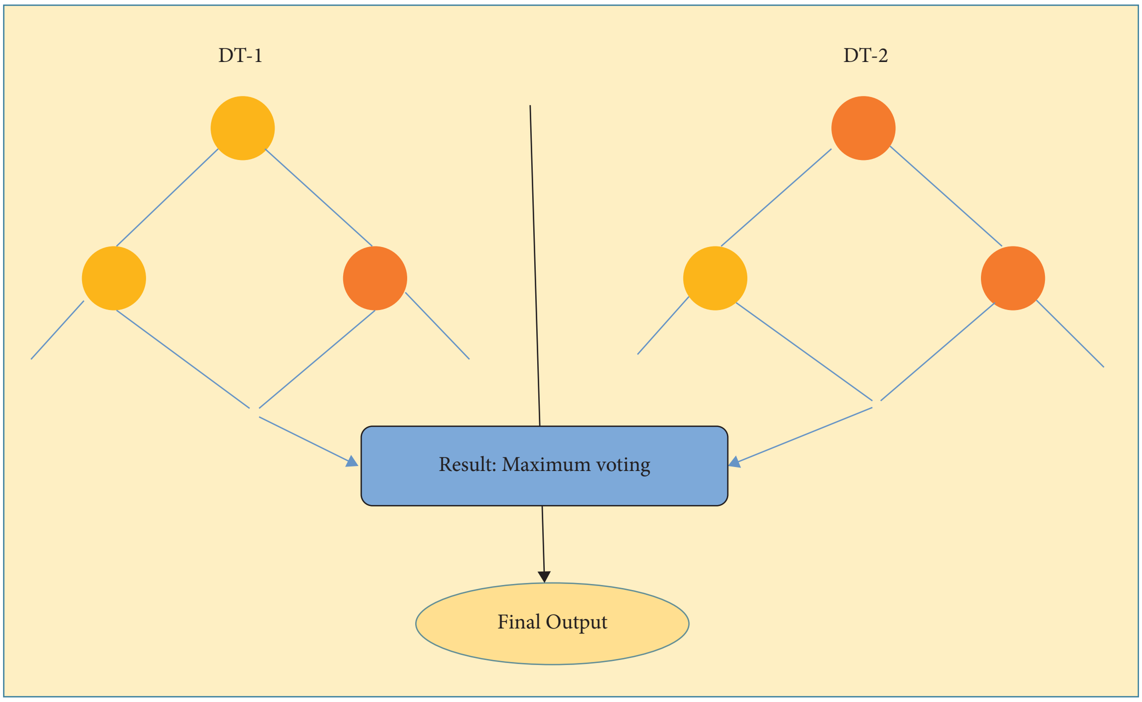


Figure 3: A basic architecture of RF.

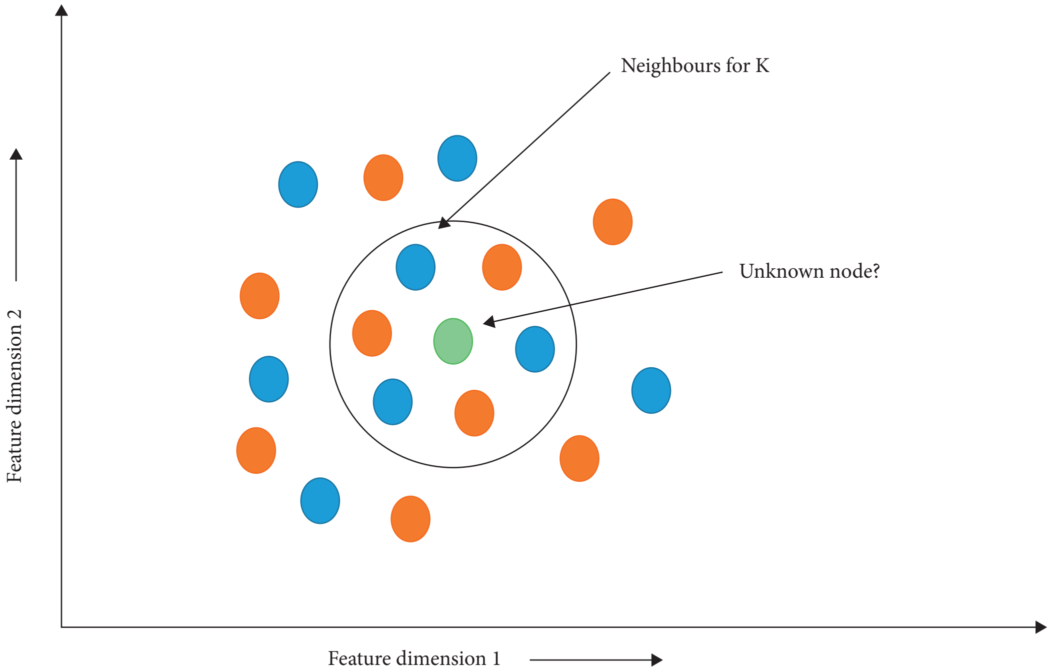


Figure 4: Principle of KNN.

reality may be combined or optimized with one more method to offer an appropriate solution for IoT protection. In practice, the following are the key disadvantages of the algorithms of AR. AR algorithms are difficult to deal with on a machine. If the frequency of the factors is reduced to an unmanageable level, association laws accumulate quickly. Although numerous productivity techniques have been developed, they are not often successful [174]. AR algorithms mostly focus on basic assumptions about variable relationships (direct relationships and occurrence). These assumptions are not always right, particularly when it comes to defending apps where attackers may try to mimic regular user behavior.

A study of Bosman et al. [175] demonstrated how to reduce the time complexity to make it suitable for systems with minimal resources in hardware, for example, IoT devices. This work suggested an ensemble-based method that is lightweight and application-independent for identifying abnormalities in the Internet of Things. The recommended framework addresses two concerns: (1) automating as well as disseminating approaches in online learning for detecting device irregularities that are limited by resources and (2) testing the proposed framework with actual evidence. According to the research, the ensemble approach produces each classification [175].

6.1.2. Unsupervised ML. In this section, we discuss the most frequently used unsupervised ML techniques (i.e., principal component analysis (PCA) and K-means clustering) and their applications, drawbacks, and advantages in IoT security.

Unsupervised learning occurs where the environment produces only inputs without regard for expected outcomes. It does not include branded data and can examine similarities between unlabeled data and classify them into distinct classes.

(1) Principal Component Analysis (PCA). The PCA is a strategic reduction function that can be used to restrict a wide range of variables to a smaller list that maintains the bulk of the data. This approach reduces a large range of potentially associated features to some more minor uncorrelated features known as principal components [176–178]. There is an increasing order of variation among these components; the first is related with the most variance in the data, and the rest follow in order. Discard the components with the smallest variance. PCA is a fantastic solution in real-time scenarios. In contrast, there are several characteristics, and it is difficult to see the connection and identify the correlation between each and every data. A large variety of accessible features also makes it quite difficult to narrow down the most important ones. Thus, the primary concept of PCA can be used to pick features for simultaneous detection of intrusion in IoT schemes. PCA helps speed up machine learning algorithms by removing linked variables that do not add to the algorithm's decision-making process. Moreover, to avoid the problem of overfitting, PCA reduces the number of dimensions in a dataset to a more manageable dimension, i.e., 2D [179, 180]. In this example, $n^2P + n^3 = O(n^2P + n^3)$ if each data point has P features and number of data is n.

The authors of the work [181] suggested a model that utilizes the reduction of features in PCA and classifiers such as KNN and softmax regression. According to the work of Zhao et al. [181], combining these classifiers with PCA produced a computationally and time-efficient technique. It can be used in IoT environments in real-time applications.

(2) K-Means Clustering. Unsupervised ML is exemplified by K-means clustering. This method seeks to find the data clusters, and k represents the several clusters that the algorithm will generate. The process is used to allocate each data point iteratively to a cluster of k according to the characteristics specified. Samples with identical characteristics can be included in each cluster. The K-means algorithm results in iterative refining. Two inputs are needed for the algorithm: several clusters (k) and a dataset with the specification of each example in the dataset. According to the estimate of the k centroids, each sample is allocated to the cluster centroid nearest to it, depending on the Euclidean distance. Secondly, after all data samples have been allocated to a given cluster, the cluster centroids would be reevaluated, applying the mean of all samples allocated to that cluster. The algorithm repeats these calculations until no samples can be used to modify the clusters [182, 183]. The preceding are the key disadvantages of clustering in K-means. To begin, the user must enter k. Second, this algorithm is based on the premise that each spherical cluster has nearly equivalent sample counts. The algorithms of K-means may be used to detect irregularities by contrasting normal and odd behavior features [184, 185]. The authors in [186] suggested an anomaly recognition scheme focused on DT and K-means (such as DT and C4.5 algorithm). Nevertheless, K-means performed poorly than directed learning approaches, especially when detecting established attacks [187]. When it is difficult to obtain labeled outcomes, unsupervised algorithms are usually a safe bet. Nonetheless, clustering methods in general, especially K-means, are still in their infancy for stable IoT structures and should be studied further. ML techniques that are not regulated have many applications in IoT network security. K-means clustering, for example, has been used to protect WSNs by detecting intrusions [188]. In research on Sybil recognition in industrial WSNs, a kernel-based scheme was suggested for clustering channel vectors to distinguish Sybil from standard sensors [189]. A clustering algorithm demonstrated the possibility of anonymizing private data in an IoT scheme [190]. In implementing anonymized data algorithms, the usage of clustering would significantly improve the protection of data sharing [190].

6.1.3. Semi-Supervised ML. Regulated machine learning is the most widely used ML process, and it derives its information from the training period on labeled data. To begin, developing predictive models from labeled data takes time, money, human interaction, and expertise. On the other hand, unsupervised learning that operates on unlabeled data frequently has an exploratory component (i.e., compression and clustering). Thus, the researchers hope that by implementing a semi-supervised approach, they would be able to solve the problem of producing vast quantities of labeled data required for training algorithms in supervised machine learning by augmenting unlabeled data [191, 192]. Accordingly, semi-supervised learning deploys both classified and unlabeled input to train a machine learning classifier. Nevertheless, while semi-supervised learning can seem to be a suitable answer to the problems associated with both managed and unsupervised approaches, it can fall short of the prediction precision attained by the algorithm in supervised machine learning. As a consequence, limited studies have looked into the use of semi-supervised methods for protection in IoT. For example, the authors in [193] described a semi-supervised multilayer clustering (SMLC) technique for detecting and preventing intrusion in the network. SMLC has demonstrated the ability to learn from incompletely labeled instances while also gaining recognition performance comparable to managed machine learning for detection and avoidance systems [194].

6.1.4. Reinforcement Learning (RL) Approaches. One of the first topics that come to mind is learning from one's own surroundings. People naturally begin their education by engaging with their surroundings. RL is driven by neuroscientific and psychological observations of animal behavior and mechanisms that enable agents to have a more significant effect on their environment [195–197]. RL shows people how to better map conditions to actions in order to maximize rewards [196]. The agent does not realize which actions to do ahead of time and must determine the acts that have the most significant benefit by assessment or error. The features' trial and error are the key characteristics of RL. As a result, the agent continues to gain expertise in order to maximize the benefits. RL has been used to address a variety of IoT-related problems. The work in [198, 199] suggested a broadband, autonomous cognitive radio anti-jamming system with an emphasis on learning enhancement. Data were used in [198] to differentiate between the swinging jammer signal, unintended interference, and former WACRs; reinforcement learning was then applied to reliably learn a selection technique of subband to stop the jammer signal as well as interfere with previous WACRs. Likewise, the authors in [199] discovered how to efficiently prevent jamming attacks from hundreds of MHz of spectrum in real life using an enhanced learning system based on Q-learning. Another work [200] used similar strengthening training to develop a cognitive radiation anti-jamming method, which was paired with deep CNN to increase the performance of RL across a wide range of frequency sources. A related method was suggested in [200] to tackle aggressive jamming using a rigorous learning approach; the findings showed that RL was an appropriate tool for modeling aggressive jamming schemes.

6.1.5. Applications of ML in IoT Security. In data analysis, semi-supervised and supervised methods are used, whereas comparative and decision-making properties are favored for reinforcement. The essence of accessible data dictates the categorization and methods used by ML. Supervised learning is used to determine the form of input data and the desired outcomes (labels). In this case, the machine was taught to only map the inputs to the necessary outputs. Regression and classification are supervised learning processes that utilize constant data regression and discrete data classification, respectively. Many regression methods, such as polynomial regression, linear regression, and SVR, are widely used [54, 201, 202]. In contrast, classification employs distinct production qualities (class labels). K-nearest neighbor, SVM, and logistical regression are provided by commonly used classification algorithms. Certain architectures, such as neural networks, may be applied for both regression and classification. Where the results are not well defined and the process must search within the raw data framework, unchecked learning approaches are applied to teach the algorithm. Clustering is a form of unattended learning in which items are clustered based on similar parameters, i.e., K-means clustering. The accuracy of predictive analytics is determined by how effectively master learning utilizes historical data to create models and how well future values are estimated. Algorithms such as Naïve Bayes and SVM are applied in predictive modeling. The one disadvantage of simple machine learning approaches is that they need a large amount of data for model testing. The learned model is then used to approximate or interpret real-world application performance. However, it should be remembered that the whole procedure would not capture the whole spectrum of data and resources. To address the shortcomings of machine learning methods, DL techniques have been deployed. DL can handle large amount of data, and its algorithms are adaptive as data volume increases, benefiting model training and possibly improving prediction precision. DL extracts high-level functionality and associated connections from input data in a complex and hierarchical manner. The majority of IoT implementations produce outcomes without labeling or with semi-labeling. DL may use unlabeled data to identify valuable trends in an unattended manner. Conventional machine learning algorithms are only effective when there is a large number of labeled data available [45].

The work of Rathore and Park [194] suggested a semi-supervised learning IoT attack prediction mechanism. The proposed system is based on the algorithm of the extreme learning machine (ELM), using fuzzy C-means (FCM) [203] approaches collectively known as extreme learning machine (ELM)-based semi-supervised fuzzy C-means (ESFCM). Similarly, ESFCM is used in fog infrastructure. One characteristic of ESFCM is that it operates with marked directories, improving the rate of detection of threats transmitted. Although the detection performance of ESFCM is lower than those of the two former DL mechanisms, it outperforms traditional attack detection machine learning algorithms. Nevertheless, the semi-supervised learning process incorporates the benefits and effectiveness of managed and uncontrolled learning. The IoT has a multitude of flavors, from body area networks to sophisticated key business facilities such as a smart grid. At the same time, identifying attacks on these infrastructures is important. In an intelligent grid, for example, steps are critical and must be recovered authentically and without alteration as a result of an attack. The authors in [121] conducted a comprehensive study in this direction to explore various algorithms in ML for attack recognition in smart grids. The researchers studied the function of space fusion, semi-monitored learning, online learning, and supervised learning algorithms. The authors considered the effectiveness of online approaches for detecting attacks in real time by concentrating on their numerical complexity that is usually below that of batch learning algorithms [121]. In contrast, all families of the above algorithms performed fairly well.

6.2. DL in IoT Security. Deep learning has developed a key topic for study in recent years in IoT systems [14, 204, 205]. The main advantage of deep learning over classical machine learning is its higher effectiveness for big datasets. Many IoT schemes produce huge quantities of data; therefore, for those systems, DL methods are well suited. Furthermore, DL dynamically generates dynamic data representations [206]. The IoT ecosystem can be connected in-depth with DL methods [207]. Deep connection is a unified protocol that facilitates automated communication between computers and applications linked to the Internet of Things. For example, in an intelligent home, IoT devices automatically talk to each other to form a completely intelligent home [14]. DL approaches use a computational paradigm that integrates many layers to learn different degrees of abstraction in data structures. Compared to traditional ML approaches, DL techniques have greatly enhanced state-of-the-art methods [208, 209]. DL is a subfield of ML using various non-linear layers of computation to abstract and turns discriminatory or generative pattern analysis functions. Since DL methods may catch hierarchical images in deep architecture, they often refer to themselves as hierarchical methods of learning. The operational theory of DL is motivated by the interpretation of impulses by the human neurons and brain. Deep networks are used to include unsupervised and supervised learning and a combination of these two learning forms, such as deep hybrid learning. This section discusses the most commonly used deep learning algorithms. Figure 5 depicts several DL classifiers for IoT security.

6.2.1. Supervised DL. This section discusses the most often used controlled DL methods. Recurrent neural networks (RNNs) and convolutional neural networks (CNNs) are two types of discriminative DL algorithms.

(1) Convolutional Neural Networks (CNNs). CNNs are created in order to decrease the number of data parameters used in typical neural artificial networks (ANNs). To minimize data parameters, three terms are used: sparse relationships, parameter sharing, and fair distribution [210, 211]. Reduced layer-to-layer relationships improve CNN scalability and complexity. A CNN comprises two types of layers: convolution and convergence. Convolutional layers combine data parameters by using a variety of analogous filters (kernels) [212]. The pooling layers' sample reduces the size of the following layers through average pooling or peak pooling. The top pooling algorithm splits the input by non-overlapping clusters, selects the highest value in the previous layer for each cluster [213, 214], and then combines the values of each cluster in the previous layer with the average pooling algorithm. The activation device is another critical layer of the CNN; each vector in the feature space has a function in non-linear activation. The ReLU is chosen since it contains nodes with the activation property $f(x) = \max(0, x)$ [215]. Figure 6 depicts how CNN works when IoT protection is extended. The biggest drawback of CNN is that it is widely used in deep learning strategies. It also enables high-performance automatic learning of raw data functions. However, because CNNs have a high machine cost, resource-constrained devices that support onboard security schemes are difficult to deploy. Distributed architectures should solve this problem. In this design, a light deep neural network (DNN) is introduced and equipped, but the algorithm is entirely trained in the strongly categorized neuron [216], with a subgroup of similar output groups on board. The advancement of CNNs is primarily targeted at image detection. As a result of their wide use, CNNs are used to create accurate and consistent models of image ID and classification for massive public image databases such as ImageNet [217, 218]. CNNs also display their worth in a variety of other applications. According to one research [219], a CNN-based IoT protection malware identification framework for Android could be created. CNN is used to acquire major malware identification characteristics from raw data.

The main argument for CNN usage is that sufficient functionality is taught concurrently with classification, eliminating the extraction step needed for conventional machine learning and producing a full model [219]. However, attackers may use the robust learning capacity of CNNs as a weapon. A past study [220] depicted that a CNN algorithm is efficiently capable of breaking cryptographic applications.

(2) Recurrent Neural Networks (RNNs). Recurrent neural networks (RNNs) are a crucial type of deep learning algorithm. RNNs are suggested to deal with temporary results. The provision of current performance is dependent on an interpretation of the similarities between several previous examples in many implementations. The neural network performance is therefore determined by its current and previous inputs. Since the input and output layers continue to be distinct, a feedforward NN is unavoidable in this design [221]. When the backpropagation algorithm was created, it was primarily used for training RNNs [208, 222, 223]. RNNs are recommended for appliances that need sequential inputs (e.g., sensor, text, and voice data) [222]. An RNN has a sequential data storage layer and knows many elements of the recurrent cells' secret units [224]. In addition to the network details, the secret units are updated and continuously changed to represent the network's current state. The RNN forecasts that the previously hidden state would be activated when working on the currently hidden state. RNNs are applied because they can efficiently manage sequential data. This skill is useful for a variety of activities, including the identification of dangers where the patterns of the danger are time-based. As a result, recurring associations may be used to strengthen neural networks and uncover prominent behavioral trends. The main disadvantage to RNNs is that gradients disappear or burst [225]. RNNs and their derivatives have outperformed in a variety of sequential data implementations, including speech recognition and translation [226–228]. Furthermore, RNNs may be used to secure IoT computers. IoT networks gather massive volume of sequential data from a variety of sources, including network traffic patterns, and are required to identify a variety of possible network attacks. An earlier study [229] checked the feasibility of RNN analysis of network traffic

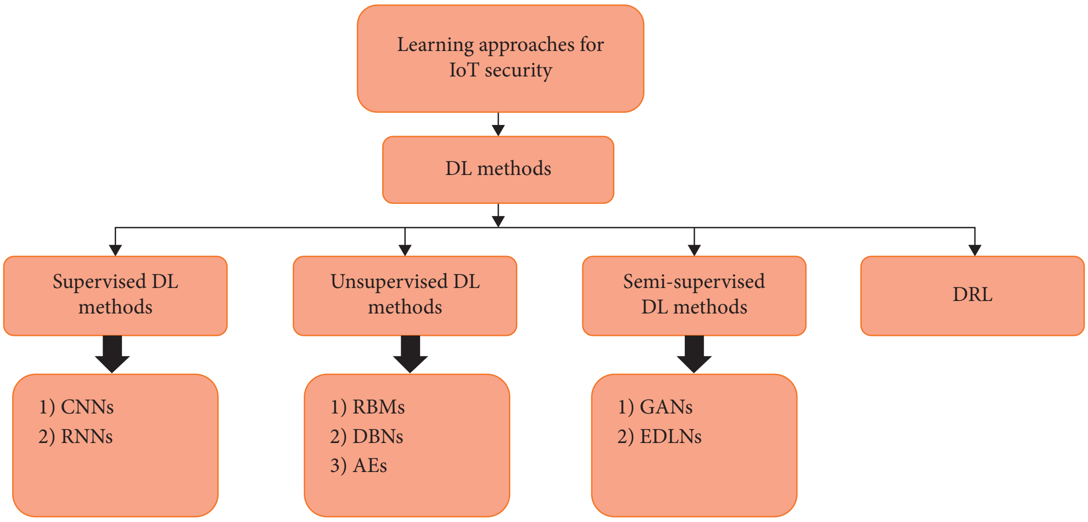


Figure 5: Various DL methods for IoT.

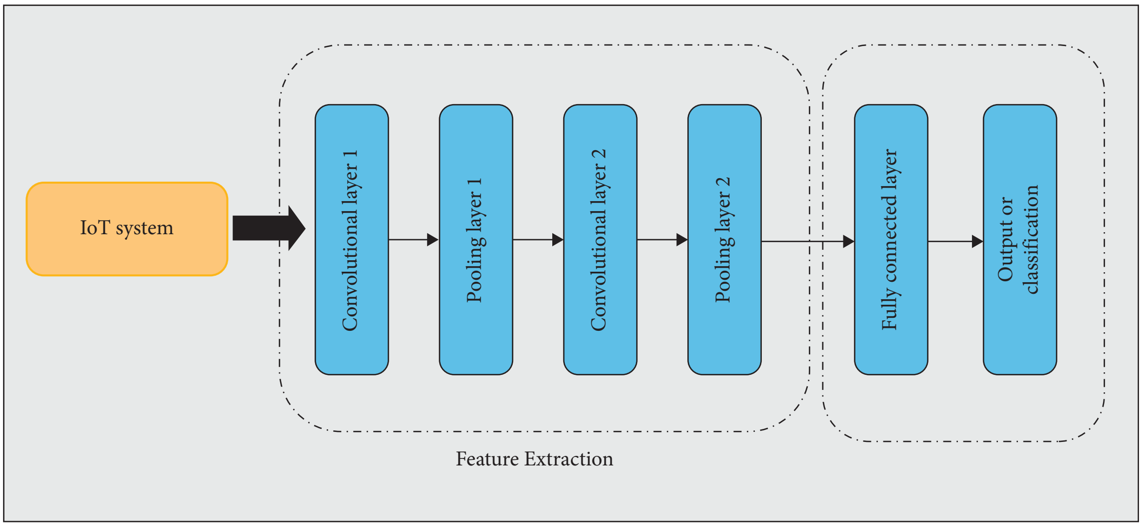


Figure 6: CNN for IoT security.

activity for identifying possible attacks (malignant action) and validated the RNN utility for network traffic classification to effectively detect malignant behavior. RNNs are also a viable solution in real-world situations. The investigation of RNNs and their variants is critical for improving IoT system protection, especially against serial-based attacks.

6.2.2. Unsupervised DL. We discuss the most often used unsupervised DL techniques, including deep restricted Boltzmann machines (RBMs), deep belief networks (DBN), and autoencoders (AEs) in this section.

(1) Restricted Boltzmann Machines (RBMs). Uncontrolled RBMs are deep generative models [230]. An RBM is an entirely undirected model in which nodes are not bound within the same sheet. RBMs are divided into two types of layers: exposed and unseen. The input is used in the visible layer, while the opaque layer comprises latent variables of various levels. RBMs accumulate data features in a hierarchical manner and are applied as hidden variables in the subsequent layer to record features in the initial layer. The study in [231] presented a model for detecting network anomalies that overcome difficulties in developing this model. This issue includes creating labeled data needed for effective model testing due to the multipart erratic nature of network traffic data collection. The second issue is that irregular behavior often evolves over time. The model can then be continually modified, allowing new attack types to be recognized and anomalies in a variety of network environments to be observed. The researchers recommended a learning model focused on a discriminatory RBM in [231], which they chose because of its ability to combine generative models with enough classification precision to identify a half-controlled network anomaly even though training figures are incomplete. However, their experimental findings revealed that the

discriminative RBM classification exactness was reduced when measured on a different network dataset than the data used to train the classifier. More research is needed to determine if an exception to a classifier in a variety of network contexts may be extended. A single RBM may only display a limited range of functions. However, RBM may be used to build DBN in significant ways by piling two or more RBMs. The part that follows goes into more detail on this strategy.

(2) Deep Belief Networks (DBNs). The generative approaches for DBNs are taken into account [232]. A DBN comprises stacked, layer-free RBMs that run greedy workouts in a stable, unmonitored setting. In a DBN, teaching is done layer by layer, with each layer being an RBM trained on the previously trained layer [224]. The initial characteristics [224] are learned using a greedy layer-specific unmonitored technique during the pretraining level. The top layer is finished using a softmax layer during the fineness phase [229]. DBNs have been successfully used to detect malware attacks. An earlier study [233] suggested a method for the security of mobile edge computers by the use of a profound learning technique to detect malicious attacks. DBN was used for automatic identification. Compared to machine learning-based algorithms, the proposed DBN model significantly improved malware detection precision [233]. This finding demonstrated that deep learning approaches, specifically DBNs, outperformed conventional manual malware identification feature engineering methods. An EA was combined with a malware detection method utilizing DBN in a recent study [234]. By non-linear projection, an algorithm in AE DL was applied to reduce the dimensionality of data and delete only the significant functions. DBNs are unregulated learning methods that are trained on unlabeled data to reflect significant features. Although DBNs use conflicting convergence to minimize processing time, they do not function for onboard computers that have limited resources.

(3) Deep Autoencoders (AEs). A deep AE is an unsupervised learning neural network that has been learned to replicate its input to output. A secret layer h specifies a code used to describe the input in an AE [210]. An AE neural network is split into two parts: the encoder function $h = f(x)$ and the decoder function $r = g(h)$, which tries to replicate the data. The encoder receives the feedback and transforms it into an abstraction known as a code. Following that, the decoder obtains the built text, which was originally generated to reflect the data, in order to reconstruct the original input. The learning phase in AEs can be completed with the least amount of reconstruction error [44, 235]. However, AEs cannot be taught to precisely reproduce the feedback. AEs are also constrained by being able to provide an approximate reproduction only by merely copying inputs similar to the training outcomes. The model must prioritize which input characteristics should be copied; therefore, useful data features are continually learned [210]. AEs have the ability to be useful for function extraction. In contrast, AEs provide a significant amount of computing time. Although AEs can learn to collect the characteristics of the training data effectively, if the training dataset does not match the test data, they can confuse the learning process instead of reflecting the data collection. In [236], network-based AEs have been used to recognize ransomware, and AEs have learned the latent representation of a dynamic function set, focusing on the cyber system's vector. The AEs outperformed the standard ML algorithms such as KNN and SVM in terms of detection efficiency [236]. In other research [234], A DBN was added to create a malware detection method that was then used to reduce data dimensions by non-linear mapping so that only the important features could be eliminated. Next, the algorithm in DBN learning was learned to identify malicious code.

6.2.3. Semi-Supervised or Hybrid DL. This section discusses the most traditional deep hybrid learning approaches. Among the hybrid DL approaches are generative adversarial networks (GANs) and network communities (EDLNs).

(1) Generative Adversarial Networks (GANs). GAN, which was recently pioneered by Goodfellow et al. [237], is already an exciting platform for deeper learning. As seen in Figure 7, two models, both generative and discriminatory, are trained concurrently by a GAN approach using an opposed mechanism. The generative model learns the data distribution and outputs data testing, while the discriminatory model predicts the probability of the results from the evaluation rather than the generative model. The objective of training the generative model is to increase the probability that it is wrongly classified by the discriminatory model [237, 238]. By changing the sample dataset, each phase trains the generative model to fool the discriminator. The model serves as a generator. In this regard, the discriminator is given many individual data samples from the training array and the generator samples. The discriminator is used to distinguish between actual and fake specimens (from the training dataset). Using incorrectly labeled samples, the outputs of unequal and generative models were quantified. The following edition's versions are then revised. The performance discriminative model aids in the generation of samples for the next iteration while optimizing samples for the next iteration [44]. GANs were recently added to IoT protection. In [239], architecture was built to protect the cyber field of IoT networks, including the training of in-depth learning algorithms to distinguish between ordinary and abnormal computing. GAN algorithms were used in the suggested architecture for the preliminary analysis, and the test results demonstrated the architecture's efficacy in detecting suspicious system activity [239].

Since GANs can learn various attack scenarios and produce a zero-day attack-like sample, they can provide algorithms with samples that are not available from current attacks. GANs are well suited for semi-supervised classification instruction. Since GANs are not sequentially needed to produce several accesses in the samples, they can produce samples faster than fully transparent DBNs. In GANs, sampling requires only one stage in the model, while RBMs need an unknown number of Markov chain iterations [237, 240]. GAN teaching, on the other hand, is risky and challenging. A GAN cannot be used to produce different data such as text [237, 240].

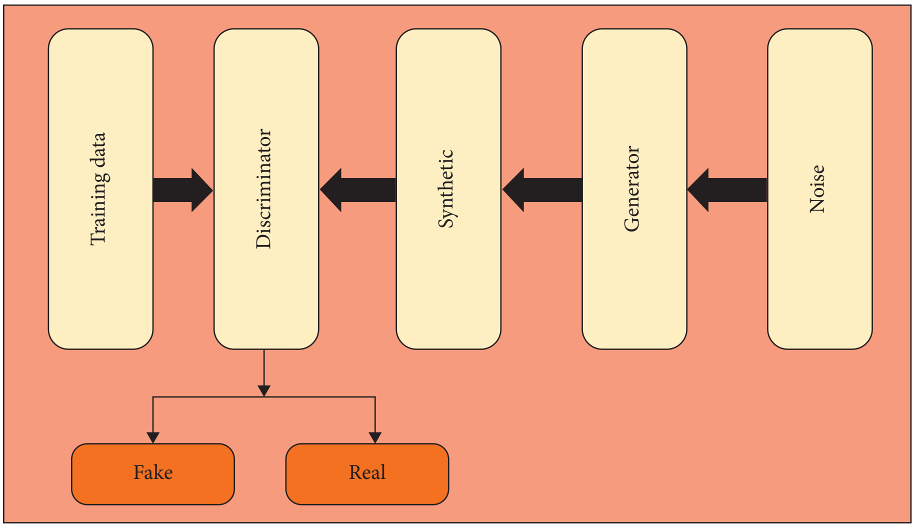


Figure 7: Working diagram of GAN.

(2) Ensemble of DL Networks (EDLNs). It is possible to create EDLNs using hybrid or discriminative and generative models. An emerging algorithm in deep learning, ensemble learning (EL), employs a variety of classification approaches to improve its performance [209, 241–243]. Homogeneous or heterogeneous multiclassifiers are often used to get an accurate result. Most issues can be solved using the various methods used by EL. When compared to other single classifier methods, EL takes a long time. Anomaly detection, virus detection, and intrusion detection are all typical applications for EL [168, 175, 244]. Combining DL classifiers may assist in achieving model variety, increasing model performance, and extending model generalization using the EDL approach. EDL's key drawback is that the system's temporal complexity may be significantly enhanced.

Due to numerous neural networks' training, assembling DNNs does not seem to be a feasible alternative due to the potential for a significant rise in computing cost. Deep networks may be trained on high-performance hardware using GPU acceleration over the course of many weeks. Explicit/implicit ensembles achieve the conflicting objective of training a single model. It acts as an ensemble of training several neural networks apart from incurring extra or as little additional cost as feasible. In this instance, the training time of an ensemble is identical to that of a single model. In implicit ensembles, model parameters are shared, and the model averaging of ensemble models is approximated by a single, unthinned network during test times. In explicit ensembles, however, model parameters are not shared. The ensemble output is determined by combining the predictions of the ensemble models using various methods such as majority voting, averaging, and so on. Dropout [245] generates an ensemble network by arbitrarily removing hidden nodes from the network during training. During the testing period, all nodes are operational. Dropout offers network regularization to prevent overfitting and adds sparsity to the output vectors. Training an exponential number of models with standard weights and providing an implicit ensemble of networks during testing reduces overfitting. Randomly dropping the units prevents coadaptation by making the existence of a specific unit unpredictable. The network with dropout takes 2 to 3 times longer to train than a regular neural network. Therefore, a suitable balance must be struck between the training duration of the network and overfitting. DropOut is described in detail in DropConnect [246]. In contrast to DropOut, which eliminates each output unit, DropConnect randomly eliminates each connection, introducing sparsity in the model's weight parameters. Similar to DropOut, DropConnect generates an implicit ensemble at test time by deleting connections (setting their weights to zero) during training. Both DropConnect and DropOut have a lengthy training period. To address this issue, deep networks with stochastic depth [247] sought to minimize the network depth during training while maintaining it during testing. Stochastic depth is an enhancement on ResNet [248] in which residual blocks are eliminated at random during training, and transformation block connections are bypassed via skip connections. Swapout [249] extends DropOut and stochastic depth.

Many deep learning algorithms will be able to outperform separately deployed algorithms if they operate together. EDLNs can be generated by integrating generative, discriminatory, and hybrid frameworks. EDLNs are frequently applied to address complicated problems, including uncertainty and a wide variety of dimensions. An EDLN is a stacked collection of heterogeneous (classifiers from separate families) or homogeneous (classifications from the similar family) classifications. They are used to boost variability, precision, efficiency, and widespread [250]. For example, the authors in [251] used a sparse autoencoder (SAE) to extract attributes and the softmax activation with a regression layer to create classifiers. The evolutionary findings indicate that we could obtain a higher level of efficacy than that in previous studies utilizing a semi-supervised intrusion detection technique. EDLNs have proven to be surprisingly successful in a variety of applications, such as activity detection for humans, but their usage in IoT protection necessitates additional testing, especially the ability to deploy light homogeneous or heterogeneous graders in a dispersed setting in order to increase IoT security system accuracy and efficiency and address machine challenges.

6.2.4. Deep Reinforcement Learning (DRL). Reinforcement learning (RL) has been introduced as an efficient technique of improving a learning agent's methods and determining the optimal solution through evaluating and failing to achieve the best long-term goal without previous environmental awareness [252]. RL is a kind of ML. An agent learns how its actions affect the environment via trial and error in RL. After each action, it calculates the reward and then proceeds to the next state [253]. It is possible to utilize RL to tackle very complicated issues that are intractable with traditional methods since it focuses on long-term outcomes. Moreover, real-world issues are assumed to be Markovian models in RL (which is not the case in reality), where the agent is in a state s at all times, performs action a, and gets an integer reward before changing states s' according to the dynamics of the environment, which is represented by $p(s'|s,a)$ in this case. If an agent is trying to maximize its returns, it will try to learn a policy based on its observations or mapping of those observations to actions (expected sum of rewards). Unlike in optimal control, in reinforcement learning, the algorithm only has access to the dynamics $p(s' \mid s,a)$ via sampling. DRL techniques use deep learning to solve Markov decision processes (MDPs) like these, frequently modeling the policy $\pi(a|s)$ or other learned functions as a neural network and building specific algorithms that operate well in this environment.

Deep reinforcement learning (DRL) approaches, for example, a deep Q-network (DQN), have been used in a variety of mobile edge computing applications to solve high dimension problems while still providing scalability and download performance [254]. The deep Q-network [231] is a recently used strengthening tool. Many proposed deep Q-network upgrades have been proposed, such as dual Q-learning [255], continuous monitoring by deep RL [256], and priority replay of knowledge [257]. The curse of dimensionality restricts its application to real-world systems, making it inapplicable. In addition, it requires a lot of data and computing to run. As RL has a number of drawbacks, it is often used in conjunction with other ML approaches. DRL is a standard combination of RL and DL.

The authors in [254] investigated access management and download for mobile edge cloud computing, as well as the integration of blockchain and DRL in application schemes in IoT networks. In another area of research, DRL was used to protect cyber security. In [258], the authors investigated many DRL cyber protection approaches, such as DRL-based cyber-physical network security mechanistic mechanisms, automatic intrusion prevention tactics, and multiagent cyberattack mitigation model DRL-based game theory. It may be a step forward to investigate these approaches within the context of IoT.

6.2.5. Application of DL in IoT Security. DL methods have been established for signal authentication in IoT settings [259, 260]. In order to derive arrays of stochastic properties from IoT device signals, Ferdowsi and Saad [259] proposed an LSTM architecture. The properties of cyberattack control in IoT systems were then watermarked, and the complex extraction of functions often helped detect eavesdropping attacks. This solution would not, however, apply to very large IoT settings as it is prohibitively difficult to authenticate all IoT devices in a centralized cloud service. In order to solve this challenge, we have merged the LSTM-based signal authentication process with game theory methods for a mixed Nash balance (NE) strategy. Although this approach is capable of handling many IoT modules, it is also highly dynamic and not suitable for IoT environments. Authentication in DL-based systems using LSTM is also recommended for IoT environments [261] since it has been designed for device recognition to be resilient to signal imperfections. Nevertheless, the method is effective for system recognition and cannot notice other severe attacks. DNN has been updated by utilizing intrusion detection systems (IDSs) to classify substantial attacks in the IoT environment [262]. In order to validate DNN performance, cross-validations and subsampling were used, and parameters in DNN were reset with a method for grid quest. While DNN performed well on many datasets, including those with imbalanced and distorted results, it was concluded that the determination of learning parameters for the grid search took more time. DL approaches have been used to defend Social IoT (SIoT) [263], with DL methods spreading via fog nodes to enable the identification of a distributed threat. The DL solution detects the following kinds of attacks with SIoT: sample, DoS, U2R, and R2L. The analysis exhibited that the DL solution outperforms other machine learning strategies, while more research is required on network intrusion detection based on payload. A deep learning methodology has been developed for dense random networks [264] to identify network attacks. The dense network-oriented DL approach focused on these metrics was used to observe attacks on IoT gateways. The main drawback of this solution is that parameter setting of the dense random network is needed since the non-optimal parameters are not properly classified. In IoT healthcare environments, protection and privacy have been maintained by the use of layering-based deep Q-networks [265] that access control, enable authentication, and mitigate intermediate attacks in the IoT environment. Packet features including protocol, IP address, post number file type, frame number, and frame length were initially gathered and stored in a local database. Deep Q-networks are used to identify medical data according to the derived functionality and the classification is achieved using the packet functionality since the extraction of optimal features from data packets achieves a higher degree of accuracy. A deep Eigen space learning approach has been suggested by the Internet of Battlefield Things (IoBT) for malware detection, [71] in which the Operational Code (OpCode) device sequence is used to classify malware. The OpCode interface was translated to vector sequences, and a deep Eigen spatial learning technology was used to differentiate between beneficial and malicious programs. The entire process started with the development and classification of a graph for each sample. This technique includes considerable computation, which limits its performance when handling big datasets. Identification of DDoS attacks was conducted in IoT systems using IoT-specific network features such as tiny endpoints and day-to-day packet cycles [142]. A network-dependent method was also suggested for detecting IoT botnets on the basis of deep autoencoders [266]. Five separate cycles of N-BaIoT functions, such as packet jitter, packet count, and packet size, have been obtained. This approach only incorporates static characteristics that restrict the self-encoder efficiency and require optimum feature selection to increase the accuracy of the autoencoder. Bidirectional LSTM-RNN (BLSTM-RNN) was applied in [267] to detect IoT botnets, and word embedding was used for text identification and the transformation of attack packets to integer sizes. Although BLSTM-RNN detects botnets with a limited number of attack vectors, it does not work when the count of attacking vectors increases. Autoencoder was introduced in a fog-enabled IoT framework for stacked unmonitored deep learning approaches [60], and architecture of fog-based IoT was built to improve attack detection latency and scalability [268]. While deep learning models exceed shallow attack detection learning algorithms, stacked autoencoders increase the computer's runtime and its accuracy. In Industrial IoT (IIoT), the systems of network intrusion detection are applied to secure a network against a range of security threats [268]. For intrusion detection, a deep autoencoder and a deep feedforward neural network are deployed, and feature transformation and normalization are used in the overall process. This method requires more research to establish how different intrusion mitigation protocols should be handled in the IIoT [269]. Significant routing attacks have been defined and evaluated with the following characteristics, such as decreased rating, hello flood, and change of the version number: average receipt speed, packet number, transmission rate, average transmission period, packet counting, and overall transfer time. Without accessibility, detection effectiveness and the possibility of detecting such extreme IoT attacks were impaired. The work of Diro and Chilamkurti [263] proposed a scheme to identify distributed attacks on IoT and contrasted their effectiveness with conventional machine learning methods and distribution networks as well as the centralized approach of detection. The results were also tested for two and four-class grades; the two-class grades included regular and attacks, while the four-class grades included U2R, DoS, probe, and natural and remote-to-local grades. The utility of distributed detection has been evaluated on a range of network training engines, relative to standard IoT learning approaches for DL intrusion detection. The findings showed that the distributed architecture exceeded the central structure with an accuracy of detection varying from 96% to 99%. Furthermore, results exhibited that DL approaches were more precise than conventional learning methods and had a lower false alarm rate of 0.85% compared to machine learning methods. DL had a 99.27% recall rate and an average 96.5% recall rate, while ML had a 97.50% recording rate and an average recording rate of 93.66%. These results show that deep learning techniques in distributed IoT environments are highly likely to identify cyberattacks. Experiment results show that distributed sensing systems outperform hierarchical methods in detecting cyber threats because they exchange variables, preventing the formation of local minima during planning. The research can be covered by a comparison of distributed approaches to deep learning to different traditional methods of learning on different datasets. Techniques for the analysis of the network load data to identify intrusions through key trends may also be further explored.

7. Solutions to IoT Threat Using DL or ML Algorithms

Privacy and security concerns can be mitigated in many ways. In Section 3, we described several types of threats in IoT. There were no solutions and discussions about how to ensure security or privacy in the IoT system. Hence, we concentrate on recent works suggesting privacy and security-preserving methods for the IoT in this section. We illustrate the solutions suggested by DL or ML algorithms as a tool for ensuring privacy and security.

A security program is a collection of policies and procedures designed to safeguard an organization's most sensitive data and assets. Instead of concentrating on people's private details, it highlights statistics and other facts. On the other hand, passwords, login information, and other sensitive data are the primary targets of privacy programs.

Safeguarding privacy, maintaining data and information's integrity, and making sure it is readily accessible are the three pillars of security. The right to the confidentiality of one's own and one's employer's private data is a cornerstone of privacy. Security measures may help provide some level of confidentiality, and the secrecy of credentials and access to data is essential to a robust security framework.

Machine learning (ML) is a data processing technology that is applied in all frameworks' data processing pipelines. For instance, a machine learning model may assess data flow into a network in order to get an up-to-date decision. Poisoning or exploratory attacks on the input data from the source to the IoT nodes, as well as on the IoT nodes to the ML model, are possible. Inversion and integrity attacks are feasible on the output [270]. As a result, the privacy and security of a system cannot be compromised simultaneously.

7.1. Security Solutions. On the basis of DL and ML methods, several suggested security measures are shown in Table 1. According to the work of Diro and Chilamkurti [61], fog computing decreased the threat of spying in communications as well as attacks in MiTM by limiting interaction to IoT gadgets in close proximity during flooding attacks. Based on this, they implemented their model by applying the long

Table 1: Summary of several security solutions in IoT using DL and ML.

IoT application	Threat	Dataset	Kind of threat	Algorithms	Accuracy	Reference
Healthcare	MiTM	Private	Impersonation	LSTM RNN	—	[78]
IoBT	Malware	Private	Code injection	DCN	98.37%	[71]
Wi-Fi	MiTM	AWI	Impersonation	ANN	99.92%	[79]
Fog	DoS	NSL-KDD	Flooding	Softmax	99.20%	[60]
NIDS	Anomaly	Kyoto 2006+	Anomaly	Softmax	88.39%	[271]
Android	Malware	Drebin and AbdroZoo	Malware	Ensemble + LR	98.10%	[73]
Fog	DoS	AWID and ISCX2012	Flooding	LSTM	AWID (98.22%) and ISCX2012 (99.91%)	[61]
IoT	Botnet	NIMS botnet and UNSW-NB15	Flooding	AdaBoost	UNSW-NB15 (99.54%)	[69]
Android	Malware	Private	Malware	KNN, C4.5, NB	—	[74]
RF communication	MiTM	Private	Impersonation	ANN	99.90%	[80]
Android	Malware	Multiple sources	Malware	Ensemble	98.40%	[272]

short-term memory (LSTM) method, which can keep track of historical data. In order to compare their findings with LR, they used the ISCX2012 dataset. It included 71,617 instances of DoS attacks and 440,991 instances of normal traffic. Although training the LSTM model took significantly more time than that of the LR model, it was 9% more accurate. After that, another work [273] used the techniques from the Aegean Wi-Fi Intrusion Dataset (AWID). This dataset includes normal traffic (1.633.190 instances for training while 530.785 instances for testing), injecting attacks (65,379 instances for training while 16,682 instances for testing), impersonation attacks (48,522 instances for training while 20,079 instances for testing), and flooding attacks (94848 instances for training while 8097 instances for testing). For multiclass classification, LSTM outperformed softmax with a 14% increase in accuracy. Similar research conducted by Abeshu et al. found that IoT devices' resource limitations rendered them vulnerable to DoS attacks [60]. In a widely dispersed network like the IoT, traditional machine learning techniques are less scalable and less accurate for detecting cyberattacks. Data from billions of IoT devices allow deep learning models to outperform shallow algorithms in learning.

The authors in [60] claimed that the majority of the used deep learning architectures applied pretraining for feature extraction. It enabled the detection of abnormalities and therefore decreased a network administrator's workload. Nevertheless, their work concentrated on networked deep learning through model and parameter exchange for fog computing applications. Fog computing decreased the IoT devices' load on processing resources as well as storage space. As a result, it is the perfect location for detecting an intrusion. For fog-to-things-based computing, parallel computing is required for the traditional stochastic gradient descent (SGD) algorithm. Consequently, a vast number of data produced by the IoT will choke the centralized SGD. Accordingly, the study provided a distributed deep learning-driven IDS based on a dataset like NSL-KDD, in which stacked autoencoder (SAE) was applied to extract features as well as softmax regression (SMR) was employed for classifying the data while SAE performed better as a deep learning than existing shallow algorithms according to accuracy, FAR, and DR. There is evidence to support both assertions of the authors in [60, 61] that deep learning models outperform shallow machine learning algorithms. Tan et al. [63] attempted to identify DoS using a triangle-area-based technique in multivariate correlation analysis (MCA). The data that made it to the target network were utilized to develop features that minimized overhead. Geometrical connections between two different characteristics were identified by applying the "triangle area map" module to improve the accuracy of zero-day attacks for detection. According to the researchers in [63], they applied Earth Mover's Distance (EMD) to determine the differences from observed traffic to a prebuilt ordinary profile, which they believed would help them improve their results from [63]. Using the KDDCup99 and ISCX datasets, MCA was applied to extract characteristics from network traffic and assess the findings for anomalies. Their findings are based on a sample-wise correlation of 99.95% on KDD data as well as 90.12% on ISCX. In any event, neither the size of data in the research nor its effect on various sample sizes was revealed. MCA was not a practical technique since the change expected was not linear. In the Internet of Things, a botnet attack is a different type of DoS attack in IoT. The authors in [69] created an IDS that integrates ANN, DT, and NB to fight botnet attacks against DNS, Message Queuing Telemetry Transport (MQTT), and HTTP. It was chosen to utilize ANN, DT, and NB to better differentiate between malicious and benign vectors since their cross-entropy values were similar. The detection rate and false-positive rate were used as performance indicators, and their proposed ensemble beat each individual algorithm inside it. The accuracy was 99.54% on the UNSW dataset and 98.29% on the NIMS dataset. MiTM attacks, which are very similar to DoS attacks, are one of the most frequently occurring attacks on the network in IoT.

Numerous technological solutions have been suggested for different application scenarios in connection to this. Due to the fact that traditional feedforward neural networks are incapable of capturing time-series and sequence data owing to their causal structure, an impersonation attack on smart healthcare was prevented with the use of LSTM-RNN, according to the researchers [78]. Additionally, the researchers were able to address the vanishing gradient issue associated with the RNN method and improve accuracy. The predicted value was first calculated using a three-month log of the dataset (for a diabetic patient who is receiving insulin injections). DL and gesture recognition were merged if the estimated and expected dosages varied by more than a certain threshold. They lacked, however, a detailed understanding of the model and analysis. The researchers in [80] employed physical unclonable function (PUF), a unique silicon chip feature that may be applied as a foundation for radio frequency (RF) communication authentication, to defend against impersonation attacks. Moreover, the authors were able to identify the device and train their system on it as a consequence of these offsets, all before knowing the gadget's degree of accuracy. The evaluation metrics were analyzed using the ANN MATLAB toolbox. According to the simulation results, machine learning can help detect 4,800 transmitter nodes with 99.9% accuracy and 10,000 nodes with 99.9% accuracy under different channel conditions. As indicated, multifactor authentication may be used alone or in combination with other security measures. In a secure server, PUF's intrinsic and low-cost nature enables the storage of the physical values of each wireless sensor rather than utilizing existing key-based authentication. Nonetheless, the researchers made the incorrect assumption in their approach that data saved on the PUF server are safe. Aminanto et al. [79] extracted features by applying C4.5, ANN, and SVM, with ANN acting as a classifier [79]. The technique of deep feature selection and extraction started with the extraction of features using SAE, followed by feature selection applying C4.5, ANN, and SVM, and finally by classification using ANN. The accuracy of the research was 99.92% due to the use of the AWID dataset, which had the lowest accuracy for impersonation attacks in a prior study, accompanied by the work of Kolias et al. [273].

Statista [274] held that the global mobile phone user base will surpass three billion by 2020. Due to the increased usage of mobile phones, they become increasingly vulnerable to virus attacks. According to the work of Azmoodeh et al. [71], OpCodes may be employed to differentiate between safe and malicious software. As a consequence, global feature selection introduces inefficiencies and may potentially reduce system proficiency, especially for the imbalance dataset. They claim that no one has ever tried to combine OpCode with DL for IoT previously. Deep convolutional networks and Eigenspace techniques were used, and the accuracy was 99.68%, while the recall and precision were 98.37% and 98.59%, respectively. Similarly, Wei et al. [74] utilized dynamic analysis to extract malware features. They trained the classifier using functional application classification on clean and damaging data and then used KNN to divide the data into recognized categories during the testing phase. We performed tenfold cross-validation using the J48 decision tree and NB. Depending on the performance metric used, this study achieved 90% accuracy. The work of Aonzo et al. [72] applied static analysis methods rather than dynamic analysis (see [74]) to extract features, taking into account all APIs that had not previously been studied. The most commonly used qualities by earlier researchers served as a roadmap for developing new characteristics. They offered 98.9% accuracy using the dataset on the second-largest malware testbed. With the advancement of sophisticated infiltration techniques, static analysis became outdated, necessitating the use of a dynamic methodology [73]. The attackers utilized static analysis because they exploited deformation technologies to avoid recognition, while dynamic investigation approaches showed promise due to their resilience to similar tactics. The authors in [73] created the EnDroid framework in response to these issues. When it came to categorizing the data, the suggested model applied "chi-square" feature extraction and a combination of five ML algorithms (linear SVM, decision tree, boosted trees, random forest, and extremely random trees), with LR serving as the meta-classifier. The dataset was created by combining the Drebin and AbdroZoo databases, yielding a 98.2% accuracy.

For example, Wang et al. claimed that static string characteristics such as API and permission use retrieved from applications were the basis for the majority of current malware detection literature [272]. However, due to the increasing sophistication of malware, relying only on a static characteristic may lead to a false positive. To identify Android malware, the DriodEnsemble model used a combination of string and structural characteristics. RF, KNN, and SVM were applied to test the model against 1,386 good applications and 1,296 bad apps. However, using just string characteristics, the research was able to achieve 98.4% accuracy, which was higher than the recognition accuracy of 95.8% achieved by applying only structural features. It is a general method that looks for anything out of the ordinary and flags it as a potential security risk. Many researchers [271, 275, 276] have tried to use machine learning techniques to create safe intrusion detection systems (IDSs). To help with this, Javaid et al. [271] used an unsupervised DL method known as STL, which relied on SAE and SMR as its foundation. Two-class classification outperformed SMR using the NSL-KDD dataset; it was superior to five-class classification by a wide margin. It was suggested by Ambusaidi et al. [275] to use mutual information (MI) in an ML-based multiclass classification. Linear correlation coefficient (LLC) was utilized for the linearly dependent variable in mutual information feature selection (MIFS). The authors utilized FMIS + MI for the non-linear dependent variable, modifying the preexisting MIFS method [277] and demonstrating their originality. An additional motivation for doing this research was that prior studies had failed to explain the processes. Kyoto 2006+, NSL-KDD, and KDDCUP99datasets were used to compare performance, while accuracy, F-measure, FPR, and DR were used as metrics. Anomaly detection using LSTM was the focus of Fernandez Maimo et al. [276]. DBN and SAE models (where the prediction may be calculated by utilizing matrix operations after an activation function) were used to reduce features because of their comparable structure, while features were extracted from flows of networks employing weighted loss functions [276]. The authors claim to have achieved up to 95% precision after utilizing the CTU-13 botnet dataset to build their model [276]. ML algorithms have been used in research that claims to decrease cyberattacks successfully. In contrast, a past study [278] used deep feature embedding learning (DFEL) as it was faster than conventional machine learning algorithms for training data. Their approach was compared by utilizing the datasets from UNSW-NB15 and NSL-KDD, and the recall value of the Gaussian Naïve Bayes classifier improved between 80.74% and 98.79%, while SVM's runtime was decreased substantially from 67.26 seconds to 6.3 seconds as a result. The previous IoT security methods were also centralized and cloud-based, which resulted in significant high power consumption and latency for end devices [279]. Fog computing was utilized in two stages to build the suggested IDS for IoT in a distributed manner. The identified threats were then compiled and evaluated on a cloud server in the second phase. The novel method outperformed the current NB, ANN, and conventional ELM in terms of accuracy, FRP, and TPR. Fog computing-based attack detection was shown to be quicker than cloud computing-based attack detection in the experiments conducted on the Azure cloud. However, no current ML/DL-based fog computing algorithms were utilized to compare the findings of the research.

7.2. Privacy Solutions. Table 2 shows some suggested privacy-preserving ML and DL methods. A MiTM attack compromises both security and privacy. Several number of works utilize ML techniques to defend against various MiTM threats. Table 2 shows some suggested privacy-preserving ML and DL methods. A PHYlayer authentication scheme based on IAG lowered the overall communication burden and improved detection precision. When working with an updated dataset, the researchers were able to improve FAR, DR, and computing costs. There was also a problem with the wearable device that was highlighted by Aksu et al. [280] in addition to user authentication difficulties. However, it is necessary to authenticate the device itself. Similar to MiTM devices, these devices may be used to authenticate users. However, if anything goes wrong in the background, it may end up giving the attacker complete access to the system. A more powerful base device can be reached only via Bluetooth with encryption and authentication. It was considerably more secure to utilize hardware-based fingerprinting since the encryption and device name secrets might be stolen so readily. The suggested framework in [280] made use of a timing technique of classic protocol packet-based and inter-packet timing-based analysis in Bluetooth. This process has a structure of four stages. Bluetooth classic packets were first captured. The characteristics were then retrieved in a second phase. The fingerprints were produced in a third stage by using probability distributions. The saved fingerprints from step three were also matched to any fresh incoming data from wearable devices as the last step in order to identify any unfamiliar wearable devices. The study claims to have achieved 98.5% accuracy by selecting the best algorithm from a set of twenty training results. You will need a large amount of data to build an ML model. For example, we can utilize past patient data to predict outcomes for each new patient. Patients, on the other hand, are apprehensive about disclosing their personal information. According to [282, 283, 285], research has attempted to address these problems. Non-linear kernel SVM was used in [285] to effectively categorize medical data while maintaining the privacy of both the service provider and the user data model. Zhu et al. [285] said that they were able to obtain 94% classification accuracy using their system, apart from sacrificing privacy. Users' private information and model outputs were categorized as model-privacy problems and learning-privacy problems, respectively, by researchers in [282]. This study relies on gradient values instead of actual data or assumes that the learning model is private, but the learned model is public or uses complex encryption techniques. Previous research has depended on these approaches [282]. The authors in [282] presented a uniform oblivious evaluation of multivariate polynomial algorithm that lacked complex encryption methods in contrast to the other research. In the end, their findings showed that the categorization data and models they learned were safe against a variety of intrusions. Model privacy was the subject of investigation [282]. However, the issue of student privacy was not addressed. This problem was addressed by Ma et al. [283] who said that although utilizing the public key to encrypt any user data was a popular technique for maintaining privacy, it came at the cost of key management. In the cloud, a cloud service provider delivers encrypted client data to a data training system that does not know what is being trained on. This is their suggested approach. They concluded from their analysis of the privacy-preserving DL multiple-keys (PDLM) that it had less efficiency than traditional non-private methods while still preserving privacy. To classify data privately, they used hyperplane decision-based private methods like decision trees and Naïve Bayes, together with private Naïve Bayes and decision trees. In a related study, it was discovered that the number of user-server iterations could be cut in half without compromising privacy. People's lives have been enhanced by Facebook and Twitter, yet privacy concerns have arisen as a result. Blacklisting methods were used by a number of businesses to screen out malicious traffic. According to this study, 90% of people will be victims of these attacks even before they are prohibited. Machine learning algorithms were used to evade these attacks successfully. However, because of their slower pace of learning, these algorithms were inefficient in real time. The authors in [281] described a multistage detection framework employing deep learning, in which the results were first detected at a mobile terminal and subsequently sent to a cloud server for additional computation. The authors stated that by utilizing the Sino Weibo dataset and CNN as a categorization technique, they obtained an accuracy of approximately 91%. When looking for a solution, researchers used distributed ML methods and collaborative IDS as well as ideas of dynamic differential privacy to protect a training dataset.

Table 2: Summary of several security solutions in IoT using DL and ML.

IoT application	Threat	Dataset	Kind of attack	Algorithms	Accuracy	Reference
Wearable devices	MiTM	Private	Authentication	Best of 20	Precision: 98.5%	[280]
MSN	Anomaly	Sino Weibo	Spam	CNN	91.34%	[281]
Distributed systems	Data privacy	Real world	Multiple	OMPE	—	[282]
Cloud	Data privacy	—	Data leakage	SGD	95%	[283]
WSN	MiTM	Private	Spoof detection	DQ, QL	—	[284]
Healthcare	Data privacy	Real world	Multiple	SVM	94%	[285]
MiTMO landmark	MiTM	Private	Spoof detection	Softmax	—	[54]
VANET	Data privacy	NSL-KDD	Inference attack	LR	—	[286]

8. New Insights in Machine and Deep Learning for IoT Security

Entrepreneurial or commercial off-the-shelf IoT devices are usually supported with the solutions of software that are insufficient to protect every IoT device or system [287, 288]. Since the IoT has many different use cases, the software-level security is poor. IoT security is an issue that some researchers [21, 289] are concerned about regarding privacy and security.

8.1. Data Privacy. IoT security is challenged by data privacy because of the significant risk of vulnerability, according to most research [290, 291]. Unauthorized access to data, eavesdropping, data fabrication, data alteration, and unlawful remote access using devices are some of the vulnerabilities [292]. As an example, personal information, such as names, addresses, phone numbers, insurance policy numbers, and bank names, is always at risk when it is stored on the cloud. Many IoT devices and apps, on the other hand, provide access to important information that might be used by attackers to gain access to the system. As a result, sensitive personal information that is unprotected and unencrypted may be exposed to an unauthorized party.

8.2. Vulnerabilities in IoT. IoT devices are now prone to several vulnerabilities. Services and data in the IoT may be susceptible to attack because of their sensitive nature [293]. Many IoT systems and a highly complex ecosystem in IoT can have increased risks from significant problems in cloud security [294]. Centralized management platforms and older systems pose substantial security risks for IoT devices [295]. When it comes to application layer security, it is possible for users to create weaknesses. One or more of many types of defects exist, including inefficient input/output filtering, poor encryption, and tampered authentication mechanisms. Few examples of vulnerabilities in IoT security are as follows.

(i) *Weak, Guessable, or Hardcoded Passwords.* To get access to a system, a user must utilize credentials that are readily brute-forced, publicly accessible, or impossible to modify [296–298]. Credentials that are both hardcoded and integrated into IoT devices constitute a threat to both IT systems and the IoT itself. Hardcoded or guessable credentials are also a benefit to hackers who want to target the device. In addition, the malicious attacker can already have access to the password of a machine if it has default passwords. In order to prevent unauthorized access, devices connected to the IoT should have measures in place, i.e., password expiry, password difficulty, and one-time password account lockout that compel users to alter the default credentials. The producers of IoT devices should, as a consequence, provide them with strong passwords straight out of the box to prevent security flaws.

(ii) *Inadequate Protection in Privacy.* Insecure, inappropriate, or unauthorized use of the personal information of users kept on the ecosystem in a device may lead to IoT security flaws [299–302]. Since IoT devices might be vulnerable, proper privacy protection must be provided for them.

(iii) *Vulnerable Interfaces in the Ecosystem.* Backend application programming interface (API), web, mobile, or cloud interfaces outside of the device ecosystem may be exploited to get access to the device or its components, making IoT vulnerable to attack [303–306]. Lack of authorization and authentication may also lead to IoT vulnerabilities [307, 308], as can inadequate encryption or lack of encryption [309], as well as a lack of output and input filtering [310]. It is possible to protect a connected device as well as create data via custom device authentication. A digital entity (computer, IoT device, etc.) may also securely send data to authorized recipients using digital certificates.

(iv) *Absence of Any Kind of Hardening Measures.* As a result of the absence of hardening measures, attackers may get critical information that might be used to help in remote attacks and achieve local control in IoT-based systems [296, 299, 311]. Account lockout, password, and complexity that forces anybody setting up a device to modify the default credentials are among the physical hardening methods [312–315]. Because of this, physical precautions such as security paradigms are needed to guard against IoT attacks and vulnerabilities.

8.3. Authorization, Authentication, and Identification. IoT devices have a number of security issues, including the inability to be identified, verified, and granted access to the network. The authorization, authentication, and identification of IoT devices is a major concern for many researchers [6, 316, 317]. Many IoT devices do not allow a single device to be uniquely identified, authenticated, and authorized, which makes things incredibly complicated.

Furthermore, there is a difficulty with authentication. To prevent unauthorized users from having full access to a network's resources, some kind of access control is required. A survey of IoT communication protocols conducted by the authors in [318] brought to light the fact that there are now just a few protocols that guarantee users' safety and confidentiality. As a result, additional research is required to improve a framework that can provide IoT device users with privacy and security.

8.4. Behavior-Based Mobile Device Authentication. Behavioral authentication on mobile systems identifies an individual according to unique qualities, including biometric authentication, that utilize patterns exhibited while networking with a system including a computer, tablet, or smartphone that contains a keyboard as well as a mouse. A secure authentication system is necessary to restrict access to tablets, cellphones, e-readers, smart watches, and laptop computers. Laptops, desktops, mobile phones, and tablets are no longer just tools for people; they are increasingly taking on their roles. These technologies have unlocked different ways to interact, play, and work. Because of their small size, they are easy to carry about in pockets, handbags, or other bags. However, mobile devices are susceptible to a variety of issues. The security and privacy of the user is at risk if the gadget is lost or stolen. It is possible to get threats from both strangers and close friends. Similarly, mobile gadgets are readily lost because of their mobility and portability. Users' private life and personal information might be made public if a thief gains access to these devices. They may also be vulnerable to extortion or blackmail. Moreover, a biometric technique aims to identify and detect the user. The United States National Science and Technology Council's Subcommittee on Biometrics separates biometrics into physiological and behavioral categories [319, 320]. As the name suggests, behavioral biometrics is concerned with identifying and quantifying human behavior patterns. The identification approach based on physiological features is quite accurate. Physiological biometrics, on the other hand, focuses on physical characteristics of the human body, such as a retinal or fingerprint scan. Conversely, behavioral biometrics denotes behavioral aspects of the human body. Behavioral biometrics analyzes data, including a user's screen pressure, navigational patterns, mobile or mouse motions, gyroscope position, typing speed, and so on. Behavioral biometrics recognizes a subject by employing behavioral qualities. Each subject is projected to differ from all others when investigated using one or more of these characteristics. Additional human aspects and behavioral biometric attributes and verification methods include gait analysis, keystroke dynamics, touchscreen, voice ID, hand waving, mouse usage characteristics, signature analysis, cognitive biometrics, electroencephalogram (EEG), profiling, and electrocardiogram (ECG). An important benefit of behavioral biometrics is that it may be used to authenticate users without the requirement for additional hardware [319]. To put it another way, adopting behavioral biometrics rather than physiological biometrics is more cost-effective. Analyzing an individual's physical characteristics is possible via the use of a variety of biometric tools, including retinal or iris scans, face identification software, and fingerprints. In the same way, it involves assessing how a person uses their pen, as well as their personality characteristics and other aspects of their everyday conduct. Authentication and identification are two of the most common uses of biometric technology. More secure systems might be created by using various authentication methods. Pin/password, authentication using pattern, speech recognition, face recognition, iris-based authentication, and fingerprint recognition are among the authentication systems mentioned as follows.

(1) *Fingerprint Recognition.* Fingerprint identification may also be accomplished with the use of a secret sign. It is described as a precise pattern of finger movement over the screen. Users may authenticate themselves using this pattern as a kind of biometric authentication. A biometric is a trait of a person's physical or mental makeup that cannot be duplicated. Using biometrics, it is possible to tell one individual from another. In other words, it is a way of figuring out someone's identity.

(2) *Iris.* The colorful part of the eye around the pupil is called the iris. It is the biometric that is often regarded as reliable. Because each person's iris has its unique patterning, a blood test may be performed accurately, quickly, and simply. The iris may be matched using a picture since the eye is a visible organ [321]. Many airports in the UK, including Manchester, Heathrow, Birmingham, and Gatwick, began using iris scanners in 2004 as part of a nationwide rollout. Their usage was later phased out since it was believed to take more time than ordinary passport inspections to complete the process. According to firms like EyeLock, the IoT and autonomous automobiles will benefit from iris scanning. Each individual can only have two different iris pictures since it is a fixed characteristic. Iris scanning equipment may take up a lot of room. Much closeness is also required.

(3) *Pin/Password.* A personal index number (PIN) or a secret pattern is the current form of authentication for cellphones, tablets, and laptop computers. Typically, a PIN requires four or more numbers to be entered by the user for verification. This code must be entered correctly for the user to access their device.

(4) *Fingerprint.* Fingerprint scanning is one of the least expensive biometrics, making it an attractive option for many organizations. Fingerprint images may be

captured using a tiny camera that can be incorporated into mobile devices like wearables or smartphones, making it very convenient. This means that mobile apps on devices with this hardware may be authenticated using this way. Because mobile devices have limited typing skills, password authentication is often unpleasant. If the child's fingerprints change, this is not the best biometric.

(5) *Facial Recognition*. In comparison to other technologies, facial recognition is a non-intrusive and low-cost option. Due to the widespread use of selfies, the smartphone is well suited for face recognition. As a result, smartphone makers have made significant investments in front-facing cameras. Using the device's screen, people may check to verify whether the camera is taking a picture of the right region of their faces. Some UK airports are already using facial recognition technology at ePassport gates. Mastercard's self-service payment app also takes advantage of it. A possible drawback is that illumination changes might alter the picture. When a person grows older or trims their hair, their facial features alter as well. Plastic surgery on the face has a significant chance of altering it. A significant difficulty for automated face identification is the failure of several face recognition algorithms to distinguish faces after cosmetic surgery. Additionally, attackers may exploit facial recognition technology [322, 323].

9. Challenges, Limitations, and Future Directions

Machine and deep learning algorithms have only recently been developed and are not intended for use in cryptographic applications. Two previous studies [142, 156] show, for example, that ML can be applied to hack a sample attack using SVMs and cryptographic constructs. Similarly, developers in [324] taught DL algorithms to decode cryptographic frameworks and concluded that DL would do so. Machine learning (SVM and RF) and logical process profiling algorithms were outperformed by CNN and AE algorithms. RNNs have previously been shown to be capable of learning decryption. The study of successful internal representations of this cipher may also be used to decode the enigma machine on an RNN with a three-thousand unit LSTM. The results also suggest that deep learning algorithms such as RNN can detect and manage polyalphabetic cipher algorithms for cryptanalysis [325]. Machine learning/deep learning research has the potential to advance the advancement of the Internet of Things.

As an enormous number of intelligent items are linked to IoT devices, it is critical that the endpoints of such devices be secure. Profiles, explicit trust connection, timestamping protocol, privileges, encoding, and so on all need robust authentication protocols [19, 326, 327].

9.1. Limitations of ML in IoT. The one disadvantage of simple ML approaches is that they need a large amount of data for model testing. The studied model is then used to approximate or categorize real-world implementation outcomes. However, it should be remembered that the whole procedure does not capture the whole spectrum of data characteristics and facilities. In this case, DL methods were used to address the shortcomings of machine learning strategies. Since DL can process vast amounts of data and its algorithms are flexible when the volume of data increases, model testing is advantageous and predictive accuracy can be enhanced. High-level functions and contrasts are derived dynamically and hierarchically from input data by DL. The majority of IoT implementations produce blank or half-marked results. Unlabeled data may be used by DL in an unsupervised manner to reveal valuable trends. Typical machine learning algorithms are only successful where there is a large amount of data on the label [45].

9.2. Limitations of DL in IoT. We conducted a thorough review, which revealed that existing research needs to be changed in order to reach higher protection requirements in IoT settings. Security issues are essential because authorization, entry security, system security, data integrity, intrusion detection techniques, and packet extraction play a role in the detection of anomalies. Security concerns are severe. The specific IDS algorithm, data preprocessing, function extraction, and the optimal set of features are all important factors in DL-based anomaly detection. Flexibility and planning are also common issues for profound learning approaches. The authors in [328] examined various DNN models and discovered that small precision improvements take a long time. Moreover, tuning the parameter is a significant issue since the number of layers and accuracy are linearly related. Many hyperparameters are thus expected if deep learning methods that are highly sensitive to data structure and size are to start optimally. The research challenges in the environments of DL-based IoT security contain the following:

(i) End-to-end safety (integrity, access management, authentication, confidentiality, and intrusion detection systems).

(ii) Data preprocessing, optimum function selection, and extraction.

9.3. Challenges of ML. As addressed further below, an obstacle is the insufficient collection of data for data-driven ML and DL methods.

(i) *Scarcity of Testing Datasets*. Datasets are available for the effective use of DL and computer education solutions. To validate and evaluate the output of various profound learning and enhancement learning algorithms, authentic databases from the actual physical world are used. The data include sensitive and personal knowledge that not only differentiates individuals but also their habits and way of life. Data created by BAN and other healthcare apps, for example, can jeopardize consumer safety, while data from intelligent homes can influence lifestyle and behavior. As a result, it is important not to jeopardize consumer safety when using ML and DL. Numerous methods of anonymization were used to anonymize data until it was used for analytics; nevertheless, the study revealed that these techniques could be hacked and models could be abused by adding fake data. It may be difficult to gather data while maintaining secrecy and privacy. Furthermore, issues such as how machine learning and deep learning algorithms would be applied, as well as the extent to which machine learning and deep learning algorithms may protect privacy, must be addressed. As a result, it is critical to investigate machine learning and IoT network deep learning analytics strategies for data security and consumer privacy safety. Keep in mind that simulation data cannot accurately represent real IoT scenarios in the universe. Furthermore, generating synthesis data to train and test deep learning models may be computationally costly.

(ii) *Data Imbalance*. When attacks are uncommon in an IoT environment, the datasets obtained for machine learning or deep learning are more likely to be unbalanced. The dependability of attack classifiers and intrusion detection systems would have a significant effect on these various datasets.

(iii) *Data Convergence*. It would be necessary to combine data from various IoT devices and network modules in order to construct machine learning and deep learning models. This may be daunting since data from various sources can differ in modality, granularity, complexity, and falsity.

9.4. Challenges of DL. ML, which is a technique for extracting information from results, has been used for both malicious and benign purposes. It has been discovered that future adversaries allow effective use of these ML and DL-based learning algorithms to crack cryptographic secrets. For instance, the authors employ recurrent neural networks for cryptanalysis. Additionally, erroneous data inputs to the ML algorithm result in an inefficient operation of the whole learning-based framework. Oversampling, an insufficient testing dataset, and function extraction are all issues to consider when applying knowledge to smart ecosystems.

9.5. Future Directions of ML. Artificial intelligence and machine learning have provided a major contribution to the progress of computer security. On the other hand, an advanced security framework cannot be deemed complete before AI and ML components are used. AI and ML solutions can mainly help identify similarities between specific previous attacks and include an automatic warning when any similar danger is identified. The most valuable feature of AI/ML is that it can consistently discern user behavior, changing use patterns, and many other anomalies [3, 329]. One of our testing recommendations, which security experts have agreed to, is to standardize the data packages accessible in order to facilitate the decoding and interpretation of data through machine learning solutions. Our data collection is calculated in exabytes. Upon specifying and optimizing datasets, machine learning algorithms can be beneficial in the protection against cyber threats. We recommend that a fine line be drawn between agreeing on a supervised solution based on features derived from our data collection on the basis of our proposed research solution. While AI and machine learning systems should run independently without human intervention, a small amount of human input should be provided to maintain the system balanced and functional. Although it is limited to creating a hybrid detection model for combating and mitigating IoT cyberattacks in a host and network infrastructure environment, we also suggest using different algorithms such as Eclat and Apriori to warn users of cyberattacks.

9.6. Future Directions of DL. Building modern IoT network architectures with protection protocols including authentication, access control, confidentiality, and intravenous system detection is a successful solution for end-to-end safety. New IoT architectures must prioritize quality of service over efficiency, and they must incorporate evolving paradigms such as SDN and fog-enabled IoT. Optimization algorithms such as genetic algorithms (GAs), bacterial foraging optimization (BFO), particle swarm optimization (PSO), and attribute extraction and selection techniques, as well as parameter tuning, can be used. Hybrid deep learning techniques may be used to increase performance without significantly raising computing time. Blockchain technology, which uses deep learning, may also be used to improve IoT stability. Blockchain technology is a relatively new solution to ensuring the secrecy and security of distributed records.

10. Suggested IoT Security Practices

(i) While past IoT security concerns have been addressed, there are more considerations that need to be made, such as the following suggestions.

(ii) A recognized IoT cybersecurity framework based on industry experience, standards, and proper procedures provided by regulatory bodies should be used.

(iii) IoT devices should not rely only on the network firewall to prevent malicious communication.

(iv) Generate a cybersecurity/IoT incident response strategy and immediately assign the router a name.

(v) Weakness examinations of devices that are linked to remote systems are very important.

(vi) It is good to periodically update the default login credentials and double-check all connected devices.

(vii) IoT systems must be partitioned or isolated to decrease the number of points of attack.

(viii) Threat intelligence must be monitored and shared. In addition, it is critical to scan all software to ensure that the network does not have any security holes.

(ix) In order to digitally fence networks and devices, it is essential that security software be installed and objects and containers are added.

(x) People, businesses, and governments need to keep an eye on and exchange information about threats.

(xi) Other attack detection measures, such as DDoS, IP spoofing, and so on, may be implemented.

(xii) Devices and networks must be updated and patched on a regular basis.

(xiii) Avoid adding devices to the network that use default passwords or have known security flaws.

(xiv) Device apps and controllers need to have their access credentials verified.

(xv) Biometrics and robust validation should be utilized for access control.

(xvi) When linked to a system, use machine validation and IoT messaging encryption, especially for data in transit.

(xvii) In order to protect the LAN from the Internet, firewalls already in use need to be upgraded to more powerful models.

(xviii) If you are using Wi-Fi, be sure you are using a secure router and using passwords that are strong and unique. Wi-Fi security also necessitates the use of high-quality encryption.

(xix) Make use of a variety of security measures, including antivirus.

(xx) Whenever feasible, make a copy of all of your data. Inbound connections to connected devices should be disabled by default.

(xxi) All data should be safeguarded from unwanted access, both while in transit and while they are stored.

(xxii) Devices must be able to delete or reject data storage items with ease. Secure USB ports, for example, should never be used by systems with exposed external interfaces.

(xxiii) It is possible to hire security specialists or to hire cloud security professionals.

(xxiv) Predictive analytics, real-time monitoring, and auditing should be done in the long term.

(xxv) To ensure the safety of all employees and the general public, security awareness training and public exposure to hacker and intruder techniques and tactics are critical.

(xxvi) The IoT should be more restricted to intruders rather than more lenient. Only trustworthy endpoints should be used to connect with IoT items, according to industry standards.

(xxvii) There should be a clear emphasis on eliminating security concerns, such as illegal hacking or operation, environmental risks, tampering, and system malfunctions in IoT systems.

(xxviii) The impact of a security vulnerability on a potential attacker should be limited, such as allowing personal identifying information and assuring rapid discovery and prompt handling of any breaches.

The objective of securing the IoT has yet to be fulfilled, despite different initiatives. IoT security is still a difficult topic to solve. However, the use of cutting-edge artificial intelligence-based cybersecurity systems may considerably deter invaders.

11. Conclusions

Traditional security and privacy strategies have many problems linked to the complexity of IoT networks. DL and ML technology can be used to adjust IoT devices to our real life. The review considered several types of IoT threats. DL and ML are addressed with several potential solutions for ensuring IoT security. A number of DL and ML models are illustrated with their application in IoT security. This review discusses the state-of-the-art solutions for IoT privacy and security utilizing deep learning and machine learning techniques and their integration. While studying machine learning privacy and security issues, we also made an effort to develop a review of IoT threats using previous studies on DL and ML. New issues and insights of ML and DL in IoT security are addressed. Moreover, future direction, security challenges, limitations, and suggestions are included for empowering future technology.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: a survey on enabling technologies, protocols, and applications,” *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] A. Sharma, P. K. Singh, and Y. Kumar, “An integrated fire detection system using IoT and image processing technique for smart cities,” *Sustainable Cities and Society*, vol. 61, Article ID 102332, 2020.

[3] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, “Machine Learning in IoT Security: Current Solutions and Future Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 22, 2020.

[4] F. Hussain, *Internet of Things: Building Blocks and Business Models*, Springer, New York, NY, USA, 2017.

[5] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A survey on internet of things: architecture, enabling technologies, security and privacy, and applications,” *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[6] M. Abomhara and G. M. K�ien, “Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks,” *Journal of Cyber Security and Mobility*, vol. 4, no. 1, pp. 65–88, 2015.

[7] S. Ray, Y. Jin, and A. Raychowdhury, "The changing computing paradigm with internet of things: a tutorial introduction," *IEEE Design & Test*, vol. 33, no. 2, pp. 76–96, 2016.

[8] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical Systems: The Next Computing Revolution," in *Proceedings of the Design Automation Conference*, pp. 731–736, IEEE, Anaheim, CA, USA, June 2010.

[9] E. Bertino and N. Islam, "Botnets and internet of things security," *Computer*, vol. 50, no. 2, pp. 76–79, 2017.

[10] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: real-time intrusion detection in the internet of things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[11] S. Bharati, M. R. H. Mondal, P. Podder, and V. B. Prasath, "Federated learning: applications, challenges and future directions," *International Journal of Hybrid Intelligent Systems*, vol. 18, pp. 1–17, 2022.

[12] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[13] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "Corrauc: A Malicious Bot-Iot Traffic Detection Method in Iot Network Using Machine Learning Techniques," *IEEE Internet of Things Journal*, vol. 8, 2020.

[14] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: deep learning for the internet of things with edge computing," *IEEE network*, vol. 32, no. 1, pp. 96–101, 2018.

[15] E. Fernandes, A. Rahmati, K. Eykholt, and A. Prakash, "Internet of things security research: a rehash of old ideas or new intellectual challenges?" *IEEE Security & Privacy*, vol. 15, no. 4, pp. 79–84, 2017.

[16] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Applied Signal Processing*, vol. 2016, pp. 67–16, 2016.

[17] A. E. Omolara, A. Alabdulatif, O. I. Abiodun et al., "The internet of things security: a survey encompassing unexplored areas and new insights," *Computers & Security*, vol. 112, Article ID 102494, 2022.

[18] S. Yao, Y. Zhao, A. Zhang et al., "Deep learning for the internet of things," *Computer*, vol. 51, no. 5, pp. 32–41, 2018.

[19] A. Riahi Sfar, E. Natalizio, Y. Challal, and Z. Chtourou, "A roadmap for security challenges in the Internet of Things," *Digital Communications and Networks*, vol. 4, no. 2, pp. 118–137, 2018.

[20] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: the road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[21] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: a survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017.

[22] P. Podder, M. R. H. Mondal, S. Bharati, and P. K. Paul, "Review on the security threats of internet of things," *International Journal of Computer Application*, vol. 176, no. 41, pp. 37–45, 2020.

[23] S. Bharati, P. Podder, M. R. H. Mondal, and P. K. Paul, "Applications and challenges of cloud integrated IoMT," in *Cognitive Internet of Medical Things for Smart Healthcare*, pp. 67–85, Springer, New York, NY, USA, 2021.

[24] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of things security: a top-down survey," *Computer Networks*, vol. 141, pp. 199–221, 2018.

[25] X. Wen, "Using deep learning approach and IoT architecture to build the intelligent music recommendation system," *Soft Computing*, vol. 25, no. 4, pp. 3087–3096, 2020.

[26] J. Granjal, E. Monteiro, and J. Sa Silva, "Security for the internet of things: a survey of existing protocols and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[27] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[28] R. H. Weber, "Internet of Things–New security and privacy challenges," *Computer Law & Security Report*, vol. 26, no. 1, pp. 23–30, 2010.

[29] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013.

[30] I. Yaqoob, E. Ahmed, M. H. Rehman et al., "The rise of ransomware and emerging security challenges in the Internet of Things," *Computer Networks*, vol. 129, pp. 444–458, 2017.

[31] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications surveys & tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[32] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 686–728, 2019.

[33] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: algorithms, strategies, and applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[34] A. Oussous, F.-Z. Benjelloun, A. Ait Lahcen, and S. Belfkih, "Big Data technologies: a survey," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 4, pp. 431–448, 2018.

[35] Y. Otoum and A. Nayak, "On Securing IoT from Deep Learning Perspective," in *Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC)*, Rennes, France, July 2020.

[36] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[37] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[38] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," *IEEE Communications Surveys & Tutorials*, vol. 22, 2020.

[39] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: a survey and new perspectives," *ACM Computing Surveys*, vol. 52, no. 1, pp. 1–38, 2020.

[40] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2392–2431, 2017.

[41] R. Li, Z. Zhao, X. Zhou et al., "Intelligent 5G: when cellular networks meet artificial intelligence," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175–183, 2017.

[42] L. Wang and R. Jones, "Big data analytics for network intrusion detection: a survey," *International Journal of Networks and Communications*, vol. 7, no. 1, pp. 24–31, 2017.

[43] K. Ota, M. S. Dao, V. Mezaris, and F. G. B. D. Natale, "Deep learning for mobile multimedia: a survey," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 13, no. 3s, pp. 1–22, 2017.

[44] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[45] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain, "Machine learning for resource management in cellular and IoT networks: potentials, current solutions, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1251–1275, 2020.

[46] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the internet of things: perspectives and challenges," *Wireless Networks*, vol. 20, no. 8, pp. 2481–2501, 2014.

[47] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of threats to the internet of things," *IEEE communications surveys & tutorials*, vol. 21, no. 2, pp. 1636–1675, 2019.

[48] I. Brass, L. Tanczer, M. Carr, M. Elsden, and J. Blackstock, "Standardising a Moving Target: The Development and Evolution of IoT Security Standards," in *Proceedings of the Living in the Internet of Things: Cybersecurity of the IoT - 2018*, London, UK, March 2018.

[49] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the internet of things," *IEEE Access*, vol. 6, Article ID 32979, 2018.

[50] B. Lee and J.-H. Lee, "Blockchain-based secure firmware update for embedded devices in an Internet of Things environment," *The Journal of Supercomputing*, vol. 73, no. 3, pp. 1152–1167, 2017.

[51] I. Butun, P. Österberg, and H. Song, "Security of the internet of things: vulnerabilities, attacks, and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 616–644, 2020.

[52] F. Restuccia, S. D'Oro, and T. Melodia, "Securing the internet of things in the age of machine learning and software-defined networking," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4829–4842, 2018.

[53] L. Xiao, D. Jiang, D. Xu, and N. An, "Secure mobile Crowdsensing with Deep Learning," 2018, https://arxiv.org/abs/1801.07379.

[54] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning: how do IoT devices use AI to enhance security?" *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41–49, 2018.

[55] M. Brewczyńska, S. Dunn, and A. Elijahu, "Data privacy laws response to ransomware attacks: a multi-jurisdictional analysis," in *Regulating New Technologies in Uncertain Times*, Springer, New York, NY, USA, 2019.

[56] P. Prabhu and K. N. Manjunath, "Secured image transmission in medical imaging applications—a survey," in *Computer Aided Intervention and Diagnostics in Clinical and Medical Images*, Springer, New York, NY, USA, 2019.

[57] K. K. F. Yuen, "Towards a Cybersecurity Investment Assessment Method Using Primitive Cognitive Network Process," in *Proceedings of the2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 68–71, IEEE, Okinawa, Japan, 2019.

[58] J. Wang, S. Hu, Q. Wang, and Y. Ma, "Privacy-preserving outsourced feature extractions in the cloud: a survey," *IEEE Network*, vol. 31, no. 5, pp. 36–41, 2017.

[59] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: threats and solutions," *IEEE Security & Privacy*, vol. 17, no. 2, pp. 49–58, 2019.

[60] A. Abeshu and N. Chilamkurti, "Deep learning: the Frontier for distributed attack detection in fog-to-things computing," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 169–175, 2018.

[61] A. Diro and N. Chilamkurti, "Leveraging LSTM networks for attack detection in fog-to-things communications," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 124–130, 2018.

[62] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "Distblocknet: a distributed blockchains-based secure sdn architecture for iot networks," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 78–85, 2017.

[63] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, "A system for denial-of-service attack detection based on multivariate correlation analysis," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 447–456, 2014.

[64] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, "Detection of denial-of-service attacks based on computer vision techniques," *IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2519–2533, 2015.

[65] C. Tselios, I. Politis, and S. Kotsopoulos, "Enhancing SDN Security for IoT-Related Deployments through Blockchain," in *Proceedings of the 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 303–308, IEEE, Forum, Berlin, November 2017.

[66] X. Jing, Z. Yan, X. Jiang, and W. Pedrycz, "Network traffic fusion and analysis against DDoS flooding attacks with a novel reversible sketch," *Information Fusion*, vol. 51, pp. 100–113, 2019.

[67] O. E. Elejla, B. Belaton, M. Anbar, B. Alabsi, and A. K. Al-Ani, "Comparison of classification algorithms on ICMPv6-based DDoS attacks detection," in *Computational Science and Technology*Springer, New York, NY, USA, 2019.

[68] M. Rezazad, M. R. Brust, M. Akbari, P. Bouvry, and N.-M. Cheung, "Detecting Target-Area Link-Flooding Ddos Attacks Using Traffic Analysis and Supervised Learning," *Advances in Intelligent Systems and Computing*, Springer, New York, NY, USA, 2018.

[69] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2019.

[70] B. Z. H. Zhao, M. Ikram, H. J. Asghar, M. A. Kaafar, A. Chaabane, and K. Thilakarathna, "A Decade of Mal-Activity Reporting: A Retrospective Analysis of Internet Malicious Activity Blacklists," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pp. 193–205, Auckland, New Zealand, July 2019.

[71] A. Azmoodeh, A. Dehghantanha, and K.-K. R. Choo, "Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning," *IEEE Transactions on Sustainable Computing*, vol. 4, no. 1, pp. 88–95, 2019.

[72] S. Aonzo, A. Merlo, M. Migliardi, L. Oneto, and F. Palmieri, "Low-resource footprint, data-driven malware detection on android," *IEEE Transactions on Sustainable Computing*, vol. 5, no. 2, pp. 213–222, 2020.

[73] P. Feng, J. Ma, C. Sun, X. Xu, and Y. Ma, "A novel dynamic Android malware detection system with ensemble learning," *IEEE Access*, vol. 6, Article ID 30996, 2018.

[74] L. Wei, W. Luo, J. Weng, Y. Zhong, X. Zhang, and Z. Yan, "Machine learning-based malicious application detection of android," *IEEE Access*, vol. 5, Article ID 25591, 2017.

[75] J. Gu, B. Sun, X. Du, J. Wang, Y. Zhuang, and Z. Wang, "Consortium blockchain-based malware detection in mobile devices," *IEEE Access*, vol. 6, Article ID 12118, 2018.

[76] S. Sharmeen, S. Huda, J. H. Abawajy, W. N. Ismail, and M. M. Hassan, "Malware threats and detection for industrial mobile-IoT networks," *IEEE Access*, vol. 6, Article ID 15941, 2018.

[77] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2027–2051, 2016.

[78] U. Ahmad, H. Song, A. Bilal, S. Saleem, and A. Ullah, "Securing Insulin Pump System Using Deep Learning and Gesture Recognition," in *Proceedings of the 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, New York, NY, USA, August 2018.

[79] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Deep abstraction and weighted feature selection for Wi-Fi impersonation detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 621–636, 2018.

[80] B. Chatterjee, D. Das, and S. Sen, "RF-PUF: IoT Security Enhancement through Authentication of Wireless Nodes Using In-Situ Machine Learning," in *Proceedings of the 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 205–208, IEEE, Washington, DC, USA, April 2018.

[81] N. Wang, T. Jiang, S. Lv, and L. Xiao, "Physical-layer authentication based on extreme learning machine," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1557–1560, 2017.

[82] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of things (IoT): taxonomy of security attacks," in *Proceedings of the 2016 3rd International Conference on Electronic Design (ICED)*, pp. 321–326, IEEE, Phuket, Thailand, 2016.

[83] A. Banerjee, K. K. Venkatasubramanian, T. Mukherjee, and S. K. S. Gupta, "Ensuring safety, security, and sustainability of mission-critical cyber–physical systems," in *Proceedings of the IEEE*, vol. 100, no. 1, pp. 283–299, 2012.

[84] K. Wan and V. Alagar, "Context-aware security solutions for cyber-physical systems," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 212–226, 2014.

[85] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, "Security and privacy for cloud-based IoT: challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26–33, 2017.

[86] R. AlTawy and A. M. Youssef, "Security tradeoffs in cyber physical systems: a case study survey on implantable medical devices," *IEEE Access*, vol. 4, pp. 959–979, 2016.

[87] S. Fosso Wamba, A. Anand, and L. Carter, "A literature review of RFID-enabled healthcare applications and issues," *International Journal of Information Management*, vol. 33, no. 5, pp. 875–891, 2013.

[88] K. Malasri and L. Wang, "Securing wireless implantable devices for healthcare: ideas and challenges," *IEEE Communications Magazine*, vol. 47, no. 7, pp. 74–80, 2009.

[89] A. C. Jose and R. Malekian, "Improving smart home security: integrating logical sensing into smart home," *IEEE Sensors Journal*, vol. 17, no. 13, pp. 4269–4286, 2017.

[90] A. Gharaibeh, M. A. Salahuddin, S. J. Hussini et al., "Smart cities: a survey on data management, security, and enabling technologies," *IEEE Commun. Surveys Tuts.*vol. 19, no. 4, pp. 2456–2501, 2017.

[91] D. Eckhoff and I. Wagner, "Privacy in the smart city—applications, technologies, challenges, and solutions," *IEEE Commun. Surveys Tuts.*vol. 20, no. 1, pp. 489–516, 2018.

[92] V. Namboodiri, V. Aravinthan, S. N. Mohapatra, B. Karimi, and W. Jewell, "Toward a secure wireless-based home area network for metering in smart grids," *IEEE Systems Journal*, vol. 8, no. 2, pp. 509–520, Jun. 2014.

[93] B.. IoT, "System | sensors and actuators," 2022, https://bridgera.com/sensors-and-actuators-in-iot/.

[94] S. Kumar, S. Sahoo, A. Mahapatra, A. K. Swain, and K. K. Mahapatra, "Security enhancements to system on chip devices for IoT perception layer," in *Proceedings of the IEEE Int. Symp. Nanoelectron. Inf. Syst. (iNIS)*, pp. 151–156, Bhopal, India, December 2017.

[95] A. P. WG. Phishing, "Activity trends report," 2022, https://docs.apwg.org/reports/apwg_trends_report_q4_2017.pdf.

[96] S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta, "A survey of middleware for Internet of Things," in *Recent Trends in Wireless and Mobile Networks*Springer, Berlin, Germany, 2011.

[97] Q. Zhang and X. Wang, "SQL injections through back-end of RFID system," in *Proceedings of the 2009 International Symposium on Computer Network and Multimedia Technology*, pp. 1–4, Wuhan, China, January 2009.

[98] R. Dorai and V. Kannan, "SQL injection-database attack revolution and prevention," *Journal of International Commercial Law & Technology*, vol. 6, no. 4, p. 224, 2011.

[99] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for internet of things: a survey," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70–95, Feb. 2016.

[100] A. Stanciu, T.-C. Balan, C. Gerigan, and S. Zamfir, "Securing the IoT gateway based on the hardware implementation of a multi pattern search algorithm," in *Proceedings of the 2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP)*, pp. 1001–1006, Brasov, Romania, May 2017.

[101] H. A. Abdul-Ghani, D. Konstantas, and M. Mahyoub, "'A comprehensive IoT attacks survey based on a building-blocked reference model," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, pp. 355–373, 2018.

[102] G. D'Agostini, "A multidimensional unfolding method based on Bayes' theorem," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 362, no. 2-3, pp. 487–498, 1995.

[103] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Computer Science*, vol. 60, pp. 708–713, 2015.

[104] M. Swarnkar and N. Hubballi, "OCPAD: one class Naive Bayes classifier for payload based anomaly detection," *Expert Systems with Applications*, vol. 64, pp. 330–339, 2016.

[105] M. Panda and M. R. Patra, "Network intrusion detection using naive bayes," *International journal of computer science and network security*, vol. 7, no. 12, pp. 258–263, 2007.

[106] S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," *Procedia Technology*, vol. 4, pp. 119–128, 2012.

[107] G. E. P. Box and G. C. Tiao, *Bayesian Inference in Statistical Analysis*, John Wiley & Sons, Hoboken, NJ, USA, 2011.

[108] A. Y. Ng and M. I. Jordan, "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes," in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, Vancouver, BC, Canada, January 2002.

[109] B. Zhang, Z. Liu, Y. Jia, J. Ren, and X. Zhao, "Network intrusion detection method based on PCA and Bayes

algorithm," *Security and Communication Networks*, Article ID 1914980, 11 pages, 2018.

[110] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer science & business media, New York, NY, USA, 2013.

[111] M. Mohammadi, T. A. Rashid, S. H. Karim et al., "A comprehensive survey and taxonomy of the SVM-based intrusion detection systems," *Journal of Network and Computer Applications*, vol. 178, Article ID 102983, 2021.

[112] H. Karimipour and V. Dinavahi, "On False Data Injection Attack against Dynamic State Estimation on Smart Power Grids," in *Proceedings of the 2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, pp. 388–393, IEEE, Oshawa, ON, Canada, August 2017.

[113] H.-S. Ham, H.-H. Kim, M.-S. Kim, and M.-J. Choi, "Linear SVM-based android malware detection for reliable IoT services," *Journal of Applied Mathematics*, pp. 1–10, 2014.

[114] Y. Liu and D. Pi, "A novel kernel SVM algorithm with game theory for network intrusion detection," *KSII Transactions on Internet & Information Systems*, vol. 11, no. 8, 2017.

[115] X. Luo, "Efficient English text classification using selected machine learning techniques," *Alexandria Engineering Journal*, vol. 60, no. 3, pp. 3401–3409, 2021.

[116] U. Naseem, M. Khushi, S. K. Khan, K. Shaukat, and M. A. Moni, "A comparative analysis of active learning for biomedical text mining," *Applied System Innovation*, vol. 4, no. 1, p. 23, 2021.

[117] S. Wambura, J. Huang, and H. Li, "Robust anomaly detection in feature-evolving time series," *The Computer Journal*, vol. 65, no. 5, pp. 1242–1256, 2021.

[118] H. Neuschmied, M. Winter, K. Hofer-Schmitz, B. Stojanovic, and U. Kleb, "Two stage anomaly detection for network intrusion detection," in *Proceedings of the ICISSP*, 2021.

[119] W. Hu, Y. Liao, and V. R. Vemuri, "Robust Support Vector Machines for Anomaly Detection in Computer Security," in *Proceedings of the 2003 International Conference on Machine Learning and Applications - ICMLA 2003*, pp. 168–174, Los Angeles, California, USA, June 2003.

[120] C. Wagner, J. François, and T. Engel, *Machine Learning Approach for Ip-Flow Record Anomaly Detection*, Springer, New York, NY, USA, 2011.

[121] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1773–1786, 2016.

[122] L. Lerman, G. Bontempi, and O. Markowitch, "A machine learning approach against a masked AES," *Journal of Cryptographic Engineering*, vol. 5, no. 2, pp. 123–139, 2015.

[123] A. Heuser and M. Zohner, *Intelligent Machine Homicide*, Springer, New York, NY, USA, 2012.

[124] M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, and H. Janicke, "Rdtids: rules and decision tree-based intrusion detection system for internet-of-things networks," *Future Internet*, vol. 12, no. 3, p. 44, 2020.

[125] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine," *Electronics*, vol. 9, no. 1, p. 173, 2020.

[126] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: a review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[127] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, "Explainable artificial intelligence (xai) to enhance trust management in intrusion detection systems using decision tree model," *Complexity*, vol. 2021, Article ID 6634811, 11 pages, 2021.

[128] W. Du and Z. Zhan, "Building Decision Tree Classifier on Private Data," in *Proceedings of the IEEE international conference on Privacy, security and data mining*, Maebashi City, Japan, 2002.

[129] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[130] S. B. Kotsiantis, "Decision trees: a recent overview," *Artificial Intelligence Review*, vol. 39, no. 4, pp. 261–283, 2013.

[131] S. K. Murthy, "Automatic construction of decision trees from data: a multi-disciplinary survey," *Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 345–389, 1998.

[132] K. Goeschel, "Reducing False Positives in Intrusion Detection Systems Using Data-Mining Techniques Utilizing Support Vector Machines, Decision Trees, and Naive Bayes for Off-Line Analysis," in *Proceedings of the SoutheastCon 2016*, pp. 1–6, IEEE, Norfolk, VA, USA, March 2016.

[133] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2014.

[134] S. Alharbi, P. Rodriguez, R. Maharaja, P. Iyer, N. Subaschandrabose, and Z. Ye, "Secure the Internet of Things with challenge Response Authentication in Fog Computing," in *Proceedings of the 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, pp. 1-2, IEEE, San Diego, CA, USA, 2017.

[135] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[136] D. R. Cutler, T. C. Edwards, K. H. Beard et al., "Random forests for classification in ecology," *Ecology*, vol. 88, no. 11, pp. 2783–2792, 2007.

[137] J. Zhang and M. Zulkernine, "A Hybrid Network Intrusion Detection Technique Using Random Forests," in *Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06)*, p. 8, IEEE, Vienna, Austria, 2006.

[138] I. H. Sarker, "CyberLearning: effectiveness analysis of machine learning security modeling to detect cyber-anomalies and multi-attacks," *Internet of Things*, vol. 14, Article ID 100393, 2021.

[139] Y. Chen, W. Zheng, W. Li, and Y. Huang, "Large group Activity security risk assessment and risk early warning based on random forest algorithm," *Pattern Recognition Letters*, vol. 144, pp. 1–5, 2021.

[140] H. Zhang, J.-L. Li, X.-M. Liu, and C. Dong, "Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection," *Future Generation Computer Systems*, vol. 122, pp. 130–143, 2021.

[141] Y. Chang, W. Li, and Z. Yang, "Network Intrusion Detection Based on Random forest and Support Vector Machine," in *Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Guangzhou, China, 2017.

[142] R. Doshi, N. Apthorpe, and N. Feamster, "Machine Learning Ddos Detection for Consumer Internet of Things Devices," in *Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW)*, pp. 29–35, IEEE, San Francisco, CA, USA, May 2018.

[143] Y. Meidan, B. Michael, S. Asaf et al., "Detection of unauthorized IoT devices using machine learning techniques," 2017, https://arxiv.org/abs/1709.04647.

[144] P. Soucy and G. W. Mineau, "A Simple KNN Algorithm for Text Categorization," in *Proceedings of the 2001 IEEE International Conference on Data Mining*, pp. 647-648, IEEE, San Francisco, CA, USA, 2001.

[145] K. K. Sharma and A. Seal, "Spectral embedded generalized mean based k-nearest neighbors clustering with S-distance," *Expert Systems with Applications*, vol. 169, Article ID 114326, 2021.

[146] M. Alsharif and D. B. Rawat, "Study of machine learning for cloud assisted iot security as a service," *Sensors*, vol. 21, no. 4, p. 1034, 2021.

[147] S. Meera and C. Sundar, "Retracted article: a hybrid metaheuristic approach for efficient feature selection methods in big data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 3743–3751, 2021.

[148] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient kNN classification algorithm for big data," *Neurocomputing*, vol. 195, pp. 143–148, 2016.

[149] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.

[150] A. O. Adetunmbi, S. O. Falaki, O. S. Adewale, and B. K. Alese, "Network intrusion detection based on rough set and k-nearest neighbour," *International Journal of Computational Intelligence Research*, vol. 2, no. 1, pp. 60–66, 2008.

[151] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: a review," *Expert Systems with Applications*, vol. 36, no. 10, Article ID 11994, 2009.

[152] L. Li, H. Zhang, H. Peng, and Y. Yang, "Nearest neighbors based density peaks approach to intrusion detection," *Chaos, Solitons & Fractals*, vol. 110, pp. 33–40, 2018.

[153] A. R. Syarif and W. Gata, "Intrusion Detection System Using Hybrid Binary PSO and K-Nearest Neighborhood Algorithm," in *Proceedings of the 2017 11th International Conference on Information & Communication Technology and System (ICTS)*, pp. 181–186, IEEE, Surabaya, Indonesia, 2017.

[154] M.-Y. Su, "Real-time anomaly detection systems for Denial-of-Service attacks by weighted k-nearest-neighbor classifiers," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3492–3498, 2011.

[155] R. Wazirali, "An improved intrusion detection system based on KNN hyperparameter tuning and cross-validation," *Arabian Journal for Science and Engineering*, vol. 45, no. 12, Article ID 10859, 2020.

[156] H. Xu, K. Przystupa, C. Fang, A. Marciniak, O. Kochan, and M. Beshley, "A combination strategy of feature selection based on an integrated optimization algorithm and weighted K-nearest neighbor to improve the performance of network intrusion detection," *Electronics*, vol. 9, no. 8, p. 1206, 2020.

[157] A. A. R. Melvin, G. J. W. Kathrine, S. S. Ilango et al., "Dynamic malware attack dataset leveraging virtual machine monitor audit data for the detection of intrusions in cloud," *Transactions on Emerging Telecommunications Technologies*, vol. 33, Article ID e4287, 2021.

[158] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K.-K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 2, pp. 314–323, 2019.

[159] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, "A new intrusion detection system based on KNN classification algorithm in wireless sensor network," *Journal of Electrical and Computer Engineering*, pp. 1–8, 2014.

[160] M. Woźniak, M. Grana, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, pp. 3–17, 2014.

[161] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[162] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*, Springer, New York, NY, USA, 2012.

[163] L. E. A. Santana, L. Silva, A. M. P. Canuto, F. Pintro, and K. O. Vale, "A Comparative Analysis of Genetic Algorithm and Ant colony Optimization to Select Attributes for an Heterogeneous Ensemble of Classifiers," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1–8, IEEE, Barcelona, Spain, July 2010.

[164] N. M. Baba, M. Makhtar, S. A. Fadzli, and M. K. Awang, "Current issues in ensemble methods and its applications," *Journal of Theoretical and Applied Information Technology*, vol. 81, no. 2, p. 266, 2015.

[165] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for IoT applications," *Wireless Personal Communications*, vol. 111, no. 4, pp. 2287–2310, 2020.

[166] I. Cvitić, D. Peraković, M. Periša, and B. Gupta, "Ensemble machine learning approach for classification of IoT devices in smart home," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 11, pp. 3179–3202, 2021.

[167] D. P. Gaikwad and R. C. Thool, "Intrusion Detection System Using Bagging Ensemble Method of Machine Learning," in *Proceedings of the 2015 International Conference on Computing Communication Control and Automation*, pp. 291–295, IEEE, Pune, India, 2015.

[168] A. A. Aburomman and M. B. Ibne Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Applied Soft Computing*, vol. 38, pp. 360–372, 2016.

[169] R. R. Reddy, Y. Ramadevi, and K. V. N. Sunitha, "Enhanced Anomaly Detection Using Ensemble Support Vector Machine," in *Proceedings of the 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, pp. 107–111, IEEE, Chirala, AP, India, 2017.

[170] S. Y. Yerima, S. Sezer, and I. Muttik, "High accuracy android malware detection using ensemble learning," *IET Information Security*, vol. 9, no. 6, pp. 313–320, 2015.

[171] R. Agrawal, T. Imieliński, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Washington, D.C., USA, 1993.

[172] H. Brahmi, I. Brahmi, and S. B. Yahia, "OMC-IDS: At the Cross-Roads of OLAP Mining and Intrusion Detection," *Advances in Knowledge Discovery and Data Mining*, Springer, Berlin, Germany, 2012.

[173] A. Tajbakhsh, M. Rahmati, and A. Mirzaei, "Intrusion detection using fuzzy association rules," *Applied Soft Computing*, vol. 9, no. 2, pp. 462–469, 2009.

[174] S. Kotsiantis and D. Kanellopoulos, "Association rules mining: a recent overview," *GESTS International Transactions on Computer Science and Engineering*, vol. 32, no. 1, pp. 71–82, 2006.

[175] H. H. W. J. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, and A. Liotta, "Ensembles of incremental learners to detect

anomalies in ad hoc sensor networks," *Ad Hoc Networks*, vol. 35, pp. 14–36, 2015.

[176] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[177] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[178] R. Bro and A. K. Smilde, "Principal component analysis," *Analytical Methods*, vol. 6, no. 9, pp. 2812–2831, 2014.

[179] S. Bhattacharya, S. R. K. S, P. K. R. Maddikunta et al., "A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU," *Electronics*, vol. 9, no. 2, p. 219, 2020.

[180] S. P. Rm, P. K. R. Maddikunta, S. Koppu, T. R. Gadekallu, C. L. Chowdhary, and M. Alazab, "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," *Computer Communications*, vol. 160, pp. 139–149, 2020.

[181] S. Zhao, W. Li, T. Zia, and A. Y. Zomaya, "A Dimension Reduction Model and Classifier for Anomaly-Based Intrusion Detection in Internet of Things," in *Proceedings of the 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pp. 836–843, IEEE, Orlando, FL, USA, 2017.

[182] J. A. Hartigan and M. A. Wong, "Algorithm as 136: a k-means clustering algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979.

[183] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[184] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," *GI/ITG Workshop MMBnet*, vol. 7, pp. 13-14, 2007.

[185] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *Ieee communications surveys & tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[186] A. P. Muniyandi, R. Rajeswari, and R. Rajaram, "Network anomaly detection by cascading k-Means clustering and C4. 5 decision tree algorithm," *Procedia Engineering*, vol. 30, pp. 174–182, 2012.

[187] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning Intrusion Detection: Supervised or Unsupervised?" in *Proceedings of the Image Analysis and Processing – ICIAP 2005*, Springer, Berlin, Germany, 2005.

[188] H.-b. Wang, Z. Yuan, and C.-d. Wang, "Intrusion detection for wireless sensor networks based on multi-agent and refined clustering,"vol. 3, pp. 450–454, in *Proceedings of the 2009 WRI International Conference on Communications and Mobile Computing*, vol. 3, IEEE, Kunming, China, January 2009.

[189] Q. Li, K. Zhang, M. Cheffena, and X. Shen, "Channel-based Sybil Detection in Industrial Wireless Sensor Networks: A Multi-Kernel Approach," in *Proceedings of the GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, IEEE, Singapore, 2017.

[190] M. Xie, M. Huang, Y. Bai, and Z. Hu, "The anonymization protection algorithm based on fuzzy clustering for the ego of data in the internet of things," *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1–10, 2017.

[191] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised Learning Using Gaussian fields and Harmonic Functions," in *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, pp. 912–919, Washington, DC, USA, August 2003.

[192] X. J. Zhu, "Semi-supervised Learning Literature Survey," Technical Report. 1530, University of Wisconsin-Madison, Madison, WI, USA, 2005.

[193] O. Y. Al-Jarrah, Y. Al-Hammdi, P. D. Yoo, S. Muhaidat, and M. Al-Qutayri, "Semi-supervised multi-layered clustering model for intrusion detection," *Digital Communications and Networks*, vol. 4, no. 4, pp. 277–286, 2018.

[194] S. Rathore and J. H. Park, "Semi-supervised learning based distributed attack detection framework for IoT," *Applied Soft Computing*, vol. 72, pp. 79–89, 2018.

[195] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[196] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, USA, 2011.

[197] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: a brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[198] M. A. Aref, S. K. Jayaweera, and S. Machuzak, "Multi-agent Reinforcement Learning Based Cognitive Anti-jamming," in *Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, Washington, DC, USA, March 2017.

[199] S. Machuzak and S. K. Jayaweera, "Reinforcement Learning Based Anti-jamming with Wideband Autonomous Cognitive Radios," in *Proceedings of the 2016 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 1–5, IEEE, Chengdu, China, July 2016.

[200] Y. Gwon, S. Dastangoo, C. Fossa, and H. T. Kung, "Competing mobile Network Game: Embracing Antijamming and Jamming Strategies with Reinforcement Learning," in *Proceedings of the 2013 IEEE Conference on Communications and Network Security (CNS)*, pp. 28–36, IEEE, National Harbor, MD, USA, 2013.

[201] S. Mukkamala, A. H. Sung, A. Abraham, and V. Ramos, "Intrusion detection systems using adaptive regression spines," in *Enterprise Information Systems VI*, Springer, Dordrecht, Netherland, 2006.

[202] A. O. Prokofiev, Y. S. Smirnova, and V. A. Surov, "A Method to Detect Internet of Things Botnets," in *Proceedings of the 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 105–108, IEEE, Moscow and St. Petersburg, Russia, January 2018.

[203] I. Hafeez, A. Y. Ding, M. Antikainen, and S. Tarkoma, "Toward secure edge networks taming device to device (D2D) communication in IoT," 2017, https://arxiv.org/abs/1712.05958.

[204] S. Shadroo, A. M. Rahmani, and A. Rezaee, "The two-phase scheduling based on deep learning in the Internet of Things," *Computer Networks*, vol. 185, Article ID 107684, 2021.

[205] M. A. Rahman and M. S. Hossain, "An Internet of Medical Things-Enabled Edge Computing Framework for Tackling COVID-19," *IEEE Internet of Things Journal*, vol. 8, 2021.

[206] F. Liang, W. Yu, X. Liu, D. Griffith, and N. Golmie, "Toward edge-based deep learning in industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4329–4341, 2020.

[207] Z. M. Fadlullah, F. Tang, B. Mao et al., "State-of-the-art deep learning: evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[208] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[209] S. Bharati, P. Podder, and M. R. H. Mondal, "Hybrid deep learning for detecting lung diseases from X-ray images," *Informatics in Medicine Unlocked*, vol. 20, Article ID 100391, 2020.

[210] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, MIT press, Cambridge, MA, USA, 2016.

[211] S. Tofigh, M. O. Ahmad, and M. N. S. Swamy, "A low-complexity modified ThiNet algorithm for pruning convolutional neural networks," *IEEE Signal Processing Letters*, vol. 29, pp. 1012–1016, 2022.

[212] X.-W. Chen and X. Lin, "Big data deep learning: challenges and perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014.

[213] D. Scherer, A. Müller, and S. Behnke, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," *Artificial Neural Networks – ICANN 2010*, pp. 92–101, Springer, Berlin, Germany, 2010.

[214] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, High Performance Convolutional Neural Networks for Image Classification," in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two*, Barcelona, Catalonia, Spain, 2011.

[215] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," 2015, https://arxiv.org/abs/1511.07289.

[216] E. De Coninck, V. Tim, V. Bert et al., "Distributed Neural Networks for Internet of Things: The Big-Little Approach," *Internet of Things. IoT Infrastructures. IoT360 2015*, pp. 484–492, Springer, New York, NY, USA, 2015.

[217] 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.

[218] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: a technical tutorial on the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, 2016.

[219] N. McLaughlin, R. Jesus Martinez del, K. BooJoong et al., "Deep android malware detection," in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, pp. 301–308, Scottsdale, Arizona, USA, March 2017.

[220] H. Maghrebi, T. Portigliatti, and E. Prouff, "Breaking Cryptographic Implementations Using Deep Learning Techniques," *Security, Privacy, and Applied Cryptography Engineering. SPACE 2016*, pp. 3–26, Springer, New York, NY, USA, 2016.

[221] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," 2013, https://arxiv.org/abs/1312.6026#:~:text=By%20carefully%20analyzing%20and%20understanding,hidden%2Dto%2Doutput%20function.

[222] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," *Advances in Neural Information Processing Systems*, vol. 26, pp. 190–198, 2013.

[223] A. Esmaeilzehi, M. O. Ahmad, and M. N. S. Swamy, "UPDResNN: a deep light-weight image upsampling and deblurring residual neural network," *IEEE Transactions on Broadcasting*, vol. 67, no. 2, pp. 538–548, 2021.

[224] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: state of the art and research challenges," *Expert Systems with Applications*, vol. 105, pp. 233–261, 2018.

[225] R. Pascanu, T. Mikolov, and Y. Bengio, *On the Difficulty of Training Recurrent Neural Networks*, PMLR, Venue, Austria, 2013.

[226] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649, IEEE, Vancouver, BC, Canada, May 2013.

[227] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Information Fusion*, vol. 42, pp. 146–157, 2018.

[228] K. Cho, M. Bart van, G. Caglar et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[229] P. Torres, C. Catania, S. Garcia, and C. G. Garino, "An Analysis of Recurrent Neural Networks for Botnet Detection Behavior," in *Proceedings of the 2016 IEEE Biennial Congress of Argentina (ARGENCON)*, pp. 1–6, IEEE, Buenos Aires, Argentina, 2016.

[230] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*, Springer, Berlin, Germany, 2012.

[231] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted Boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, 2013.

[232] 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.

[233] Y. Chen, Y. Zhang, S. Maharjan, M. Alam, and T. Wu, "Deep learning for secure mobile edge computing in cyber-physical transportation systems," *IEEE Network*, vol. 33, no. 4, pp. 36–41, 2019.

[234] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *International Journal of Security and Its Applications*, vol. 9, no. 5, pp. 205–216, 2015.

[235] S. Harush, Y. Meidan, and A. Shabtai, "DeepStream: autoencoder-based stream temporal clustering and anomaly detection," *Computers & Security*, vol. 106, Article ID 102276, 2021.

[236] M. Yousefi-Azar, V. Varadharajan, L. Hamey, and U. Tupakula, "Autoencoder-based Feature Learning for Cyber Security Applications," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 3854–3861, IEEE, Anchorage, AK, USA, 2017.

[237] I. Goodfellow, J. Pouget-Abadie, M. Mehdi et al., "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, Montreal, Canada, 2014.

[238] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, https://arxiv.org/abs/1411.1784.

[239] R. E. Hiromoto, M. Haney, and A. Vakanski, "A secure architecture for IoT with supply chain risk management,"vol. 1, pp. 431–435, in *Proceedings of the 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 1, IEEE, Bucharest, Romania, 2017.

[240] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proceedings of the 30th International Conference*

on Neural Information Processing Systems, Barcelona, Spain, December 2016.

[241] A. Mohebbi, H. Abdzadeh-Ziabari, W.-P. Zhu, and M. O. Ahmad, "Doubly selective channel estimation algorithms for millimeter wave hybrid MIMO systems," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, Article ID 12821, 2021.

[242] A. Esmaeilzehi, M. O. Ahmad, and M. N. S. Swamy, "SRNHARB: a deep light-weight image super resolution network using hybrid activation residual blocks," *Signal Processing: Image Communication*, vol. 99, Article ID 116509, 2021.

[243] M. R. H. Mondal, S. Bharati, and P. Podder, "CO-IRv2: optimized InceptionResNetV2 for COVID-19 detection from chest CT images," *PLoS One*, vol. 16, no. 10, Article ID e0259179, 2021.

[244] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1184–1199, 2011.

[245] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[246] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning, PMLR*, pp. 1058–1066, PMLR, Venue, Austria, 2013.

[247] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *European Conference on Computer Vision*, Springer, New York, NY, USA, 2016.

[248] 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.

[249] S. Singh, D. Hoiem, and D. Forsyth, "Swapout: learning an ensemble of deep architectures," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[250] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, John Wiley & Sons, Hoboken, NJ, USA, 2014.

[251] M. E. Aminanto and K. Kim, "Detecting Active Attacks in Wi-Fi Network by Semi-supervised Deep Learning," *Conference on Information Security and Cryptography*, 2017.

[252] I.-S. Comşa, S. Zhang, M. E. Aydin et al., "Towards 5G: a reinforcement learning-based scheduling solution for data traffic management," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1661–1675, 2018.

[253] F. Hussain, A. Anpalagan, A. S. Khwaja, and M. Naeem, "Resource allocation and congestion control in clustered M2M communication using Q-learning," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, p. e3039, 2017.

[254] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Secure Computation Offloading in Blockchain Based IoT Networks with Deep Reinforcement Learning," 2019, https://arxiv.org/abs/1908.07466.

[255] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Phoenix, Arizona, February 2016.

[256] T. P. Lillicrap, J. H. Jonathan, P. Alexander et al., "Continuous control with deep reinforcement learning," 2015, https://arxiv.org/abs/1509.02971.

[257] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2015, https://arxiv.org/abs/1511.05952?context=cs.

[258] T. T. Nguyen and V. J. Reddi, "Deep Reinforcement Learning for Cyber Security," 2019, https://arxiv.org/abs/1906.05799.

[259] A. Ferdowsi and W. Saad, "Deep Learning-Based Dynamic Watermarking for Secure Signal Authentication in the Internet of Things," in *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Kansas City, MO, USA, May 2018.

[260] A. Ferdowsi and W. Saad, "Deep learning for signal authentication and security in massive internet-of-things systems," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1371–1387, 2019.

[261] R. Das, A. Gadre, S. Zhang, S. Kumar, and J. M. F. Moura, "A Deep Learning Approach to IoT Authentication," in *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Kansas City, MO, USA, May 2018.

[262] B. A. Tama and K.-H. Rhee, "Attack classification analysis of IoT network via deep learning approach," *Res. Briefs Inf. Commun. Technol. Evol.(ReBICTE)*, vol. 3, pp. 1–9, 2017.

[263] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2018.

[264] O. Brun, Y. Yin, E. Gelenbe, Y. M. Kadioglu, J. Augusto-Gonzalez, and M. Ramos, "Deep Learning with Dense Random Neural Networks for Detecting Attacks against Iot-Connected home Environments," *Communications in Computer and Information Science*, Springer Cham, New York, NY, USA, 2018.

[265] P. Mohamed Shakeel, S. Baskar, V. R. Sarma Dhulipala, S. Mishra, and M. M. Jaber, "Maintaining security and privacy in health care system using learning based deep-Q-networks," *Journal of Medical Systems*, vol. 42, no. 10, pp. 1–10, 2018.

[266] Y. Meidan, M. Bohadana, Y. Mathov et al., "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[267] C. D. McDermott, F. Majdani, and A. V. Petrovski, "Botnet Detection in the Internet of Things Using Deep Learning Approaches," in *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Rio de Janeiro, Brazil, July 2018.

[268] M. Al-Hawawreh, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," *Journal of Information Security and Applications*, vol. 41, pp. 1–11, 2018.

[269] F. Y. Yavuz, D. Ünal, and E. Gül, "Deep learning for detection of routing attacks in the internet of things," *International Journal of Computational Intelligence Systems*, vol. 12, no. 1, pp. 39–58, 2018.

[270] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. M. Leung, "A survey on security threats and defensive techniques of machine learning: a data driven view," *IEEE Access*, vol. 6, Article ID 12103, 2018.

[271] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," *Eai Endorsed Transactions on Security and Safety*, vol. 3, no. 9, p. e2, 2016.

[272] W. Wang, Z. Gao, M. Zhao, Y. Li, J. Liu, and X. Zhang, "DroidEnsemble: detecting Android malicious applications with ensemble of string and structural static features," *IEEE Access*, vol. 6, Article ID 31798, 2018.

[273] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 184–208, 2016.

[274] Smart Phone, "Number of smartphone users worldwide from 2014 to 2020 (in billions)," 2018, https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

[275] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2986–2998, 2016.

[276] L. Fernandez Maimo, A. L. Perales Gomez, F. J. Garcia Clemente, M. Gil Perez, and G. Martinez Perez, "A self-adaptive deep learning-based system for anomaly detection in 5G networks," *IEEE Access*, vol. 6, pp. 7700–7712, 2018.

[277] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, MA, USA, 1988.

[278] Y. Zhou, M. Han, L. Liu, J. S. He, and Y. Wang, "Deep Learning Approach for Cyberattack Detection," in *Proceedings of the IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Honolulu, HI, USA, April 2018.

[279] S. Prabavathy, K. Sundarakantham, and S. M. Shalinie, "Design of cognitive fog computing for intrusion detection in Internet of Things," *Journal of Communications and Networks*, vol. 20, no. 3, pp. 291–298, 2018.

[280] H. Aksu, A. S. Uluagac, and E. Bentley, "Identification of Wearable Devices with Bluetooth," *IEEE Transactions on Sustainable Computing*, vol. 6, 2018.

[281] B. Feng, Q. Fu, M. Dong, D. Guo, and Q. Li, "Multistage and elastic spam detection in mobile social networks through deep learning," *IEEE Network*, vol. 32, no. 4, pp. 15–21, 2018.

[282] Q. Jia, L. Guo, Z. Jin, and Y. Fang, "Preserving model privacy for machine learning in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 8, pp. 1808–1822, 2018.

[283] X. Ma, J. Ma, H. Li, Q. Jiang, and S. Gao, "PDLM: Privacy-Preserving Deep Learning Model on Cloud with Multiple Keys," *IEEE Transactions on Services Computing*, vol. 14, 2018.

[284] L. Xiao, Y. Li, G. Han, G. Liu, and W. Zhuang, "PHY-layer spoofing detection with reinforcement learning in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, Article ID 10037, 2016.

[285] H. Zhu, X. Liu, R. Lu, and H. Li, "Efficient and privacy-preserving online medical prediagnosis framework using nonlinear SVM," *IEEE journal of biomedical and health informatics*, vol. 21, no. 3, pp. 838–850, 2017.

[286] T. Zhang and Q. Zhu, "Distributed privacy-preserving collaborative intrusion detection systems for VANETs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 148–161, 2018.

[287] J. Porras, J. Khakurel, A. Knutas, and J. Pänkäläinen, "Security challenges and solutions in the internet of things," *Nordic and Baltic Journal of Information and Communications Technologies*, vol. 2018, pp. 177–206, 2018.

[288] J. Wurm, K. Hoang, O. Arias, A.-R. Sadeghi, and Y. Jin, "Security analysis on consumer and industrial IoT devices," in *Proceedings of the 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 519–524, IEEE, Macao, China, 2016.

[289] B. Rashid and M. H. Rehmani, "Applications of wireless sensor networks for urban areas: a survey," *Journal of Network and Computer Applications*, vol. 60, pp. 192–219, 2016.

[290] L. Malina, J. Hajny, R. Fujdiak, and J. Hosek, "On perspective of security and privacy-preserving solutions in the internet of things," *Computer Networks*, vol. 102, pp. 83–95, 2016.

[291] R. Roman, P. Najera, and J. Lopez, "Securing the internet of things," *Computer*, vol. 44, no. 9, pp. 51–58, 2011.

[292] A. E. Omolara, A. Jantan, O. Isaac Abiodun, K. Victoria Dada, H. Arshad, and E. Emmanuel, "A deception model robust to eavesdropping over communication for social network systems," *IEEE Access*, vol. 7, Article ID 100881, 2019.

[293] M. Ferretti, S. Nicolazzo, and A. Nocera, "H2O: secure interactions in IoT via behavioral fingerprinting," *Future Internet*, vol. 13, no. 5, p. 117, 2021.

[294] N. Torres, P. Pinto, and S. I. Lopes, "Security vulnerabilities in LPWANs—an attack vector analysis for the IoT ecosystem," *Applied Sciences*, vol. 11, no. 7, p. 3176, 2021.

[295] H. Wang, Z. Zhang, and T. Taleb, "Editorial: special issue on security and privacy of IoT," *World Wide Web*, vol. 21, no. 1, pp. 1–6, 2018.

[296] B. R. Chandavarkar, "Hardcoded credentials and insecure data transfer in IoT: National and international status," in *Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–7, IEEE, Kharagpur, India, 2020.

[297] R. S. Verma, B. R. Chandavarkar, and P. Nazareth, "Mitigation of hard-coded credentials related attacks using QR code and secured web service for IoT," in *Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–5, IEEE, Kanpur, India, 2019.

[298] P. Ferrara, A. K. Mandal, A. Cortesi, and F. Spoto, "Static analysis for discovering IoT vulnerabilities," *International Journal on Software Tools for Technology Transfer*, vol. 23, no. 1, pp. 71–88, 2021.

[299] O. I. Abiodun, E. O. Abiodun, M. Alawida, R. S. Alkhawaldeh, and H. Arshad, "A review on the security of the internet of things: challenges and solutions," *Wireless Personal Communications*, vol. 119, no. 3, pp. 2603–2637, 2021.

[300] Y. Yu, L. Guo, S. Liu, J. Zheng, and H. Wang, "Privacy protection scheme based on CP-ABE in crowdsourcing-IoT for smart ocean," *IEEE Internet of Things Journal*, vol. 7, no. 10, Article ID 10061, 2020.

[301] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[302] C. Yin, J. Xi, R. Sun, and J. Wang, "Location privacy protection based on differential privacy strategy for big data in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3628–3636, 2018.

[303] X. Jiang, M. Lora, and S. Chattopadhyay, "An experimental analysis of security vulnerabilities in industrial IoT devices,"

ACM Transactions on Internet Technology, vol. 20, no. 2, pp. 1–24, 2020.

[304] N. N. Thilakarathne, “Security and privacy issues in iot environment,” *International Journal of Engineering and Management Research*, vol. 10, no. 1, pp. 26–29, 2016.

[305] V. Visoottiviseth, P. Sakarin, J. Thongwilai, and T. Choobanjong, “Signature-based and Behavior-Based Attack Detection with Machine Learning for Home IoT Devices,” in *Proceedings of the 2020 IEEE REGION 10 CONFERENCE (TENCON)*, pp. 829–834, IEEE, Osaka, Japan, 2020.

[306] N. Ferry and P. H. Nguyen, “Towards model-based continuous deployment of secure IoT systems,” in *Proceedings of the 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp. 613–618, IEEE, Munich, Germany, September 2019.

[307] H. Kim and E. A. Lee, “Authentication and authorization for the internet of things,” *IT Professional*, vol. 19, no. 5, pp. 27–33, 2017.

[308] M. Shahzad and M. P. Singh, “Continuous authentication and authorization for the internet of things,” *IEEE Internet Computing*, vol. 21, no. 2, pp. 86–90, 2017.

[309] M. Grabovica, S. Popić, D. Pezer, and V. Knežević, “Provided security measures of enabling technologies in Internet of Things (IoT): a survey,” in *Proceedings of the 2016 Zooming Innovation in Consumer Electronics International Conference (ZINC)*, pp. 28–31, IEEE, Novi Sad, Serbia, 2016.

[310] W. Li, T. Logenthiran, V.-T. Phan, and W. L. Woo, “A novel smart energy theft system (SETS) for IoT-based smart home,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5531–5539, 2019.

[311] A. M. Bossler and T. J. Holt, “On-line activities, guardianship, and malware infection: an examination of routine activities theory,” *International Journal of Cyber Criminology*, vol. 3, no. 1, 2009.

[312] Ž. Turk, B. García de Soto, B. R. K. Mantha, A. Maciel, and A. Georgescu, “A systemic framework for addressing cybersecurity in construction,” *Automation in Construction*, vol. 133, Article ID 103988, 2022.

[313] E. M. Rudd, A. Rozsa, M. Günther, and T. E. Boult, “A survey of stealth malware attacks, mitigation measures, and steps toward autonomous open world solutions,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1145–1172, 2017.

[314] I. ASaeed, A. Selamat, and A. M A Abuagoub, “A survey on malware and malware detection systems,” *International Journal of Computer Application*, vol. 67, no. 16, pp. 25–31, 2013.

[315] A. Al Hayajneh, M. Z. A. Bhuiyan, and I. McAndrew, “Improving internet of things (IoT) security with software-defined networking (SDN),” *Computers*, vol. 9, no. 1, p. 8, 2020.

[316] P. Čisar and S. M. Čisar, “General vulnerability aspects of internet of things,” in *Proceedings of the 2015 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, pp. 117–121, IEEE, Budapest, Hungary, November 2015.

[317] S. S. Basu, S. Tripathy, and A. R. Chowdhury, “Design Challenges and Security Issues in the Internet of Things,” in *Proceedings of the 2015 IEEE Region 10 Symposium*, pp. 90–93, IEEE, Ahmedabad, India, 2015.

[318] K. T. Nguyen, M. Laurent, and N. Oualha, “Survey on secure communication protocols for the Internet of Things,” *Ad Hoc Networks*, vol. 32, pp. 17–31, 2015.

[319] A. Alzubaidi and J. Kalita, “Authentication of smartphone users using behavioral biometrics,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1998–2026, 2016.

[320] K. Shepard, B. Wing, C. Miles, and D. Blackburn, “Iris Recognition Systems in a Non-Cooperative Environment,” *The Biometric Computing, Chapman and Hall/CRC*, 2006.

[321] P. Podder, M. R. H. Mondal, and J. Kamruzzaman, “Iris feature extraction using three-level Haar wavelet transform and modified local binary pattern,” in *Applications of Computational Intelligence in Multi-Disciplinary Research*-Elsevier, Amsterdam, Netherlands, 2022.

[322] M. Nappi, S. Ricciardi, and M. Tistarelli, “Deceiving faces: when plastic surgery challenges face recognition,” *Image and Vision Computing*, vol. 54, pp. 71–82, 2016.

[323] S. A. E. said and H. M. A. Atta, “Geometrical face recognition after plastic surgery,” *International Journal of Computer Applications in Technology*, vol. 49, no. 3/4, pp. 352–364, 2014.

[324] Q. Yan, M. Wang, W. Huang, X. Luo, and F. R. Yu, “Automatically synthesizing DoS attack traces using generative adversarial networks,” *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 12, pp. 3387–3396, 2019.

[325] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K.-K. R. Choo, “A deep recurrent neural network based approach for internet of things malware threat hunting,” *Future Generation Computer Systems*, vol. 85, pp. 88–96, 2018.

[326] S. Bharati and M. R. Hossain Mondal, “Computational Intelligence for Managing Pandemics,” in *12 Applications and Challenges of AI-Driven IoHT for Combating Pandemics: A Review*, A. Khamparia, R. Hossain Mondal, P. Podder, B. Bhushan, V. H. C. d. Albuquerque, and S. Kumar, Eds., De Gruyter, Berlin, Germany, 2021.

[327] M. R. A. Robel, S. Bharati, P. Podder, and M. R. H. Mondal, “IoT driven healthcare monitoring system,” *Fog, Edge, and Pervasive Computing in Intelligent IoT Driven Applications*, John Wiley & Sons, Hoboken, NY, USA, 2020.

[328] A. Canziani, A. Paszke, and E. Culurciello, “An Analysis of Deep Neural Network Models for Practical Applications,” 2016, https://arxiv.org/abs/1605.07678.

[329] S. M. Tahsien, H. Karimipour, and P. Spachos, “Machine learning based solutions for security of Internet of Things (IoT): a survey,” *Journal of Network and Computer Applications*, vol. 161, Article ID 102630, 2020.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 3810670, 11 pages
https://doi.org/10.1155/2022/3810670

Research Article

Edge Computing Server Placement Strategy Based on SPEA2 in Power Internet of Things

Yongling Lu, Zhen Wang, Chengbo Hu, Ziquan Liu, and Xueqiong Zhu

Electric Power Science Research Institute, State Grid Jiangsu Electric Power Co., Ltd., Nanjing 211103, China

Correspondence should be addressed to Yongling Lu; yongling_lu@hotmail.com

Received 10 May 2022; Revised 24 June 2022; Accepted 20 July 2022; Published 24 August 2022

Academic Editor: Qi Li

In order to meet the edge services placement demand for multiobjective optimization of Power Internet of Things, an edge services placement strategy based on an improved strength Pareto evolutionary algorithm (SPEA2) is proposed in this paper. Firstly, we model the delay, resource utilization, and energy consumption. Then, a multiobjective optimization is proposed. Finally, an enhanced genetic algorithm is used to derive the decision candidate set. Moreover, the optimal solution in the candidate set is selected to be utilized in the iteration of the multicriteria decision and the superior-inferior solution distance method. Numerical results and analysis show that the proposed strategy is more effective in reducing system delay, improving resource utilization, and saving energy consumption than the other two benchmark algorithms.

1. Introduction

With the rapid development of the Power Internet of Things (IoT), the IoT nodes of the power supply terminal, including smart devices and emerging applications, show explosive growth, which leads to massive heterogeneity and complex processing of the data [1, 2]. In the power industry, cloud computing architecture is usually used to upload terminal data to the cloud platform for centralized processing. However, the traditional cloud computing center is far away from the power grid equipment, and uploading data to the cloud platform can lead to large time delays [3]. In addition, centralizing data in the cloud platform can cause a burden on network communication and computing resources, resulting in transmission interruption or link congestion. Therefore, it is difficult for the cloud computing architecture to meet the service requirements of terminal equipment in the Power Internet of Things [4, 5].

Edge computing improves the service capability of the network by deploying the edge servers at radio access network side to provide power grid equipment with powerful computing and storage capabilities. Nowadays, edge computing has been widely used in many fields, such as mobile big data analytics and Power Internet of Things [6–8]. In addition, for the current power grid equipment, the deployment of edge computing could relieve the challenge caused by the lack of power and computing capacity. However, the proper edge service placement strategy needs to be designed to optimize other parameters such as energy consumption and resource utilization while simultaneously providing high-performance services to power grid equipment.

A lot of researches have been done on the service placement of edge computing and some constructive solutions have been proposed. In [9], to address the problem of edge computing service placement under resource constrained conditions, the authors have regarded that the ubiquitous MEC could implement service migration in mobile networks with highly dynamic characteristics by supporting multiserver collaboration. To maximize the system utility of the system, the optimization problem has been formulated by joint considering the constraints of server storage capacity and service delay. Firstly, the long-term optimization problem has been decomposed into a series of immediate optimization by the Lyapunov method. Then, a stochastic algorithm based on sample average approximation is proposed to approximate future expected system utility values. Next, the distributed Markov approximation algorithm is used to determine the service placement policy. For addressing cost and energy consumption in edge computing power scenarios, in [10], the authors have considered that the energy consumption of servers is an important part of service cost in edge computing systems. Thus, the energy-aware edge computing application service placement problem has been designed; then the problem has been modeled as a multistage stochastic programming problem. The objective is to maximize the Quality of Service (QoS), under the energy budget constraints of the edge computing server. Finally, a novel sampling average approximation algorithm has been designed to solve the problem.

To address the problem of multiobjective optimization requirements for the placement of edge computing service, in [11], the authors have considered that one of the main challenges of edge computing is to consider service load variations and determine multiobjective performance optimization to make service placement decisions. The optimal service placement problem has been solved by further considering how to allocate the service loads placed at different locations. And a dynamic predictive service combined with load allocation strategy has been proposed by estimating the performance-cost tradeoff for service migration. The strategy has utilized the small amount of predictive processing to reduce the impact of load fluctuations. In [12], the authors have defined a network entity with a flexible allocation of communication, computing, and storage capabilities so that the resource constrained devices could use the communication and computation resources required for the service. In addition, the spectrum-aware service placement in edge computing has been investigated. The authors have formulated the service placement as a stochastic optimization problem. Then, the authors have jointed the optimized service placement, traffic routing, and spectrum allocation. Based on those, an enhanced coarse-grained service placement algorithm has been proposed.

Edge computing service architectures have recently attracted a lot of attention. In [13], the authors consider the multidimension of task requirements in mobile crowd sensing and propose a task-oriented user selection incentive mechanism to achieve higher task completion rate and maximize resource utilization. In order to solve the problem of insufficient accuracy of the medium-edge service model in the Industrial Internet of Things, a new smart contract was constructed in [14] to encourage multiple marginal service users to participate, thereby improving the model accuracy. In addition, a scale weighted aggregation strategy was proposed to verify the model parameters to improve the accuracy of the model. In [15], the Graph theory was introduced into the edge caching network architecture to reduce the processing complexity. Considering the physical attributes and social attributes, a cache solution based on physical-social weighted direction is proposed to minimize the average download latency of all edge users within a macrocell.

The improved genetic algorithm used in this paper has been partially explored by some scholars in the field of service placement using genetic algorithms. In [16], the authors have proposed an algorithm that combined the genetic algorithm with Monte Carlo simulations. The algorithm can greatly improve the efficiency of exhaustive search service placement strategy. First, an optimization model has been developed for the genetic algorithm; the main body of the model has been the QoS objective function, cost objective function, and the resource utilization objective function. Then, the FogTorch Monte Carlo framework has been utilized to address the problem. The proposed algorithm could minimize the resource consumption and service placement cost in the fog while guaranteeing QoS. By defining a representation of an application placement in a biased-random-key chromosome and using a fault-tolerance distributed pool model, the GRECO algorithm was proposed in [17] to solve the application placement problem in constrained hybrid cloud environment. In this paper, the multiobjective problem is also optimized using a genetic algorithm but different from [16, 17]. We utilize multicriteria decision and superior-inferior solution distance method to combine three fitness functions into a single meritocratic function to assist the search.

In this paper, we focus on multiobjective optimization requirements in power scenarios. To address the above issues, we develop an improved genetic algorithm based edge service placement (IGA-ESP) strategy, which optimizes the delay, energy consumption, and resource utilization parameters. The main contributions of this paper are summarized as follows: (1) We study the edge service placement problem in the Power Internet of Things and establish a multiobjective optimization problem under the constraint of the edge cloud capacity and a single service request per time slot. The objectives of this problem include minimizing service delay and energy consumption while maximizing resource utilization. (2) We propose the IGA-ESP strategy to solve this multiobjective optimization problem. Firstly, the decision candidate set is obtained by using the improved genetic algorithm. Then, the optimal solution in the candidate set is filtered using the multicriteria decision and the superior-inferior solution distance method. (3) Simulation results show that the proposed strategy can effectively reduce system delay, improve resource utilization, and save energy consumption. Furthermore, compared with TS algorithm and Greedy algorithm, IGA-ESP strategy can reduce the average end-to-end delay of power grid equipment by 7.8% and 16.7%, respectively.

2. System Model

In Power Internet of Things, edge computing networks can provide powerful infrastructure resource and value-added service capabilities for power grid terminal applications with insufficient power, computing power, and storage resource, such as remote monitoring, smart home, and VR. The guarantee of low-latency services requires a reasonable edge service placement strategy. In this section, the edge computing servers are deployed at the edge of the network closer to the power grid equipment. Besides, the computing power of edge computing is utilized to process service requests from power grid equipment, close to the edge of the network. In this section, we first present the network model. Then, the placement model of each edge server is proposed. Finally, we present the wireless communication model between the grid equipment and the edge cloud.

2.1. Network Model. The network architecture is shown in Figure 1. There are multiple Small Base Stations (SBS) in the coverage area of Macro Base Station (MBS). In addition, all the SBS with edge computing server enhancements are called edge cloud (EC), and it is expressed as $E = \{1, 2, 3, \ldots, i, \ldots, N\}$. To improve service quality for power grid equipment and reduce service deployment costs for Application Service Provider (ASP), ASPs deploy a limited number of power popular application services, such as intelligence operations and video surveillance service in each EC by MBS. The set of all the service types of the system is represented by $\kappa = \{1, 2, 3, \ldots, k, \ldots, K\}$. In this model, the power grid equipment first uploads the service request to the local EC through the wireless channel. If the requested service already exists in the local EC, the power grid equipment will be served by the EC; otherwise the service request of the power grid equipment will be uploaded to the MBS via the local EC and forwarded to the cloud server of the ASP by the MBS.

2.2. Service Placement Model. The service placement model is implemented using containers, which are configured to allocate resource to power grid equipment to provide edge services [18].

It is assumed that each edge server has a certain number of unit containers and each unit container has a fixed amount of storage and compute resources. In addition, all edge servers use the same size unit container but are differences in the number of unit containers. Each container occupies an integer number of unit container resource. ASP places the service k on the EC i at the slot t denoted by $y_i^k(t) = 1$; otherwise, $y_i^k(t) = 0$. We define the number of containers per unit that the storage service k needs to occupy as r_k. The limited number of unit containers of EC results in the number of containers per unit occupied by the hosted service of EC i which needs to meet the following constrains at the slot t.

$$\sum_{k=1}^{K} r_k y_i^k(t) \leq R_i, \tag{1}$$

where r_k denotes the number of containers per unit that the service k needs to occupy. R_i denotes the total number of unit containers owned by EC i.

2.3. Wireless Communication Model. The power grid equipment uploads the service request to the local EC which would incur a wireless communication delay; thus, we assume that the data to be uploaded after the service requested by the power grid equipment has been processed as $d_{i,j}^k(t)$, and the uplink channel gain between the power grid equipment j and the base station i is $H_{i,j}^k(t)$. The effect on the channel gain is negligible because the change of power grid equipment location within a time slot is very small. Therefore, the channel gain between the power grid equipment and the base station is assumed to be constant within a time slot. The transmission power of the power grid equipment is represented by $P_j(t)$; thus, the uplink transmission bandwidth between the power grid equipment j and the EC i can be expressed according to the Shannon channel capacity formula as

$$c_{i,j}^k(t) = B \log 2\left(1 + \frac{P_j(t) H_{i,j}(t)}{N_0 B + I}\right), \tag{2}$$

where B represents the channel bandwidth. N_0 is bilateral power spectral density of additive white Gaussian noise. I denotes the stochastic noise power.

3. Problem Formulation and Analysis

The edge computing networks can provide distributed computing resources and low-latency services for power grid equipment with insufficient battery capacity and computing resources. A reasonable edge service placement strategy can enhance the power grid equipment service quality in the edge computing networks. The resource constraints reduced QoS of delay-sensitive tasks and heavy traffic load applications. Thus, we deploy the edge computing server at the Power Internet of Things network edge closer to the power grid equipment and utilize the computing capacity of the edge computing to process power grid equipment service requests close to the network edge. In this section, we first introduce the heterogeneous service network architecture of the edge computing network. Then we calculate the service placement model and communication model according to the power grid equipment access network conditions.

3.1. Service Delay Model. In the edge computing network, $U_i, i \in E$, represents the number of power grid equipment served by EC i. Considering the limited computing resources and battery capacity of power grid equipment, we upload the power grid equipment's service requests to the covered edge cloud for computing and processing [19, 20].

$x_{i,j}^k(t), k \in \kappa$, denotes the service type requested by power grid equipment j in edge cloud i at time t. $x_{i,j}^k(t) = 1$ represents the k-th service required by power grid equipment j in edge cloud i; otherwise $x_{i,j}^k(t) = 0$. We assume that power grid equipment j served by edge cloud i at any time t can only request a type of service, and $x_{i,j}^k(t)$ is expressed as

$$\sum_{k=1}^{K} x_{i,j}^k(t) = 1, \quad \forall i \in E,\ 1 \leq j \leq U_i. \tag{3}$$

In addition, in Power Internet of Things, considering the limited computing and storage resources of edge servers, ASP can only deploy limited services in each edge server, so ECs in service hotspot areas are prone to overload. In this regard, if the associated EC of a power grid equipment does not host the service required by the power grid equipment,

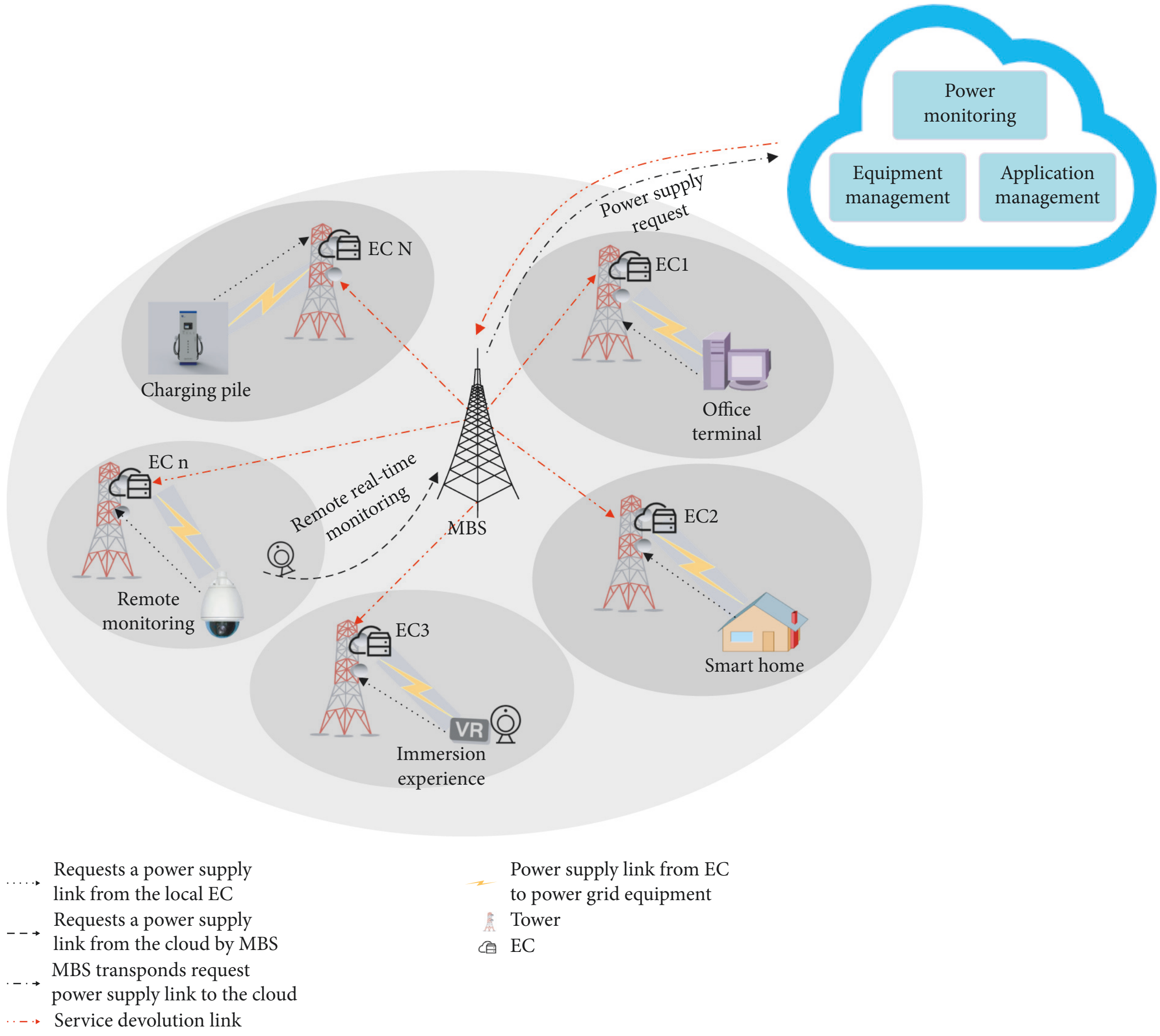

Power monitoring
Equipment management
Application management
Power supply request
EC N
Charging pile
EC1
Office terminal
EC n
Remote real-time monitoring
MBS
EC2
Remote monitoring
EC3
Smart home
Immersion experience
VR
Requests a power supply link from the local EC
Requests a power supply link from the cloud by MBS
MBS transponds request power supply link to the cloud
Service devolution link
Power supply link from EC to power grid equipment
Tower
EC

Figure 1: Edge computing service placement architecture.

the power grid equipment can only upload the service request to the ASP cloud server hosting all services through the EC. Therefore, the calculation of the power grid equipment's uplink transmission delay mainly includes two cases.

(i) When there is the k-th service requested by power grid equipment j in EC i, the uplink transmission delay of power grid equipment j served by EC i can be calculated as

$$T_{i,j}^{k,e}(t) = \frac{d_{i,j}^{k}(t)}{c_{i,j}^{k}(t)}. \qquad (4)$$

(ii) When there is no k-th service requested by power grid equipment j EC i, the requested service can only be uploaded to the MBS through the EC and then forwarded to the cloud center through the core network. Thus, the uplink transmission delay of power grid equipment j served by EC i can be derived as

$$T_{i,j}^{k,c}(t) = d_{i,j}^{k}(t)\left\{\frac{1}{c_{i,j}^{k}(t)} + \frac{1}{c_{i}^{\text{cloud}}} + \frac{1}{c^{\text{core}}}\right\}, \qquad (5)$$

where c_i^{cloud} represents the backhaul link bandwidth between EC i and MBS. c^{core} denotes the data transmission rate of the core network. Therefore, the final uplink transmission delay of power grid equipment j served by EC i can be expressed as

$$T_{i,j}^{k}(t) = x_{i,j}^{k}(t)\left[y_{i}^{k}(t)T_{i,j}^{k,e}(t) + \left(1 - y_{i}^{k}(t)\right)T_{i,j}^{k,c}(t)\right]. \qquad (6)$$

Subsequently, we model the calculation delay. Considering the limited computing capacity of EC and the cloud computing center, there will be a certain computing delay when the service requested by power grid equipment is completed. c_k denotes the number of central processing unit (CPU) cycles required to complete the k-th service in edge cloud, and the unit is CPU cycle. Simultaneously, the computing capacity of the unit container is expressed as F^{rb}, and the unit is CPU cycle/s.

Thus, the calculation delay of power grid equipment j served by EC i can be calculated as

$$T_{i,j}^{k,ec}(t) = \frac{x_{i,j}^{k}(t)y_{i}^{k}(t)c_k}{F^{rb}r_k}. \tag{7}$$

It is worth noting that the power grid equipment's service request can only be uploaded to the cloud server of ASP through the Power Internet of Things local EC when the power grid equipment's service request cannot be responded to by the local edge cloud. As the cloud server has a strong computing capacity, it is assumed that the cloud server will always provide the services for each power grid equipment with F^c computing capacity, and the unit is CPU cycle. Thus, when the service request of power grid equipment j served by EC i is responded to by the cloud server, the calculation delay can be derived as

$$T_{i,j}^{k,cc}(t) = \frac{x_{i,j}^{k}(t)\left(1 - y_{i}^{k}(t)\right)c_k}{F^{c}}. \tag{8}$$

For the condition of cloud server storage service, we assume that there are all types of services in the cloud server of ASP [21].

To sum up, the total computing delay of power grid equipment j served by EC i is expressed as

$$T_{i,j}^{\text{comp}}(t) = T_{i,j}^{k,ec}(t) + T_{i,j}^{k,cc}(t). \tag{9}$$

Therefore, the end-to-end delay of all power grid equipment requesting services in all ECs at time t can be calculated as

$$D(t) = \sum_{i=1}^{N}\sum_{j=1}^{U_i}\sum_{k=1}^{K} T_{i,j}^{k}(t) + T_{i,j}^{\text{comp}}(t). \tag{10}$$

3.2. Resource Utilization Model. We allocate containers for power grid equipment services to provide services [22], and each container occupies a certain unit container. The resources of the cloud center are relatively sufficient, so the resource utilization only refers to the utilization of the edge cloud unit container, and the resource utilization of the edge cloud is expressed as

$$r_{i}^{\text{ratio}}(t) = \frac{1}{R_i}\sum_{k=1}^{K} r_k \cdot x_{i,j}^{k}(t). \tag{11}$$

Then, the resource utilization of all participating edge clouds is expressed as

$$\text{RU}(t) = \frac{1}{N}\sum_{i=1}^{N} r_{i}^{\text{ratio}}(t). \tag{12}$$

3.3. Energy Consumption Model. The edge computing server energy consumption in the Power Internet of Things networks is mainly divided into two categories: (1) the basic energy consumption to ensure the operation of the edge cloud and the cloud center, and (2) the computing unit container energy consumption used by the edge cloud and the cloud center to provide services.

The key factor that affects the basic energy consumption of edge cloud is the service duration t_{si} of edge cloud n, which is determined by the service that provides power grid equipment with the longest service time among all services and is given as

$$t_{si} = \max_{k=1}^{K} x_{i,j}^{k}(t)y_{i}^{k}(t) \cdot T_{i,j}^{k,ec}(t), \tag{13}$$

where $T_{i,j}^{k,ec}(t)$ denotes the execution time of service k in edge cloud i.

The basic energy consumption of all edge clouds is expressed as

$$P_{b}^{e}(t) = \sum_{i=1}^{N} t_{si}P_{\text{EC}}. \tag{14}$$

P_{EC} represents the operating basic power of the edge cloud. To facilitate representation, we assume that the operating power of all edge clouds is stable and constant, and all edge clouds operate at the same basic power.

The cloud center service duration is determined by the service with the longest execution time in the cloud, calculated as

$$t_{s0} = \max_{k=1}^{K} x_{i,j}^{k}(t)\left(1 - y_{i}^{k}(t)\right) \cdot T_{0,j}^{k,\text{ec}}(t), \tag{15}$$

where $T_{0,j}^{k,\text{ec}}(t)$ denotes the execution time of service k in the cloud center.

The basic energy consumption of the cloud center is calculated as

$$P_{b}^{c}(t) = t_{s0} \cdot P_c, \tag{16}$$

where P_c is the operating basic power of the cloud center. For the convenience of representation, we assume that the operating power of the cloud center is unvarying.

The total energy consumption of all unit containers required to request services from the edge is expressed as

$$P_{s}^{e}(t) = \sum_{i=1}^{N}\sum_{j=1}^{U_i}\sum_{k=1}^{K} x_{i,j}^{k}(t)y_{i}^{k}(t) \cdot T_{i,j}^{k,\text{ec}} \cdot r_k \cdot \gamma, \tag{17}$$

where γ represents the average operating power of the unit container. For the convenience of manifestation, it is assumed that the average running power per unit container required for all deployment services is stable and unchangeable.

The energy consumption of computing resources required to request services from the cloud center is calculated as

$$P_{s}^{c}(t) = \sum_{k=1}^{K} x_{i,j}^{k}(t)\left(1 - y_{i}^{k}(t)\right) \cdot T_{0,j}^{k,\text{cc}} \cdot \zeta. \tag{18}$$

The computing resources allocated by the cloud center are always stable, so it is assumed that the energy consumption per unit time generated by allocating computing resources in the cloud is ζ.

Therefore, after the service placement decision is made at time t in the system, its total energy consumption is expressed as

$$P(t) = P_b^e(t) + P_b^c(t) + P_s^e(t) + P_s^c(t). \tag{19}$$

3.4. Problem Formulation. To sum up, for any edge cloud EC i and power grid equipment j, the goal of this paper is to minimize the power grid equipment-aware end-to-end service delay $D(t)$ and the energy consumption $P(t)$ of edge cloud in the Power Internet of Things and maximize the utilization of edge cloud resources $RU(t)$, that is, to minimize energy consumption while improving overall service performance, and the problem is formulated as

$$\begin{aligned} & P1 \min D(t), P(t) \\ & \max RU(t) \\ & C1 \sum_{k=1}^{K} r_k y_i^k(t) \le R_i \\ & C2 \sum_{k=1}^{K} x_{i,j}^k(t) = 1, \forall i \in E,\ 1 \le j \le U_i \end{aligned}, \tag{20}$$

where $C1$ indicates that the number of unit containers requesting services cannot exceed the total number of edge cloud unit containers. $C2$ means that each power grid equipment requests only one service at a time t. Because the problem $P1$ is a multiobjective optimization problem and NP-hard, it is difficult to solve the problem directly. The heuristic algorithm has strong robustness and global search ability and is widely used in various optimization problems. In addition, the heuristic algorithm is more efficient than the traditional search algorithm and can obtain the approximate global optimal solution in a short time. Therefore, in this paper, we consider using the improved genetic algorithm to solve the problem.

4. Edge Service Placement Strategy Based on Improved Genetic Algorithm

Genetic algorithm is an adaptive heuristic intelligent search algorithm that simulates the evolutionary process in nature to solve optimization problems [23]. The algorithm updates individuals through selection, crossover, and mutation operations and obtains an approximate optimal solution after several generations of evolution. Moreover, it can automatically adjust the search direction according to the population selection, so that it has a better global optimization ability. Compared with other heuristic algorithms, genetic algorithm avoids falling into a local optimum in the solution process through gene mutation, making the solution closer to the global optimum. Therefore, it is more suitable for solving complex nonlinear optimization problems [24].

In this section, we propose an edge computing service placement strategy based on IGA-ESP, which achieves multiobjective optimization. In this strategy, population chromosomes can be used to represent candidate solutions to the problem. If a solution satisfies the constraints of the problem, it is feasible; otherwise, it is infeasible. The proposed improved genetic algorithm uses chromosome representation, crossover, and mutation operators according to the needs of the problem, which are used to penalize infeasible solutions, so that these infeasible solutions have a smaller selection (or survival) probability. Specifically, we first use the IGA-ESP algorithm to select the candidate solutions. Then, we use the distance method of superior and inferior solutions and the multicriteria decision selection genetic algorithm to generate the optimal placement decision among the candidate decisions.

4.1. Service Placement Strategy Based on Improved Genetic Algorithm

4.1.1. Chromosomes and Chromosome Codes. For all genetic algorithms, what is considered firstly is how the chromosomes are encoded. In this paper, each chromosome is represented by an integer array, and each chromosome represents a complete service placement strategy; thus all solutions of the problem space can be expressed as the designed genotype, and any genotype corresponds to a possible solution, in which the array of each element corresponds to a service placement decision parameter for each service placement strategy. Actually, the service placement decision of each EC is an array. The algorithm proposed in this paper defines the chromosome as the set of all array service placement strategies. We first number the ECs, then put the service placement decision parameters into the array in order, and finally get an array with a length of NK. Numbers 1 and 0 in the array indicate that the service is placed in and not placed in the edge cloud, respectively.

4.1.2. Initialization. The initial population size in genetic algorithms has a critical influence on searchability. If the size is small, searchability is limited and rapid convergence occurs in early runs. Conversely, a larger initial population size leads to dispersion of solutions, which affects the efficiency and effectiveness of the algorithm. In addition, the crossover probability, mutation probability, and the number of iterations need to be set. The setting of these parameters requires extensive experimental exploration.

4.1.3. Selection Operator. In this paper, we adopt a binary tournament selection operator to select individuals with better performance, thereby enhancing the performance of the algorithm.

The binary tournament selection operator compares two individuals. If the matching pool is sufficient, a pruning process is performed to remove individuals with poor fitness. If the match pool is insufficient, the selection process continues until the match pool is sufficient. The selection operator can gradually eliminate inferior genes, so the performance of the algorithm can be promoted in the continuous iterative process.

4.1.4. Crossover and Mutation Operator. We exploit the single-point crossover operator to randomly select crossover points from 1 to the number of genes per chromosome. The single-point crossover operator refers to first selecting a crossover point in genes with a certain random probability and then exchanging the gene codes located in the same position to generate new individuals. The mutation operator alters partial genes in a single chromosome with a certain probability, resulting in better chromosomes and preventing rapid convergence. Inappropriate mutation probability will have a malign influence on the algorithm results. The confirmation of the mutation operator requires repeated attempts, and the optimal mutation operator is selected by exploring the actual effect of the algorithm obtained by multiple mutation operators within a reasonable range. The mutation operator ensures the diversity of the population and has a very critical impact on the local search ability of the genetic algorithm.

4.1.5. Fitness Function and Constraints. The fitness function is utilized to calculate the environmental fitness of each individual. Representing solutions as individuals, all solutions constitute a population. The fitness function needs to be divided into three types: power grid equipment-aware delay fitness function, resource utilization fitness function, and energy consumption fitness function. Considering there is only one function for the fitness function judgment of the genetic algorithm, the selection function of the optimal solution will be obtained after processing these three functions through the multiattribute decision in the multicriteria decision and the superior-inferior solution distance method.

4.2. Optimal Strategy Using Superior-Inferior Solution Distance and Multicriteria Decision. Among all the strategies generated by the improved SPEA2 algorithm, the optimal placement strategy is obtained by using the superior-inferior solution distance method and the multicriteria decision method. In the superior and inferior solution distance method, the solutions are sorted according to the Euclidean distance between the candidate solution and the superior and inferior solution, and the superior solution is defined as the object closest to the ideal solution and furthest away from the negative ideal solution. Similarly, the inferior solution is defined as the object closest to the negative ideal solution and farthest from the ideal solution.

We assume that SPEA2 obtains H strategies to wait for the next step after analyzing all the strategies. Each strategy includes delay, resource utilization, and power consumption, which can be denoted as $D_{\text{attribute}} = [D_1, D_2, D_{31}, \ldots, D_H]$, $R_{\text{attribute}} = R_1, R_2, R_3, \ldots, R_H$, and $E_{\text{attribute}} = E_1, E_2, E_3, \ldots, E_H$, respectively.

The normalized delay can be expressed as

$$V_h^D = \frac{D_h}{\sqrt{\sum_{h=1}^{H} D_h^2}}. \tag{21}$$

The normalized resource utilization can be expressed as

$$V_h^R = \frac{R_h}{\sqrt{\sum_{h=1}^{H} R_h^2}}. \tag{22}$$

The normalized energy consumption can be expressed as

$$V_h^E = \frac{E_h}{\sqrt{\sum_{h=1}^{H} E_h^2}}. \tag{23}$$

The weight values of delay, resource utilization, and energy consumption are ω_D, ω_R, and ω_E, respectively. Therefore, their weighted normalized values are defined as

$$\begin{aligned} W_h^D &= \omega_D \cdot V_h^D, \\ W_h^R &= \omega_R \cdot V_h^R, \\ W_h^E &= \omega_E \cdot V_h^E. \end{aligned} \tag{24}$$

We aim to maximize resource utilization and delay and minimize the energy consumption in the Power Internet of Things through the analysis of the problem. Therefore, we define the resource utilization as the ideal solution, while delay and energy consumption are the nonideal solution. $W_{\max}^R$, $W_{\max}^D$, and $W_{\max}^E$ are the maximum values of the three targets, while $W_{\min}^R$, $W_{\min}^D$, and $W_{\min}^E$ are the minimum values.

The distance between the ideal solution and alternative solution can be denoted as

$$D_h^{\text{IS}} = \sqrt{\left(W_h^R - W_{\max}^R\right)^2 + \left(W_h^D - W_{\max}^D\right)^2 + \left(W_h^E - W_{\max}^E\right)^2}. \tag{25}$$

The distance between the nonideal solution and alternative solution is given by

$$D_h^{\text{NS}} = \sqrt{\left(W_h^R - W_{\min}^R\right)^2 + \left(W_h^D - W_{\min}^D\right)^2 + \left(W_h^E - W_{\min}^E\right)^2}. \tag{26}$$

The proximity between the ideal solution and alternative solution can be represented as

$$C_h^{\text{IS}} = \frac{D_h^{\text{NS}}}{D_h^{\text{NS}} + D_h^{\text{IS}}}. \tag{27}$$

According to the proximity of alternative solutions, the superior solution can be expressed as

$$\begin{aligned} &OP = \max_{h=1}^{H} C_h^{\text{IS}}, \\ &\text{s.t. } \omega_D + \omega_R + \omega_E = 1, \\ &\omega_D, \omega_R, \omega_E \in [0, 1], \end{aligned} \tag{28}$$

where the constraints indicate that the weights of delay, resource utilization, and energy consumption are 0 to 1, and the sum of the three weight factors is equal to 1.

The specific process of IGA-ESP is summarized in Algorithm 1. The workflow of IGA-ESP is shown in Algorithm 1. The input of the algorithm is the iteration times I, and the output is the optimal service placement strategy OP. The algorithm obtains the service set and performs crossover and mutation operations firstly, then performs the calculation of fitness function and selects the best individual for the next generation, next calculates the proximity of service placement strategy, and selects the placement strategy with the maximum proximity. The above process is repeated until the maximum number of iterations is reached to output the best strategy.

5. Numerical Results and Analysis

5.1. Simulation Parameter Settings. Matlab platform is very suitable for the simulation of complex systems because of its powerful computing ability. To shorten the simulation time, Huawei FusionServer Pro rack servers with strong computing performance are used for cloud computing, virtualization, high-performance computing, databases, and SAP HANA computation-intensive scenarios. In addition, the integrated high reliability design of the whole process, BSST system startup accelerated storage, DEMT smart energy efficiency, FDM smart diagnosis, and other technologies can further improve system performance.

In Power Internet of Things, we assume that the number of edge clouds within the coverage area of MBS is 3, and user power grid equipment is distributed within each EC. This paper assumes that the number of edge clouds within the coverage area of MBS is 3, and user power grid equipment is evenly distributed within each EC uniformly. The uplink transmission bandwidth allocated by the edge cloud for each power grid equipment is 10 Mbps. The number of service requests in the Power Internet of Things system is 4. Each edge cloud has a unit container range of 50 to 200. The storage capacity of the unit container is 1 GB, and the computing capacity of the unit container is 1 GHz. Other simulation parameter settings are shown in Table 1.

In this section, we compare the IGA-ESP algorithm with other two benchmark algorithms. The first one is the Tabu Search (TS) algorithm [25]; the algorithm has considered the cost optimal service placement problem and proposed a delay aware service placement strategy based on the placement cost, which can guarantee the minimum QoS requirements of the service and balance the delay performance and deployment cost. The other one is the Greedy algorithm [26]; the algorithm can meet the requirements of load balancing and delay performance and reduce the problem of QoS degradation caused by edge computing resource constraints.

The power grid equipment-aware delay is an important parameter to determine the network performance. Resource utilization and energy consumption are mainly considered to reduce costs, which must be based on the delay performance. The tradeoff among lower delay value, higher resource utilization, and lower energy consumption can be achieved by adjusting weight parameters. Since the weight parameters of delay, resource utilization, and energy consumption are determined by ASP in power scenarios; in this simulation, the weight parameter of delay is set to 2/3, and the weight parameters of resource utilization and energy consumption are set to 1/6.

5.2. Simulation Results Analysis. In Figure 2, it depicts the relationship between the number of power grid equipment and the average end-to-end delay of power grid equipment. It can be found that the IGA-ESP algorithm can achieve the best performance. Compared with TS algorithm and Greedy algorithm, IGA-ESP algorithm reduces the average end-to-end delay by 7.8% and 16.7% under different power grid equipment. The average delay of the three algorithms increases with the increase of the number of power grid equipment. As the number of power grid equipment increases, edge servers cannot deal with all tasks locally, and some services need to be transmitted to the cloud center through MBS, which will increase delay. Moreover, with the increase of power grid equipment, the types of requested services also increase. The edge server cannot store services that meet all requests of power grid equipment. A large number of power grid equipment need to request services from the cloud server, which results in an obvious increase in delay when the number of power grid equipment changes from 9 to 12. The average end-to-end delay of power grid equipment gradually slows down when the number of power grid equipment continues to increase. It is because, with the increase of power grid equipment, the edge servers are nearly full, and some services begin to turn to the cloud center. Even if the number of power grid equipment continues to increase, the growth trend is not obvious when the edge servers are not full. Further, it can be seen that the TS algorithm minimizes the service placement cost while meeting the power grid equipment QoS. Therefore, more services are placed in the cloud, resulting in higher delay. Greedy algorithm considers delay and load balance; its delay is close to the IGA-ESP algorithm at first; however, with the increase of the number of power grid equipment, its delay is higher and higher.

Figure 3 describes computing power per container versus average delay of power grid equipment. When the computing capacity of unit container increases gradually, the average end-to-end delay shows a decreasing trend, and the IGA-ESP algorithm always has the lowest delay. It can be seen that when the computing ability of the unit container is 0.25 GHz, the edge cloud computing ability is too weak. Even if the transmission delay of cloud computing is long, the performance is still better than that of the service placement of edge computing. Therefore, these three algorithms choose to place all services in the cloud for processing at the initial time. When the unit container computing ability begins to increase, IGA-ESP and Greedy algorithm move some services that were not significant to the edge, which is dependent on the computing power, and the average end-to-end delay was reduced by 0.125 s. As the computing ability per container continues to increase, the delay performance of the IGA-ESP algorithm is gradually better than that of the Greedy algorithm and much

```
Input: maximum number of iterations I
Output: optimal service placement strategy OP
getting service set
fors = 1 to S do
    $i = 1$
    while$i < I$do
        mutating and crossover
        for all individuals in the population do
            calculating $D(t)$ using (10)
            calculating $RU(t)$ using (12)
            calculating $P(t)$ using (19)
        end for
        selecting and confirming the offspring
        $i = i + 1$
    end while
    estimating the relative proximity $C_h^{IS}$ according to (26)
    selecting the best service placement strategy OP according to (27)
end for
return OP
```

Algorithm 1: IGA-ESP.

Table 1: Simulation parameter setting.

Parameter	Value
Storage capacity required by each service	20~80 Gb
Number of CPU cycles required by each service	10~50 gigacycles
Computing ability provided by the cloud server	50 Ghz
Data size of each service request	0.1~0.3 MB
Power of edge cloud	300 W
Operating power of edge cloud	5 W
Basic power of cloud center	1500 W
Computing power of cloud center	800 W
Number of iterations	500
Probability of crossover	0.9
Probability of mutation	0.007

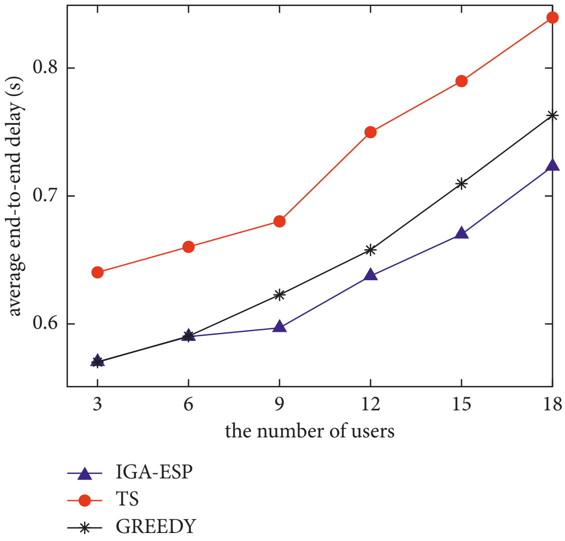


Figure 2: The number of power grid equipment versus average end-to-end delay.

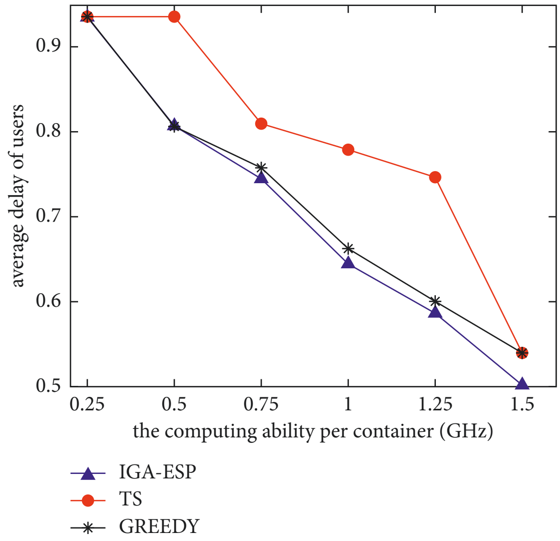


Figure 3: The computing ability per container versus average delay of power grid equipment.

lower than that of the TS algorithm. It indicates that the cost considered in TS algorithm has a great impact on the delay performance. When the computing ability of unit container increases, the performance of the two algorithms is basically consistent and worse than the IGA-ESP algorithm.

Figure 4 shows the unit container computing ability versus resource utilization. It can be seen that, with the increase of the computing ability of the unit container, the resource utilization shows an upward trend. When the computing ability of the unit container reaches 1 GHz, the resource utilization of IGA-ESP and TS algorithm reaches the saturation state under the current parameters. TS algorithm prefers to directly request services from remote cloud center; to consider the cost of service placement, its

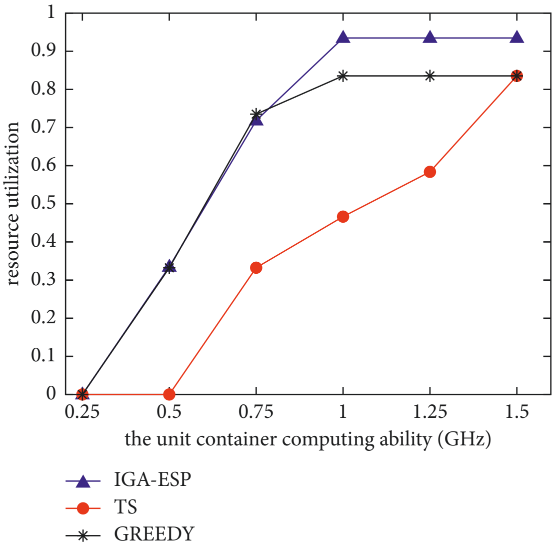


Figure 4: The unit container computing ability versus resource utilization.

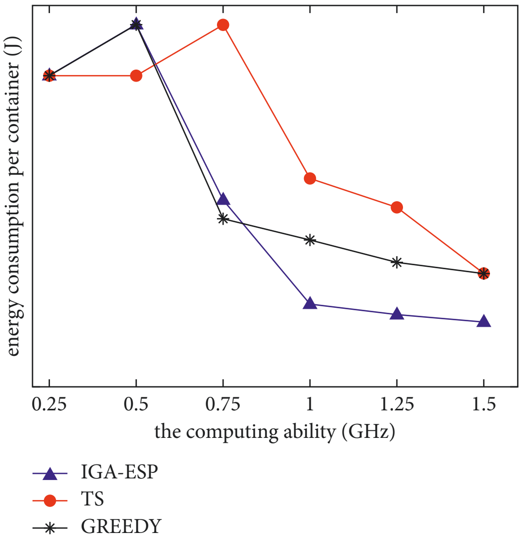


Figure 5: The computing ability versus energy consumption per container.

edge resource utilization has always maintained a low value. However, when the edge computing ability is very strong and reaches 1.5 GHz, its resource utilization suddenly reaches 0.83 together with TS algorithm. It indicates that the cost-centered TS algorithm only considers to place the service to the edge cloud when the edge performance is extremely strong and the delay is extremely low. The utilization of edge resource reflects the cost effect of the algorithm to some extent. After the edge server is established, the efficiency of ASP can be improved by deploying more services to the edge.

Figure 5 illustrates the computing ability versus energy consumption per container. As can be seen, the total energy consumption of the three algorithms has a trend of rising first and then decreasing and gradually flattening. The reason for the increase in energy consumption is the addition of the basic power consumption of edge cloud. Although the basic power consumption and operating power consumption of cloud computing are both high when only cloud computing is used to provide services at the beginning, there is no edge server, so the basic power consumption of three edge servers is 900 W. When more services are placed in the edge cloud, the total energy consumption of the system drops sharply. This is because the execution power of edge computing is far less than that of cloud center. When the execution power of edge computing is distributed among various services, the influence on the total energy consumption curve of the system becomes smaller. Another reason is that the increase of edge cloud computing ability will lead to a continuous decrease in computing ability. The total energy consumption will also decrease under the same power consumption. Finally, when a large number of services are deployed to the edge cloud, the low power characteristic of edge computing gradually flattens out. Even if more services are placed to the edge for hosting, they cannot be lower than the threshold limit of system energy consumption. It also implies that the IGA-ESP algorithm has the characteristics of low power consumption, which can reduce the total energy consumption of the system.

6. Conclusions

This paper studies the problem of edge computing service placement multiobjective optimization in Power Internet of Things system. Considering that the transmission distance of mobile cloud service is too long to guarantee the delay and the energy consumption, the edge server with limited resources is deployed at the power grid equipment edge of the network side to realize the nearby service firstly. Secondly, the service delay, resource utilization, and energy consumption are modeled. Finally, an edge service placement strategy based on SPEA2 algorithm is proposed, which can improve the overall performance of EC while optimizing multiple objectives and control the cost by reducing energy consumption. Simulation results show that, compared with the other two benchmark algorithms, the proposed strategy can effectively reduce system delay, improve resource utilization, and save energy consumption.

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 research was supported by the Foundations of State Grid Corporation of China, under Grant no. J2021207.

References

[1] J. Liu, X. Zhao, P. Qin, S. Geng, and S. Meng, "Joint dynamic task offloading and resource scheduling for WPT enabled space-air-ground power Internet of things," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 2, pp. 660–677, 2022.

[2] J. Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A survey of honeypots and honeynets for Internet of things, industrial Internet of things, and cyber-physical systems," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2351–2383, 2021.

[3] M. Rohith, A. Sunil, and Mohana, "Comparative analysis of edge computing and edge devices: key technology in IoT and computer vision applications," in *Proceedings of the 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, pp. 722–727, Bangalore, India, August 2021.

[4] X. Li, L. Huang, H. Wang, S. Bi, and Y.-J. A. Zhang, "An integrated optimization-learning framework for online combinatorial computation offloading in MEC networks," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 170–177, February 2022.

[5] S. Gong, M. Li, S. Wu, H. Cheng, and X. Yin, "Intelligent networking model at the edge of the power Internet of Things," in *Proceedings of the 2021 IEEE 5th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC)*, pp. 841–844, Xi'an, China, October 2021.

[6] M. Babar, M. A. Jan, X. He, M. U. Tariq, S. Mastorakis, and R. Alturki, "An optimized IoT-enabled big data analytics architecture for edge-cloud computing," *IEEE Internet of Things Journal*, p. 1, 2022.

[7] G. Huang, G. Chen, J. Yi, M. Huang, and Y. Zhang, "Workload modelling method of edge computing terminals for distribution service under power Internet of things," in *Proceedings of the 2021 6th Asia Conference on Power and Electrical Engineering (ACPEE)*, pp. 430–435, Chongqing, China, April 2021.

[8] H. Wei, H. Weng, and M. Zhai, "Research on the application of 5G edge computing technology in the power Internet of things," in *Proceedings of the 2021 IEEE 5th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC)*, pp. 600–605, Xi'an, China, October 2021.

[9] Z. Ning, P. Dong, X. Wang et al., "Distributed and dynamic service placement in pervasive edge computing networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 6, pp. 1277–1292, 2021.

[10] H. Badri, T. Bahreini, D. Grosu, and K. Yang, "Energy-aware application placement in mobile edge computing: a stochastic optimization approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 4, pp. 909–922, 1 April 2020.

[11] A. M. Maia, Y. Ghamri-Doudane, D. Vieira, and M. F. de Castro, "Dynamic service placement and load distribution in edge computing," in *Proceedings of the 2020 16th International Conference on Network and Service Management (CNSM)*, pp. 1–9, Izmir, Turkey, November 2020.

[12] H. Ding, Y. Guo, X. Li, and Y. Fang, "Beef up the edge: spectrum-aware placement of edge computing services for the Internet of things," *IEEE Transactions on Mobile Computing*, vol. 18, no. 12, pp. 2783–2795, 2019.

[13] J. Xiong, X. Chen, Q. Yang, L. Chen, and Z. Yao, "A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2347–2360, 2020.

[14] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, "A blockchain-based machine learning framework for edge services in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, 2022.

[15] D. Wu, J. Li, P. He, Y. Cui, and R. Wang, "Graph-based edge-user collaborative caching with social attributes," in *Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Madrid, Spain, December 2021.

[16] A. Brogi, S. Forti, C. Guerrero, and I. Lera, "Meet genetic algorithms in Monte Carlo: optimised placement of multi-service applications in the fog," in *Proceedings of the 2019 IEEE International Conference on Edge Computing (EDGE)*, pp. 13–17, Milan, Italy, July 2019.

[17] R. Mennes, B. Spinnewyn, S. Latré, and J. F. Botero, "GRECO: a distributed genetic algorithm for reliable application placement in hybrid clouds," in *Proceedings of the 2016 5th IEEE International Conference on Cloud Networking (Cloudnet)*, pp. 14–20, Pisa, Italy, October 2016.

[18] Vmware, *vSphere Single Host Management -VMware Host Client*, Vmware, Palo Alto, CA, USA, 2020, https://docs.vmware.com/cn/VMware-vSphere/5.5/vsphere-html-host-client-12-guide.pdf.

[19] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, and L. Qi, "Trust-oriented IoT service placement for smart cities in edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4084–4091, May 2020.

[20] P. Bellavista, A. Corradi, L. Foschini, and D. Scotece, "Differentiated service/data migration for edge services leveraging container characteristics," *IEEE Access*, vol. 7, Article ID 139746, 2019.

[21] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, May 2019.

[22] L. Wang, L. Jiao, T. He, J. Li, and H. Bal, "Service placement for collaborative edge applications," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 34–47, 2021.

[23] K. Saadallah, V. Gustavo, W. Nannan, X. Wang, and P. Palacharla, "Service placement for real-time applications: rate-adaptation and load-balancing at the network edge," in *Proceedings of the 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 207–215, New York, NY, USA, August 2020.

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

[25] 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.

[26] C. E. F. Caetano, A. B. Lima, J. O. S. Paulino, W. C. Boaventura, I. J. S. Lopes, and E. N. Cardoso, "A conductor arrangement that overcomes the effective length issue in transmission line grounding," *Electric Power Systems Research*, vol. 46, no. 5, pp. 159–162, 2018.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 3572404, 13 pages
https://doi.org/10.1155/2022/3572404

Research Article

Matching Cybersecurity Ontologies on Internet of Everything through Coevolutionary Multiobjective Evolutionary Algorithm

Xingsi Xue[1] and Wenbin Tan[2]

[1] *Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou, Fujian, China*
[2] *School of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan, Shanxi, China*

Correspondence should be addressed to Xingsi Xue; jack8375@gmail.com

Received 14 April 2022; Accepted 11 July 2022; Published 10 August 2022

Academic Editor: Youliang Tian

Since Internet of Everything (IoE) makes all the connections that come online more relevant and valuable, they are subject to numerous security and privacy concerns. Cybersecurity ontology is a shared knowledge model for tackling the security information heterogeneity issue on IoE, which has been widely used in the IoE domain. However, the existing CSOs are developed and maintained independently, yielding the CSO heterogeneity problem. To address this issue, we need to use the similarity measure (SM) to calculate two entities' similarity value in two CSOs and, on this basis, determine the entity correspondences, i.e., CSO alignment. Usually, it is necessary to integrate various SMs to enhance the result's correctness, but how to combine and tune these SMs to improve the alignment's quality is still a challenge. To face this challenge, this work first models CSO matching problem as a Constrained Multiobjective Optimization Problem (CMOOP) and then proposes a Coevolutionary Multiobjective Evolutionary Algorithm (CE-MOEA) to effectively address it. In particular, CE-MOEA uses the multiobjective evolutionary paradigm to avoid the solutions' bias improvement and introduces the coevolutionary mechanism to trade off Pareto Front's (PF's) diversity and convergence. The experiment uses Ontology Alignment Evaluation Initiative's (OAEI's) bibliographic track and conference track and five real CSO matching tasks to test CE-MOEA's performance. Comparisons between OAEI's participants and EA- and MOEA-based matching techniques show that CE-MOEA is able to effectively address various heterogeneous ontology matching problems and determine high-quality CSO alignments.

1. Introduction

Internet of Everything (IoE) is one such technological advancement that represents an interconnected network of people, processes, data, and things. Since IoE makes all the connections that come online more relevant and valuable, they are subject to numerous security and privacy concerns [1]. Cybersecurity ontology is the shared knowledge model for standardizing the security terminologies, setting up the relationship among them, and eliminating semantic differences between different security policies on IoE [2]. Figure 1 shows a fragment of a CSO, where an oval node denotes a concept, such as concept "SecurityPolicy" and "SecurityObject"; the edge connecting two nodes represents the relationship of two concepts; e.g., concept "SecurityToken" is subsumed by concept "SecurityAssertion"; a concept might have properties; e.g., concept "AlternativeType" owns the properties "Capability" and "Requirement."

However, the existing CSOs are developed and maintained independently, yielding the CSO heterogeneity problem. Finding the semantically equivalent entity pairs in two security ontologies, i.e., CSO matching, is an effective solution to this issue. When matching two CSOs, it is necessary to use the similarity measure (SM) to calculate two entities' similarity value. However, no SM can ensure its effectiveness in all contexts, and we usually need to comprehensively aggregate several SMs to improve the results'

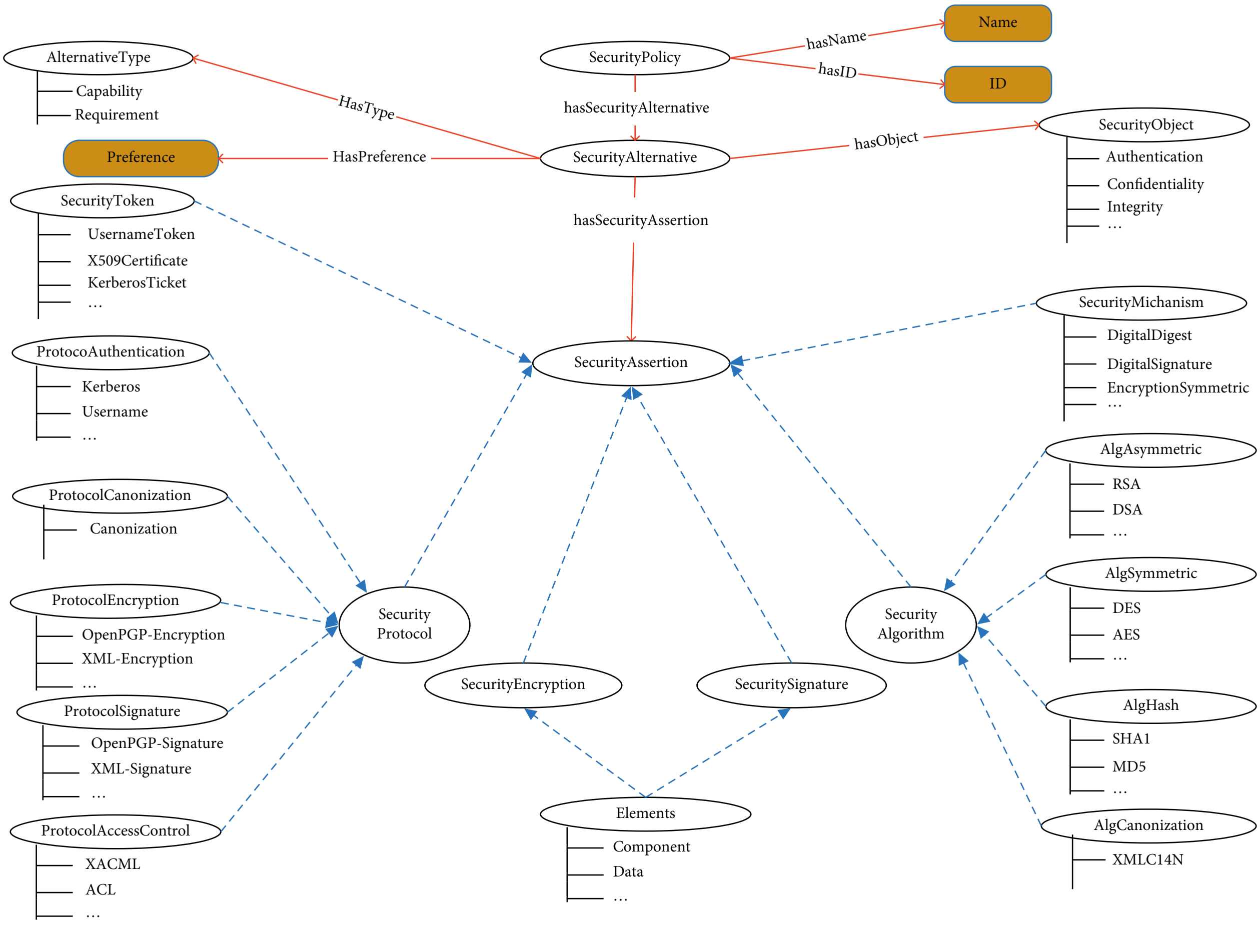


Figure 1: An example of sensor ontology matching.

confidence. In recent years, Evolutionary Algorithm (EA) [3] has become a popular method of optimizing SM's aggregating weights [4, 5], being dedicated to maximizing the alignment's f-measure [6]. According to Xue et al. [7], the single-objective EA tends to improve the solution's quality by improving recall (which measures the alignment's completeness) or precision (which measures the alignment's correctness) while sacrificing the other, yielding solution's bias improvement. To improve the ontology alignment's quality, this work makes the following contributions: (1) A Constrained Multiobjective Optimization Model for the CSO matching problem is constructed, trying to simultaneously optimize the alignment's completeness and correctness. (2) A Coevolutionary Multiobjective Evolutionary Algorithm (CE-MOEA) is proposed to determine the solutions that represent the trade-offs between the alignment's completeness and correctness. In particular, CE-MOEA uses a new paradigm of coevolutionary framework to solve the Constrained Multiobjective Optimization Problem (CMOOP) with the assistance of solving a helper problem. The helper problem is a simpler version of the original MOP, and they are separately addressed by the same multiobjective optimizer. CE-MOEA is characterized by the weak cooperation between two populations, which can be more effective than strong cooperation in existing MOEAs for solving CMOOP [8].

The rest of this paper is organized as follows: after surveying EA-based ontology matching techniques (Section 2), the definition of the cyber ontology matching problem is given (Section 3), and the problem-specific CE-MOEA for addressing this problem is presented (Section 4), followed by the experiment and the corresponding analysis (Section 5). Finally, the conclusion is drawn and future work is presented (Section 6).

2. Related Work

2.1. Evolutionary Algorithm Based Ontology Matching Technique. How to combine and tune different similarity measures to improve the ontology alignment's quality is a challenging problem [9], and EA is a state-of-the-art methodology to face it [10]. Martinez et al. [11] first propose to improve ontology alignment through EA. They are dedicated to finding a suitable weight set for aggregating three kinds of similarity measures in parallel. After that, Ginsca et al. [12] and Naya et al. [13] further optimize another parameter for the matching process, i.e., the threshold for determining the final alignment. The above three works with the objective of maximizing the alignment's quality suffer from two drawbacks: (1) a reference alignment should be provided in advance to evaluate the alignment's quality, but it is not always available in the practical matching task; (2) a bias improvement on the solutions caused by f-measure would bring negative impacts on the results. To overcome these issues, Xue et al. [14] propose the approximate evaluating metrics on alignment's quality and introduce Unanimous Improvement Ratio (UIR) to ensure the solutions' unanimous improvement during algorithm's search process. Their work is able to match more than one pair of heterogeneous ontologies and find the uniform aggregating weights. Later on, Lv et al. [15] not only use the approximate metrics to evaluate the solutions, but also introduce the adaptive selection pressure to improve the algorithm's efficiency. Moreover, the local search strategy and compact encoding mechanism are also combined with EA to improve its searching efficiency [16]. More recently, Lin et al. [17] propose to use EA to aggregate several similarity measures and optimize the alignment's quality. To better trade off the completeness and correctness of the alignment and improve the searching efficiency, Acampora et al. [18] and Xue et al. [19, 20] regard the matching problem as a multiobjective optimizing process and, respectively, used two popular MOEAs, i.e., NSGA-II [21] and MOEA/D [22], to address it. Their approaches aim to find a set of non-dominated solutions that represent a balance between an alignment's completeness and correctness, and the solutions with the best sub-objective values in the Pareto Front (PF) are selected as the output. To improve the algorithm's efficiency, the meta-model is introduced to evaluate the solution's fitness, which can effectively address the expensive evaluating issue [23]. However, the constrained multiobjective CSO matching problem poses stiff challenge to the existing MOEA-based matching techniques, because it is difficult for them to handle both objectives and constraints so as to ensure the solutions' convergence and diversity.

2.2. Coevolutionary Algorithm for Constrained Multiobjective Optimization Problem. For decades, MOEAs have shown their effectiveness in solving Multiobjective Optimization Problem (MOOP) [24], and in recent years, more attention has been drawn to CMOOP, such as collaborative CTAEA [25] and PPS with biphasic search [26]. A CMOOP is formally defined as follows:

$$\begin{cases} \max\ f(x) = (f_1(x), f_2(x)\cdots f_m(x)), \\ s.t. x \in \Omega, \\ g_i(x) \leq 0, i = 1,\cdots p, \\ h_j(x) = 0, j = 1,\cdots, q, \end{cases} \tag{1}$$

where $x = (x_1,\cdots x_D) \in \Omega$ is D-dimensional decision variable; $\Omega \in \mathbb{R}^D$ is the decision space; $f\colon \Omega \rightleftarrows \mathbb{R}^D$ consists of M objectives; and $g_i(x)$ and $h_j(x)$ are, respectively, the inequality constraints and the jth equality constraints. The constraints define a feasible region for CMOOP, and the algorithm should determine the feasible solutions to minimize the objectives as much as possible. Since the constraints and the objectives should be separately handled and balanced, CMOOP should not be regarded as the extension of classical MOOP.

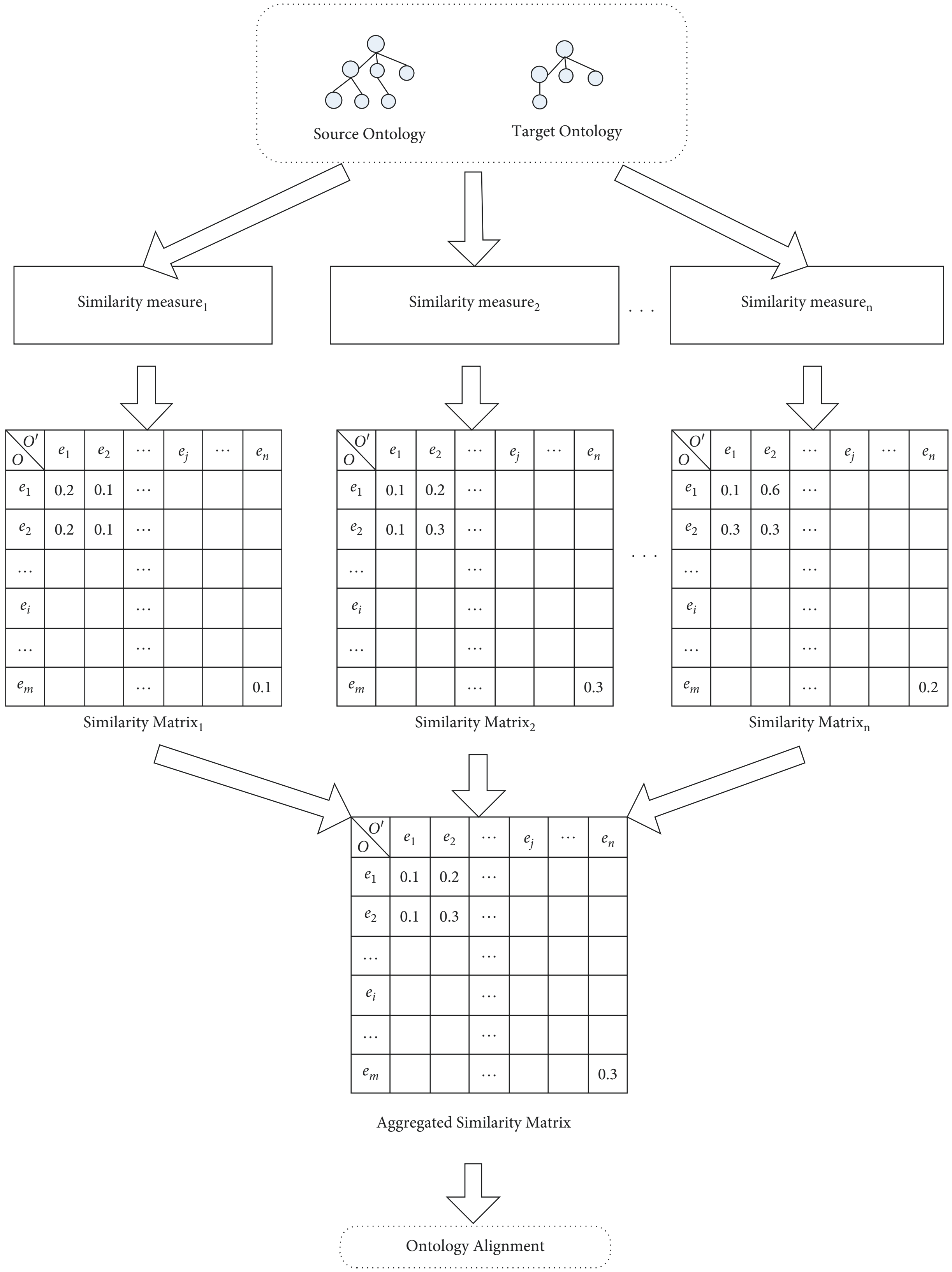
With the development of Coevolutionary Algorithm and its effectiveness on many challenging problems, the coevolutionary constraint handling technique is used in addressing CMOOP. Ceollo [27] and Huang et al. [28], respectively, propose a Coevolutionary EA and Coevolutionary Differential Evolution (DE) Algorithm to address the CMOOP. To balance the constraints and objectives, they assign each subpopulation an independent penalty factor and evolve them simultaneously. Liu et al. [29] propose a coevolutionary framework that consists of two subpopulations. One subpopulation is dedicated to optimizing the objectives without considering the constraints, while the other tries to minimize the violation of constraints. Kieffer et al. [30] first decompose the constraints and assign each constraint to a subpopulation. After that, each subpopulation tries to satisfy more constraints with the requisition that its assigned constraint is met. Wang et al. [31] use M subpopulations to address M constrained single-objective optimization problem, and then find a new subpopulation for solving M-objective CMOOP.

Although CMOOP has been studied for two decades and various techniques have been suggested in the state-of-the-art MOEAs, it is still difficult to address the CMOOP with small feasible region, which might lead to a poor convergence and diversity [32]. In addition, the strong cooperation between subpopulations yields the difficulties of keeping the population's convergence and diversity. To address these issues, we propose a CE-MOEA, which makes use of two subpopulations with weak cooperating framework to address the multiobjective CSO matching problem. CE-MOEA is able to better balance the solutions' convergence and diversity.

3. Cybersecurity Ontology Matching Problem

A CSO consists of the class set, the datatype property set, and the object property set [17], and the existing CSOs can be generally categorized into three categories, i.e., generalized security ontologies, specialized security ontologies, and miscellaneous security ontologies [33]. Due to the human subjectivity, these CSOs might have different ways of class definitions, yielding the ontology heterogeneity problem, which hampers their communications. Ontology matching is dedicated to finding the set of entity correspondences between heterogeneous entities, i.e., ontology alignment. Each entity correspondence consists of two entities, their relationships (typically equivalence ≡) and the confidence that it holds [34]. Figure 2 shows the flowchart of matching two ontologies. Each SM is used to construct a similarity matrix for two ontologies under alignment, whose row and column are the entities of two ontologies, and its element is the similarity value of two corresponding entities. After that, these similarity matrices are aggregated into one matrix, which is then converted into the ontology alignment.

Recall and precision are two classical metrics for evaluating the quality of an alignment [5], which, respectively, measure an alignment's completeness and correctness:



Similarity Matrix$_1$

O' O	e_1	e_2	⋯	e_j	⋯	e_n
e_1	0.2	0.1	⋯			
e_2	0.2	0.1	⋯			
⋯			⋯			
e_i			⋯			
⋯			⋯			
e_m			⋯			0.1

Similarity Matrix$_2$

O' O	e_1	e_2	⋯	e_j	⋯	e_n
e_1	0.1	0.2	⋯			
e_2	0.1	0.3	⋯			
⋯			⋯			
e_i			⋯			
⋯			⋯			
e_m			⋯			0.3

Similarity Matrix$_n$

O' O	e_1	e_2	⋯	e_j	⋯	e_n
e_1	0.1	0.6	⋯			
e_2	0.3	0.3	⋯			
⋯			⋯			
e_i			⋯			
⋯			⋯			
e_m			⋯			0.2

O' O	e_1	e_2	⋯	e_j	⋯	e_n
e_1	0.1	0.2	⋯			
e_2	0.1	0.3	⋯			
⋯			⋯			
e_i			⋯			
⋯			⋯			
e_m			⋯			0.3

Figure 2: The flowchart of matching ontologies.

```
initialize population$_1$;
initialize population$_2$;
evaluate population$_1$ by $f_{\text{origin}}$;
evaluate population$_2$ by $f_{\text{help}}$;
gen = 0;
**While** gen < MaxGen **do**
    parent$_1$ = Random_Selection (population$_1$, $N/2$);
    parent$_2$ = Random_Selection (population$_2$, $N/2$);
    offspring$_1$ = Generate_Offspring (population$_1$, $N/2$);
    offspring$_2$ = Generate_Offspring (population$_2$, $N/2$);
    population$_1$= population$_1$Èoffspring$_1$Èoffspring$_2$;
    population$_2$ = population$_2$Èoffspring$_2$Èoffspring$_1$;
    evaluate population$_1$ by $f_{\text{origin}}$;
    evaluate population$_2$ by $f_{\text{help}}$;
    Non − dorminated_Sorting (population$_1$);
    population$_1$ = Environmental_Selection (population$_1$, $N$);
    population$_2$ = Environmental_Selection (population$_2$, $N$);
    gen = gen+1;
end while
Return population$_1$
```

Algorithm 1: Pseudocode of CE-MOEA.

$$\text{Recall} = \frac{|A \cap RA|}{|RA|},$$
$$\text{Precision}\,(A) = \frac{|A \cap RA|}{|A|}. \quad (2)$$

where $|A|$ and $|RA|$ are, respectively, the numbers of correspondences in the alignment A and the reference alignment RA, and $|A \cap RA|$ is the number of the true positive correspondences in A.

It is difficult to obtain a perfect ontology alignment, whose recall and precision are both equal to 1.00; therefore, we need to balance them during the matching process [34]. Assume n is the number of similarity measures; CSO matching problem f_{origin} is formally defined as follows:

$$\begin{cases} \max f(x) = (f_1(x), f_2(x)), \\ s.t.\ X = (x_1, x_2, \ldots, x_n), x_i \in \{0, 1\}, \\ \sum x_i = 1, \end{cases} \quad (3)$$

where x_i is the aggregating weight of $i-th$*i*th similarity matrix, and (f_1) and (f_2), respectively, calculate the decision variable $X's$ corresponding alignment's recall and precision.

We can also use the statistic-based approach to approximately calculate the recall and precision of an alignment, which are, respectively, defined as follows [35]:

$$\text{Recall}'\,(A) = \frac{\left|\text{Entity}_{\text{Mapped}}\right|}{\left|O_1\right| + \left|O_2\right|},$$
$$\text{Precision}'\,(A) = \frac{\sum_i \text{sim}_i}{|A|}. \quad (4)$$

where $|O_1|$, $|O_2|$, and $|A|$ are, respectively, the entity numbers of ontologies O_1 and O_2, and their alignment A; $|\text{Entity}_{\text{Mapped}}|$ is the number of mapped entities in A; and sim_i is the ith correspondence's similarity value. The motivations behind the metrics (recall') and $(\text{precision}')$ are that the more the mapped entities are, the more the correct mappings found could be (i.e., the higher the recall could be), and the higher the average similarity value is, the higher the confidence of the alignment could be (i.e., the higher the precision could be). On this basis, we define the helper problem f_{help} as follows:

$$\begin{cases} \max f'(X) = (f_1'(X), f_2'(X)), \\ s.t.\ X = (x_1, x_2, \ldots, x_n), x_i \in \{0, 1\}, \\ \sum x_i = 1, \end{cases} \quad (5)$$

where $f_1'(X)$ and $f_2'(X)$, respectively, calculate the decision variable X's corresponding alignment's approximate recall and precision. f_{origin} uses the most sound metric to ensure the population's convergence, while f_{help} is relatively relaxed, and the population for addressing it could be more diverse. The cooperation between two populations, which aim to address f_{origin} and f_{help}, respectively, can bring mutual benefits for them and guide the algorithm to ensure both convergence and diversity of the population.

4. Coevolutionary Multiobjective Evolutionary Algorithm

This work uses the binary encoding mechanism; please see also our previous work [36] for more details. As shown in Algorithm 1, two subpopulations with size N are first randomly initialized and then evaluated by the original problem f_{origin} and helper problem f_{help}. In each generation, two parent sets parent$_1$ and parent$_2$ with size $N/2$ are randomly selected from population$_1$ and population$_2$. Each parent set generates an offspring population with size $N/2$

Table 1: Comparison on OAEI's testing cases in terms of recall and precision. The symbols r and p, respectively, denote recall and precision.

	OAEI's bibliographic track						
Testing case	AMLC [51]	LogMap [52]	LogMapLt [52]	XMap [53]	EA	NSGA-II	CE-MOEA
	r (p)	r (p)	r (p)	r (p)	r (p)	r (p)	r (p)
101	1.00 (1.00)	0.88 (0.96)	0.78 (0.64)	0.93 (1.00)	0.78 (0.84)	1.00 (1.00)	1.00 (1.00)
202	0.80 (0.92)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.72 (0.87)	0.80 (0.91)	0.88 (0.96)
221	0.49 (0.53)	0.87 (0.98)	0.76 (0.69)	0.95 (1.00)	0.87 (0.87)	0.97 (0.92)	1.00 (1.00)
222	0.71 (0.32)	0.00 (0.00)	0.76 (0.69)	0.80 (0.75)	0.78 (0.85)	0.97 (0.92)	1.00 (1.00)
223	0.40 (0.62)	0.90 (0.98)	0.76 (0.69)	0.98 (0.96)	0.87 (0.87)	0.86 (0.95)	1.00 (1.00)
224	0.58 (0.45)	0.90 (0.98)	0.82 (0.98)	0.98 (0.96)	0.94 (0.85)	0.86 (0.95)	1.00 (1.00)
225	0.51 (0.52)	0.92 (0.97)	0.76 (0.69)	0.98 (0.96)	0.78 (0.85)	0.82 (0.87)	1.00 (1.00)
228	1.00 (1.00)	0.92 (0.97)	0.58 (0.40)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
232	0.51 (0.52)	0.87 (0.98)	0.88 (0.93)	0.98 (0.96)	0.81 (0.95)	0.86 (0.95)	1.00 (1.00)
233	1.00 (1.00)	0.92 (0.97)	0.58 (0.40)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
236	1.00 (1.00)	0.92 (0.97)	0.72 (0.87)	1.00 (1.00)	0.82 (0.88)	0.92 (0.92)	1.00 (1.00)
237	0.42 (0.58)	0.00 (0.00)	0.88 (0.93)	0.80 (0.75)	0.87 (0.80)	0.82 (0.87)	0.94 (0.94)
238	0.51 (0.52)	0.96 (0.93)	0.88 (0.93)	0.98 (0.96)	0.82 (0.92)	0.86 (0.95)	1.00 (1.00)
239	1.00 (1.00)	0.91 (0.93)	0.58 (0.40)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
240	1.00 (1.00)	0.91 (0.93)	0.58 (0.40)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
241	1.00 (1.00)	0.91 (0.93)	0.72 (0.87)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
246	1.00 (1.00)	0.88 (0.96)	0.72 (0.87)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
247	1.00 (1.00)	0.88 (0.96)	0.72 (0.87)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)
Average	0.77 (0.77)	0.75 (0.80)	0.69 (0.68)	0.91 (0.90)	0.89 (0.91)	0.93 (0.95)	0.99 (0.99)

	OAEI's conference track						
Testing case	AMLC	LogMap	LogMapLt	XMap	EA	NSGA-II	CE-MOEA
cmt-conference	0.53 (0.67)	0.53 (0.73)	0.33 (0.56)	0.00 (0.00)	0.68 (0.76)	0.68 (0.76)	0.73 (0.92)
cmt-confOf	0.56 (0.90)	0.31 (0.83)	0.38 (0.67)	0.44 (0.88)	0.65 (0.68)	0.65 (0.68)	0.75 (0.75)
cmt-edas	0.77 (0.91)	0.62 (0.89)	0.62 (0.73)	0.69 (0.75)	0.65 (0.72)	0.68 (0.76)	0.82 (0.95)
cmt-ekaw	0.55 (0.75)	0.55 (0.75)	0.45 (0.56)	0.64 (0.70)	0.65 (0.68)	0.65 (0.68)	0.80 (0.77)
cmt-iasted	1.00 (0.80)	0.84 (0.80)	0.90 (0.89)	0.93 (0.80)	0.75 (0.89)	0.83 (0.87)	0.88 (0.93)
cmt-sigkdd	0.92 (0.92)	0.88 (0.95)	0.67 (0.89)	0.83 (0.91)	0.75 (0.75)	0.87 (0.90)	0.92 (0.94)
conference-confOf	0.87 (0.87)	0.73 (0.85)	0.60 (0.90)	0.80 (0.71)	0.78 (0.67)	0.80 (0.88)	0.88 (0.93)
conference-edas	0.65 (0.73)	0.65 (0.85)	0.53 (0.75)	0.65 (0.79)	0.78 (0.67)	0.68 (0.78)	0.79 (0.79)
conference-ekaw	0.72 (0.78)	0.48 (0.60)	0.32 (0.62)	0.60 (0.58)	0.74 (0.66)	0.70 (0.78)	0.79 (0.85)
conference-iasted	0.36 (0.83)	0.50 (0.88)	0.29 (0.80)	0.36 (0.62)	0.68 (0.52)	0.75 (0.60)	0.75 (0.75)
conference-sigkdd	0.73 (0.85)	0.73 (0.85)	0.53 (0.80)	0.60 (0.58)	0.75 (0.75)	0.75 (0.75)	0.78 (0.82)
confOf-edas	0.58 (0.92)	0.53 (0.77)	0.58 (0.58)	0.53 (0.91)	0.65 (0.72)	0.65 (0.72)	0.71 (0.79)
confOf-ekaw	0.80 (0.94)	0.70 (0.93)	0.50 (0.77)	0.80 (0.76)	0.88 (0.75)	0.88 (0.75)	0.88 (0.75)
confOf-iasted	0.44 (0.80)	0.54 (0.89)	0.54 (0.90)	0.67 (0.43)	0.62 (0.51)	0.69 (0.51)	0.71 (0.78)
confOf-sigkdd	0.88 (0.95)	0.81 (0.90)	0.68 (0.88)	0.57 (0.80)	0.88 (0.73)	0.89 (0.78)	0.88 (0.96)
edas-ekaw	0.48 (0.79)	0.52 (0.75)	0.43 (0.59)	0.52 (0.75)	0.65 (0.68)	0.65 (0.68)	0.65 (0.79)
edas-iasted	0.47 (0.82)	0.37 (0.88)	0.37 (0.88)	0.42 (0.57)	0.63 (0.57)	0.62 (0.82)	0.65 (0.88)
edas-sigkdd	0.75 (0.84)	0.47 (0.88)	0.47 (0.88)	0.62 (0.81)	0.75 (0.75)	0.68 (0.76)	0.76 (0.88)
ekaw-iasted	0.70 (0.84)	0.70 (0.78)	0.60 (0.60)	0.70 (0.58)	0.68 (0.76)	0.82 (0.74)	0.85 (0.85)
ekaw-sigkdd	0.73 (0.80)	0.70 (0.78)	0.70 (0.78)	0.64 (0.78)	0.78 (0.67)	0.70 (0.81)	0.78 (0.83)
iasted-sigkdd	0.87 (0.81)	0.88 (0.82)	0.73 (0.73)	0.87 (0.68)	0.76 (0.75)	0.80 (0.85)	0.87 (0.89)
Average	0.68 (0.83)	0.62 (0.82)	0.53 (0.75)	0.61 (0.68)	0.70 (0.69)	0.73 (0.55)	0.79 (0.84)

with the single-point crossover operator and flip-bit mutation [37]. Afterwards, population$_1$ and population$_2$ are both combined with two offspring populations, offspring$_1$ and offspring$_2$, which are, respectively, evaluated by f_{origin} and f_{help}. Finally, we execute NSGA-II's non-dominated sorting and environmental selection on population$_1$ and population$_2$. When the generation gen reaches the maximum generation MaxGen, the algorithm terminates and returns population$_1$ as the output.

CE-MOEA always evaluates population$_1$ by f_{origin} and evolves population$_2$ to solve f_{help}, and since f_{help} is a simplified version of f_{origin}, the evaluation by f_{help} does not increase the algorithm's computational complexity. f_{help} is simpler, and thus populations$_2$ usually converges quickly and has better diversity. f_{help} assists in solving f_{origin} by sharing its offspring, which is able to improve population$_1'$s converging speed and helps it jump out of the local optimums. Different from other MOEAs which make subpopulations cooperate in the whole evolving process, CE-MOEA evolves two subpopulations separately except for sharing their offspring in each generation. CE-MOEA uses a weak cooperation to offer each subpopulation freedom to evolve and makes one subpopulation to assist the other to address the original optimization problem. According to Tian et al. [7], the coevolutionary paradigm with weak cooperation is more effective than a strong cooperation.

Table 2: Comparison on OAEI's testing cases in terms of f-measure.

			OAEI's bibliographic track				
Testing case	AMLC	LogMap	LogMapLt	XMap	EA	NSGA-II	CE-MOEA
101	1.00	0.95	0.71	0.97	0.81	1.00	1.00
202	0.86	0.00	0.00	0.00	0.79	0.85	0.92
221	0.51	0.94	0.72	0.97	0.87	0.95	1.00
222	0.50	0.00	0.72	0.78	0.82	0.95	1.00
223	0.51	0.94	0.72	0.97	0.87	0.90	1.00
224	0.51	0.94	0.90	0.97	0.90	0.90	1.00
225	0.51	0.95	0.72	0.97	0.82	0.85	1.00
228	1.00	0.92	0.48	1.00	1.00	1.00	1.00
232	0.51	0.94	0.90	0.97	0.88	0.90	1.00
233	1.00	0.92	0.48	1.00	1.00	1.00	1.00
236	1.00	0.92	0.80	1.00	0.85	0.92	1.00
237	0.50	0.00	0.91	0.78	0.84	0.85	0.94
238	0.51	0.95	0.90	0.97	0.87	0.90	1.00
239	1.00	0.92	0.48	1.00	1.00	1.00	1.00
240	1.00	0.92	0.48	1.00	1.00	1.00	1.00
241	1.00	0.92	0.80	1.00	1.00	1.00	1.00
246	1.00	0.92	0.80	1.00	1.00	1.00	1.00
247	1.00	0.92	0.80	1.00	1.00	1.00	1.00
Average	0.77	0.78	0.68	0.91	0.91	0.94	0.99
			OAEI's conference track				
cmt-conference	0.59	0.62	0.42	0.00	0.72	0.72	0.84
cmt-confOf	0.69	0.45	0.48	0.58	0.66	0.66	0.75
cmt-edas	0.83	0.73	0.67	0.72	0.68	0.72	0.88
cmt-ekaw	0.63	0.63	0.50	0.67	0.68	0.72	0.79
cmt-iasted	0.89	0.89	0.89	0.89	0.82	0.85	0.90
cmt-sigkdd	0.92	0.91	0.76	0.87	0.75	0.89	0.93
conference-confOf	0.87	0.79	0.72	0.75	0.78	0.83	0.90
conference-edas	0.69	0.73	0.62	0.71	0.73	0.73	0.79
conference-ekaw	0.75	0.53	0.42	0.59	0.70	0.74	0.82
conference-iasted	0.50	0.64	0.42	0.45	0.59	0.68	0.75
conference-sigkdd	0.79	0.79	0.64	0.69	0.75	0.75	0.80
confOf-edas	0.71	0.62	0.58	0.67	0.68	0.68	0.75
confOf-ekaw	0.86	0.80	0.61	0.78	0.82	0.82	0.88
confOf-iasted	0.57	0.62	0.62	0.52	0.58	0.60	0.75
confOf-sigkdd	0.92	0.83	0.73	0.67	0.80	0.85	0.92
edas-ekaw	0.59	0.62	0.50	0.62	0.66	0.62	0.72
edas-iasted	0.60	0.52	0.52	0.48	0.60	0.66	0.77
edas-sigkdd	0.80	0.61	0.61	0.64	0.75	0.72	0.82
ekaw-iasted	0.78	0.74	0.60	0.64	0.72	0.78	0.85
ekaw-sigkdd	0.76	0.74	0.74	0.70	0.73	0.76	0.81
iasted-sigkdd	0.84	0.85	0.73	0.76	0.76	0.82	0.88
Average	0.74	0.70	0.61	0.64	0.71	0.74	0.82

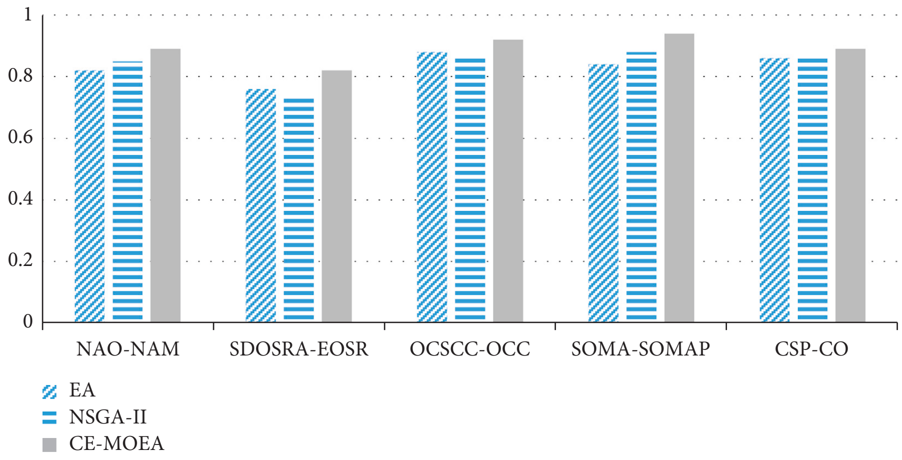


Figure 3: Comparison on cybersecurity ontology matching tasks in terms of recall.

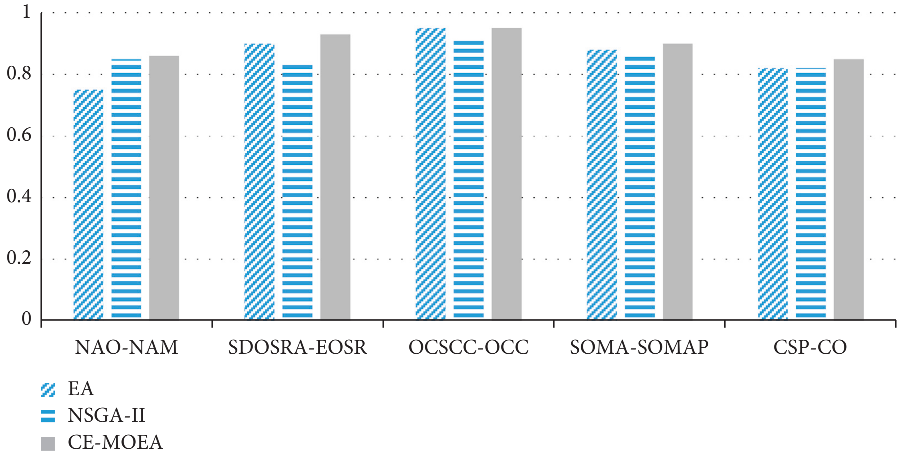


Figure 4: Comparison on cybersecurity ontology matching tasks in terms of precision.

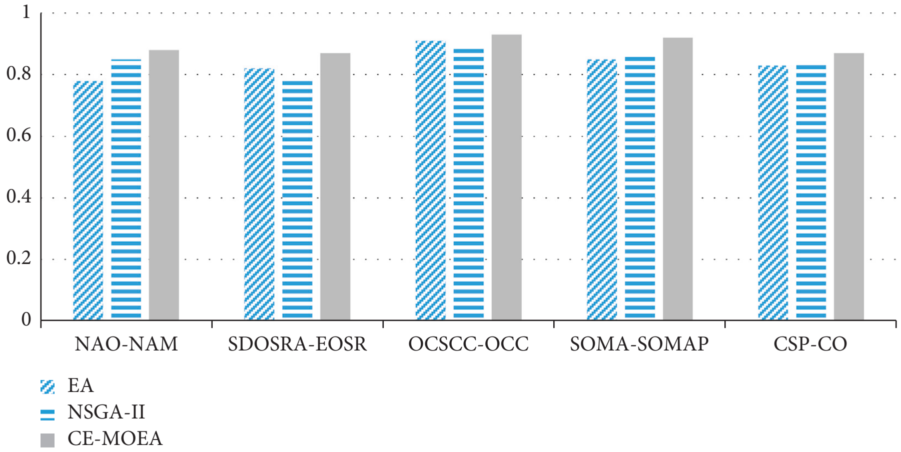


Figure 5: Comparison on cybersecurity ontology matching tasks in terms of f-measure.

5. Experiment

5.1. Experimental Configuration. In the experiment, we first compare our approach with EA-based matching technique [17], NSGA-II-based matching technique [19], and OAEI's participants on bibliographic track and conference track provided by Ontology Alignment Evaluation Initiative (OAEI). In particular, OAEI's bibliographic track requires matching two bibliographic ontologies, and the target ontology's entity names could be random strings or synonyms. The hierarchy could be expanded or flattened, the properties could be suppressed, and the classes could be refined by several subclasses or flattened. OAEI's conference track requires matching 16 different ontologies on the conference organization, which have been used in some actual conference series and the corresponding conference web sites. After that, we compare CE-MOEA with EA-based and NSGA-II-based matching techniques on five pairs of real CSOs, which are all popular ontologies in the cybersecurity domain and own large quantities of heterogeneous entities:

(1) Network Security Ontologies: Network Attack Ontology (NAO) [38] and Ontology-based Attack Model (NAM) [39].

(2) Security Requirement-related Ontologies: Security and Domain Ontology for Security Requirement Analysis (SDOSRA) [40] and Extended Ontology for Security Requirements (EOSR) [41].

(3) Miscellaneous Security Ontologies: Ontological approach toward Cybersecurity in Cloud Computing (OCSCC) [42] and Ontology in Cloud Computing (OCC) [43].

(4) Application-Based Security Ontologies: Security Ontology for Mobile Applications (SOMA) [44] and Security Ontology for Mobile Agents Protection (SOMAP) [45].

(5) Cloud Security Ontologies: Cloud Security Policy (CSP) [46] and Cloud Ontology (CO) [47].

Finally, we carry out the *T*-test to statistically compare three EA-based matching techniques. In particular, the configurations of EA and NSGA-II are referred to in their papers, and the configuration of CE-MOEA is as follows:

(1) Population size = 20.

(2) Maximum generation = 2000.

(3) Crossover rate = 0.65.

Table 3: Comparisons between EA, NSGA-II, and CE-MOEA in terms of mean f-measure and standard deviation.

	OAEI's bibliographic track		
Testing case	EA f-measure (stdDev)	NSGA-II f-measure (stdDev)	CE-MOEA f-measure (stdDev)
101	0.81 (0.01)	1.00 (0.01)	1.00 (0.01)
202	0.79 (0.03)	0.85 (0.02)	0.92 (0.02)
221	0.87 (0.02)	0.95 (0.02)	1.00 (0.01)
222	0.82 (0.02)	0.95 (0.01)	1.00 (0.01)
223	0.87 (0.01)	0.90 (0.02)	1.00 (0.01)
224	0.90 (0.01)	0.90 (0.02)	1.00 (0.01)
225	0.82 (0.02)	0.85 (0.01)	1.00 (0.01)
228	1.00 (0.01)	1.00 (0.01)	1.00 (0.01)
232	0.88 (0.01)	0.90 (0.01)	1.00 (0.01)
233	1.00 (0.01)	1.00 (0.01)	1.00 (0.01)
236	0.85 (0.03)	0.92 (0.02)	1.00 (0.01)
237	0.84 (0.02)	0.85 (0.03)	0.94 (0.01)
238	0.87 (0.03)	0.90 (0.01)	1.00 (0.01)
239	1.00 (0.01)	1.00 (0.01)	1.00 (0.01)
240	1.00 (0.01)	1.00 (0.01)	1.00 (0.01)
241	1.00 (0.01)	1.00 (0.01)	1.00 (0.01)
246	1.00 (0.01)	1.00 (0.01)	1.00 (0.01)
247	1.00 (0.01)	1.00 (0.01)	1.00 (0.01)
	OAEI's conference track		
cmt-conference	0.72 (0.02)	0.72 (0.02)	0.84 (0.02)
cmt-confOf	0.66 (0.03)	0.66 (0.02)	0.75 (0.01)
cmt-edas	0.68 (0.02)	0.72 (0.01)	0.88 (0.01)
cmt-ekaw	0.68 (0.02)	0.72 (0.01)	0.79 (0.02)
cmt-iasted	0.82 (0.02)	0.85 (0.02)	0.90 (0.01)
cmt-sigkdd	0.75 (0.02)	0.89 (0.01)	0.93 (0.01)
conference-confOf	0.78 (0.02)	0.83 (0.02)	0.90 (0.01)
conference-edas	0.73 (0.03)	0.73 (0.02)	0.79 (0.02)
conference-ekaw	0.70 (0.02)	0.74 (0.01)	0.82 (0.01)
conference-iasted	0.59 (0.03)	0.68 (0.01)	0.75 (0.01)
conference-sigkdd	0.75 (0.03)	0.75 (0.01)	0.80 (0.01)
confOf-edas	0.68 (0.02)	0.68 (0.01)	0.75 (0.01)
confOf-ekaw	0.82 (0.02)	0.82 (0.02)	0.88 (0.01)
confOf-iasted	0.58 (0.03)	0.60 (0.01)	0.75 (0.01)
confOf-sigkdd	0.80 (0.01)	0.85 (0.01)	0.92 (0.01)
edas-ekaw	0.66 (0.03)	0.62 (0.03)	0.72 (0.02)
edas-iasted	0.60 (0.02)	0.66 (0.02)	0.77 (0.01)
edas-sigkdd	0.75 (0.03)	0.72 (0.03)	0.82 (0.02)
ekaw-iasted	0.72 (0.01)	0.78 (0.02)	0.85 (0.01)
ekaw-sigkdd	0.73 (0.02)	0.76 (0.01)	0.81 (0.02)
iasted-sigkdd	0.76 (0.01)	0.82 (0.01)	0.88 (0.01)
	Cybersecurity ontology matching tasks		
NAO-NAM	0.78 (0.02)	0.85 (0.02)	0.88 (0.02)
SDOSRA-EOSR	0.82 (0.02)	0.78 (0.02)	0.87 (0.01)
OCSCC-OCC	0.91 (0.02)	0.89 (0.02)	0.93 (0.01)
SOMA-SOMAP	0.85 (0.02)	0.87 (0.01)	0.92 (0.02)
CSP-CO	0.83 (0.01)	0.84 (0.01)	0.87 (0.01)

(4) Mutation rate = 0.012.

Three categories of similarity measures used by CE-MOEA are as follows:

(1) Syntax-based similarity measure: Levenshtein distance [48].

(2) Linguistic-based similarity measure: WordNet-based distance [49].

(3) Taxonomy-based similarity measure: context-based distance [50].

The algorithm's configurations are determined through the empirical experiments, and their robustness against different heterogeneous matching tasks is verified through the experimental results. Three similarity measures are the classical ones that belong to three categories of similarity measures in ontology matching domains, which have been proved to have mutual benefits in enhancing the results' confidence [11].

5.2. Experimental Results. Tables 1 and 2 make comparisons on OAEI's testing cases in terms of recall, precision, and

Table 4: *T*-test on alignment's quality.

Testing case	(EA, CE-MOEA) *T* value (*p* value)	(NSGA-II, CE-MOEA) *T* value (*p* value)
OAEI's bibliographic track		
101	−73.5866 (0.0000)	0.0000 (0.5000)
202	−19.7484 (0.0012)	−13.5554 (0.0027)
221	−31.8433 (0.0004)	−12.2474 (0.0033)
222	−44.0908 (0.0002)	−19.3649 (0.0013)
223	−50.3487 (0.0001)	−24.4949 (0.0008)
224	−38.7298 (0.0003)	−24.4949 (0.0008)
225	−44.0908 (0.0002)	−58.0947 (0.0001)
228	0.0000 (0.5000)	0.0000 (0.5000)
232	−46.4758 (0.0002)	−38.7298 (0.0003)
233	0.0000 (0.5000)	0.0000 (0.5000)
236	−25.9807 (0.0007)	−19.59592 (0.0012)
237	−24.4948 (0.0008)	−15.5884 (0.0020)
238	−22.5166 (0.0009)	−38.7298 (0.0003)
239	0.0000 (0.5000)	0.0000 (0.5000)
240	0.0000 (0.5000)	0.0000 (0.5000)
241	0.0000 (0.5000)	0.0000 (0.5000)
246	0.0000 (0.5000)	0.0000 (0.5000)
247	0.0000 (0.5000)	0.0000 (0.5000)
OAEI's conference track		
cmt-conference	−23.2379 (0.0009)	−23.2379 (0.0009)
cmt-confOf	−15.5884 (0.0020)	−22.0454 (0.0010)
cmt-edas	−48.9897 (0.0002)	−61.9677 (0.0001)
cmt-ekaw	−21.3014 (0.0010)	−17.1464 (0.0016)
cmt-iasted	−19.5959 (0.0012)	−12.2474 (0.0033)
cmt-sigkdd	−44.0908 (0.0002)	−15.4919 (0.0020)
conference-confOf	−29.3938 (0.0005)	−17.1464 (0.0016)
conference-edas	−9.1146 (0.0059)	−11.6189 (0.0036)
conference-ekaw	−29.3938 (0.0005)	−30.9838 (0.0005)
conference-iasted	−27.7128 (0.0006)	−27.1108 (0.0006)
conference-sigkdd	−8.6602 (0.0065)	−19.3649 (0.0013)
confOf-edas	−17.1464 (0.0016)	−27.1108 (0.0006)
confOf-ekaw	−14.6969 (0.0023)	−14.6969 (0.0023)
confOf-iasted	−29.4448 (0.0005)	−58.0947 (0.0001)
confOf-sigkdd	−46.4758 (0.0002)	−27.1108 (0.0006)
edas-ekaw	−9.1146 (0.0059)	−15.1910 (0.0021)
edas-iasted	−41.6413 (0.0002)	−26.94 (0.0006)
edas-sigkdd	−10.6337 (0.0043)	−15.1910 (0.0021)
ekaw-iasted	−50.3487 (0.0001)	−17.1464 (0.0016)
ekaw-sigkdd	−15.4919 (0.0020)	−12.2474 (0.0033)
iasted-sigkdd	−46.4758 (0.0002)	−23.2379 (0.0009)
Cybersecurity ontology matching tasks		
NAO-NAM	−19.3649 (0.0013)	−5.8094 (0.0141)
SDOSRA-EOSR	−12.2474 (0.0033)	−22.0454 (0.0010)
OCSCC-OCC	−4.8989 (0.0196)	−9.7979 (0.0051)
SOMA-SOMAP	−13.5554 (0.0027)	−12.2474 (0.0033)
CSP-CO	−15.4919 (0.0020)	−11.6189 (0.0036)

f-measure. In particular, f-measure is a uniform mean of recall and precision. Figures 3, 4, and 5 respectively compare EA, NSGA-II, and CE-MOEA on CSO matching tasks. Table 3 compares CE-MOEA with EA and NSGA-II with the mean f-measure and the corresponding standard deviation stdDev, and in Table 4, the statistical *T*-test [51] is executed on the data presented in Table 3. The results of EA, NSGA-II, and CE-MOEA presented in the tables and figures are the mean values of 30 independent runs.

As shown in Tables 1 and 2, compared with OAEI's participants, EA-, NSGA-II-, and CE-MOEA-based matching techniques comprehensively take into consideration several similarity measures, whose precision values are generally high. In addition, the iterative refinement on the alignment is an effective way of finding more correct entity correspondences; therefore, EA-based matching techniques' recall values are also high in general.

In Figures 3, 4, and 5, since MOEA is able to better trade off the alignment's recall and precision, NSGA-II and CE-MOEA's results are better than those of classical EA. With the introduction of the coevolutionary mechanism, CE-MOEA is able to further improve the results' quality by helping the algorithm jump out of the local optimum. In particular, the subpopulation for the helper problem can improve the diversity in general, while the subpopulation for the original problem ensures the algorithm's convergence. The cooperation between them is able to better trade off the PF's diversity and convergence and further improve the alignment's quality.

In Table 4, T-test's degree of freedom of is 2, and the significant level is 0.05. On all testing cases, the p values are all smaller than 0.05, and thus, we can draw the conclusion that CE-MOEA statistically outperforms EA- and NSGA-II-based matching techniques at the significance level of 5%. To conclude, CE-MOEA-based ontology matching technique is able to effectively address various ontology heterogeneity problems and determine high-quality CSO alignments.

6. Conclusion and Future Work

Due to the distributed and independent nature of cybersecurity systems, it is necessary to match various heterogeneous CSOs to manage cybersecurity knowledge on IoE. To this end, this work proposes a CE-MOEA-based matching technique to effectively determine CSO alignment. CE-MOEA uses the multiobjective evolutionary paradigm to avoid the solutions' bias improvement and introduces the coevolutionary mechanism to trade off PF's diversity and convergence. The experiment uses OAEI's bibliographic track and conference track and five real CSO matching tasks to test CE-MOEA's performance. Comparisons between OAEI's participants and EA- and CE-MOEA-based matching techniques show that our proposed algorithm is able to effectively address various heterogeneous ontology matching problems and determine high-quality cybersecurity ontology alignments. The experimental results also show that the evolutionary paradigm is able to find better alignment than other artificial techniques and the weak cooperating framework is effective in further improving MOEA's performance.

Although CE-MOEA-based aligning technique shows its superiority in the experiment, it is not able to detect the *m:n* correspondence; i.e., multiple source entities are mapped with multiple target entities, which is a common complex correspondence pattern. In addition, CE-MOEA is also not able to find other semantic relationships among the entities, such as the subsumption. The divide-and-conquer approach has been proved to be a viable method that can facilitate the effectiveness of matching process [52], and we are also interested in utilizing the clustering algorithm, such as graph clustering algorithm [53], to partition two CSOs, which can be of help to improve the efficiency of matching process [54–56].

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 62172095), the Natural Science Foundation of Fujian Province (nos. 2020J01875 and 2019J01771), and the Scientific Research Foundation of Fujian University of Technology (no. GY-Z17162).

References

[1] Y. Tian, Ta Li, J. Xiong, M. Bhuiyan, J. Ma, and C. Peng, "A blockchain-based machine learning framework for edge services in IIoT," in *Proceedings of the IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, March 2022.

[2] G. Denker, L. Kagal, and Tim Finin, "Security in the semantic web using owl," *Information Security Technical Report*, vol. 10, no. 1, pp. 51–58, 2005.

[3] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks*, pp. 43–55, Springer, Berlin, German, 2019.

[4] G. Acampora, V. Loia, and A. Vitiello, "Enhancing ontology alignment through a memetic aggregation of similarity measures," *Information Sciences*, vol. 250, pp. 1–20, 2013.

[5] X. Xue, J. Lu, and J. Chen, "Using nsga-iii for optimising biomedical ontology alignment," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 3, pp. 135–141, 2019.

[6] C. J. Van Rijsberge, *Information Retrieval*, University of Glasgow, Butterworth, London, 1975.

[7] X. Xue and Y. Wang, "Using memetic algorithm for instance coreference resolution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 580–591, 2016.

[8] Ye Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, "A coevolutionary framework for constrained multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 102–116, 2021.

[9] P. Shvaiko and J.. Euzenat, "Ontology matching: state of the art and future challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 158–176, 2013.

[10] G. Acampora, V. Loia, S. Salerno, and A. Vitiello, "A hybrid evolutionary approach for solving the ontology alignment problem," *International Journal of Intelligent Systems*, vol. 27, no. 3, pp. 189–216, 2012.

[11] J. Martinez-Gil, E. Alba, and J. F. Aldana-Montes, "Optimizing ontology alignments by using genetic algorithms," in *Proceedings of the Workshop on Nature Based Reasoning for the Semantic Web*, pp. 1–15, Karlsruhe, Germany, October 2008.

[12] A.-L. Ginsca and I. Adrian, "Using a genetic algorithm for optimizing the similarity aggregation step in the process of ontology alignment," in *Proceedings of the 9th Roedunet International Conference*, pp. 118–122, Sibiu, Romania, June 2010.

[13] M. José, M. Marcos, P. Javier, R. Munteanu, and A. Pazos Sierra, "Improving ontology alignment through genetic algorithms," *Soft Computing Methods for Practical Environment Solutions: Techniques and Studies*, pp. 240–259, 2010.

[14] X. Xue and Y. Wang, "Optimizing ontology alignments through a memetic algorithm using both matchfmeasure and unanimous improvement ratio," *Artificial Intelligence*, vol. 223, pp. 65–81, 2015.

[15] Q. Lv, C. Jiang, and H. Li, "Solving ontology meta-matching problem through an evolutionary algorithm with

approximate evaluation indicators and adaptive selection pressure," *IEEE Access*, vol. 9, pp. 3046–3064, 2021.

[16] X. Xue and J. Chen, "Using compact evolutionary tabu search algorithm for matching sensor ontologies," *Swarm and Evolutionary Computation*, vol. 48, pp. 25–30, 2019.

[17] W. Lin and R. Haga, "Matching cyber security ontologies through genetic algorithm-based ontology alignment technique," *Security and Communication Networks*, pp. 1–7, 2021.

[18] G. Acampora, H. Ishibuchi, and A. Vitiello, "A comparison of multi-objective evolutionary algorithms for the ontology meta matching problem," in *Proceedings of the 2014 IEEE congress on Evolutionary Computation (CEC)*, pp. 413–420, IEEE, Beijing, China, July 2014.

[19] X. Xue, "Complex ontology alignment for autonomous systems via the compact Co-evolutionary brain storm optimization algorithm," *ISA Transactions*, pp. 1–9, 2022.

[20] X. Xue, Y. Wang, and W. Hao, "Using moea/d for optimizing ontology alignments," *Soft Computing*, vol. 18, no. 8, pp. 1589–1601, 2014.

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

[22] Q. Zhang and H. Li, "Moea/d: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[23] X. Xue and Q. Huang, "Generative adversarial learning for optimizing ontology alignment," *Expert Systems*, pp. 1–12, 2022.

[24] A. Zhou, Bo-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: a survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[25] Ke Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2019.

[26] Z. Fan, W. Li, X. Cai et al., "Push and pull search for solving constrained multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 44, pp. 665–679, 2019.

[27] C. A. Coello Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.

[28] Fu-zhuo Huang, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Applied Mathematics and Computation*, vol. 186, no. 1, pp. 340–356, 2007.

[29] Bo Liu, H. Ma, X. Zhang, and Y. Zhou, "A memetic coevolutionary differential evolution algorithm for constrained optimization," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, pp. 2996–3002, IEEE, Singapore, September 2007.

[30] E. Kieffer, G. Danoy, B. Pascal, and A. Nagih, "A new co-evolutionary algorithm based on constraint decomposition," in *Proceedings of the 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 492–500, IEEE, Lake Buena Vista, FL, USA, June 2017.

[31] J. Wang, G. Liang, and J. Zhang, "Cooperative differential evolution framework for constrained multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2060–2072, 2019.

[32] Z. Ma and Y. Wang, "Evolutionary constrained multiobjective optimization: test suite construction and performance comparisons," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 972–986, 2019.

[33] V. Singh and S. K. Pandey, "Cloud security ontology (cso)," in *Cloud Computing for Geospatial Big Data Analytics*pp. 81–109, Springer, 2019.

[34] X. Xue and P.-W. Tsai, "Integrating Energy Smart Grid's ontologies through multi-objective particle swarm optimization algorithm with competitive mechanism," *Sustainable Energy Technologies and Assessments*, vol. 53, Article ID 102442, 2022.

[35] J.. Bock and J. Hettenhausen, "Discrete particle swarm optimisation for ontology alignment," *Information Sciences*, pp. 152–173, 192.

[36] X. Xue, W. Liu, and A. Ren, "Integrating heterogeneous ontologies in asian languages through compact genetic algorithm with annealing Re-sample inheritance mechanism," *ACM Transactions on Asian and Low-Resource Language Information Processing*, pp. 1–12, 2022.

[37] X. Xue and Y. Wang, "Ontology alignment based on instance using nsga-ii," *Journal of Information Science*, vol. 41, no. 1, pp. 58–70, 2015.

[38] P. Van Heerden, B. Irwin, and I. Burke, "Classifying network attack scenarios using an ontology," in *Proceedings of the 7th International Conference on Information-Warfare & Security (ICIW 2012)*, Academic Conferences and Publishing International Limited, Seattle, USA, pp. 311–324, 2012.

[39] J.-bo Gao, B.-wen Zhang, X.-hua Chen, and Z. Luo, "Ontology-based model of network and computer attacks for security assessment," *Journal of Shanghai Jiaotong University*, vol. 18, no. 5, pp. 554–562, 2013.

[40] S. Amina, C. Salinesi, I. Wattiau, and H. Mouratidis, "Using security and domain ontologies for security requirements analysis," in *Proceedings of the 2013 IEEE 37th Annual Computer Software and Applications Conference Workshops*, pp. 101–107, IEEE, Japan, July 2013.

[41] F. Massacci, J. Mylopoulos, F. Paci, T. Thun Tun, and Y. Yu, "An extended ontology for security requirements," in *International Conference on Advanced Information Systems Engineering*, pp. 622–636, Springer, Berlin, Germany, 2011.

[42] T. Takahashi, Y. Kadobayashi, and H. Fujiwara, "Ontological approach toward cybersecurity in cloud computing," in *Proceedings of the 3rd International Conference on Security of Information and Networks*, pp. 100–109, New YorkNYUnited States, September 2010.

[43] L. Youseff, M. Butrico, and D. Da Silva, "Toward a unified ontology of cloud computing," in *Grid Computing Environments Workshop*, pp. 1–10, IEEE, Austin, TX, USA, 2008.

[44] S. Beji and N. El Kadhi, "Security ontology proposal for mobile applications," in *Proceedings of the 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pp. 580–587, IEEE, Washington DC, USA, May 2009.

[45] H. Razouki, "Security policy modelling in the mobile agent system," *International Journal of Computer Network and Information Security*, vol. 11, no. 10, pp. 26–36, 2019.

[46] C. Choi, J. Choi, and P. Kim, "Ontology-based access control model for security policy reasoning in cloud computing," *The Journal of Supercomputing*, vol. 67, no. 3, pp. 711–722, 2014.

[47] A. Herzog, N. Shahmehri, and C. Duma, "An ontology of information security," *International Journal of Information Security and Privacy*, vol. 1, no. 4, pp. 1–23, 2007.

[48] I. Vladimir, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics - Doklady*, vol. 10, pp. 707–710, 1966.

[49] G. A. Miller, "Wordnet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[50] X. Xue and J. Zhang, "Matching large-scale biomedical ontologies with central concept based partitioning algorithm and adaptive compact evolutionary algorithm," *Applied Soft Computing*, vol. 106, Article ID 107343.

[51] P. Grzegorzewski, "Testing statistical hypotheses with vague data," *Fuzzy Sets and Systems*, vol. 112, no. 3, pp. 501–510, 2000.

[52] W. Hu, Y. Qu, and G. Cheng, "Matching large ontologies: a divide-and-conquer approach," *Data & Knowledge Engineering*, vol. 67, no. 1, pp. 140–160, 2008.

[53] Y. Tian, Z. Zhang, J. Xiong, L. Chen, J. Ma, and C. Peng, "Achieving graph clustering privacy preservation based on structure entropy in social IoT," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2761–2777, 2022.

[54] B. Lima, D. Faria, F. M. Couto, I. F. Cruz, and C. Pesquita, "OAEI 2020 results for AML and AMLC," in *Proceedings of 19th International Semantic Web Conference*pp. 154–160, Cham, Switzerland, 2020.

[55] E. Jiménez-Ruiz, B. Cuenca Grau, and V. Cross, "Logmap family participation in the oaei 2017," in *Proceedings of 16th International Semantic Web Conference*, Springer, Cham, Swizerland, pp. 1–5, 2017.

[56] W. Eddine Djeddi, M. Tarek Khadir, and S. Ben Yahia, "Xmap: results for oaei 2015," in *Proceedings of 14th International Semantic Web Conference*, Springer, Cham, Swizerland, pp. 216–221, 2015.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 4126494, 11 pages
https://doi.org/10.1155/2022/4126494

Research Article

QoE-Aware Video Delivery in Multimedia IoT Network with Multiple Eavesdroppers

Dapeng Wu,[1,2,3] Ruixin Xu,[1,2,3] Hong Zhang,[1,2,3] Zhidu Li,[1,2,3] Ruyan Wang,[1,2,3] Alexander Fedotov,[4] and Vladimir Badenko[4]

[1] *School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China*
[2] *Advanced Network and Intelligent Connection Technology Key Laboratory of Chongqing Education Commission of China, Chongqing, China*
[3] *Chongqing Key Laboratory of Ubiquitous Sensing and Networking, Chongqing, China*
[4] *Peter the Great St. Petersburg Polytechnic University, Saint Petersburg, Russia*

Correspondence should be addressed to Hong Zhang; hongzhang@cqupt.edu.cn

Received 19 February 2022; Accepted 15 June 2022; Published 8 August 2022

Academic Editor: Anwar Ghani

How to deal with the increasing video traffic and diverse service demands while ensuring the security of transmission is an open issue in the multimedia Internet of Things (IoT). This paper addresses this issue and studies a secure delivery scheme under a multicast scenario in the presence of multiple eavesdroppers where small base stations (SBSs) can send videos to users cooperatively. Aiming at potential eavesdroppers, a channel model including artificial noise is introduced to reduce the harm of illegal data acquisition. A network quality of experience (QoE) optimization problem is first formulated to account for video quality and delivery delay. In order to solve the nonconvex problem, the successive convex approximation (SCA) technique is applied to optimize multicast group beamforming, reduce the possibility of multicast video eavesdropping, and select video quality where a heuristic scheme is proposed to maximize the network QoE. The effectiveness of the proposed scheme is finally validated by extensive simulations in terms of algorithm convergence performance and network QoE-enhanced performance.

1. Introduction

Recently, with the rapid development of mobile communication networks, the multimedia-oriented Internet of Things (IoT) network has shown a strong development momentum. Diversified multimedia services, such as video surveillance, make video data occupy a large proportion of multimedia IoT [1–3]. According to the existing data, the mobile network traffic data has increased by 42% from 2020 to 2021 [4]. By the end of 2022, the multimedia content data will account for 82% of the global mobile traffic [5]. The integration of various mobile terminals and multimedia IoT will promote the rapid growth of this number. In particular, the continuous upgrading of the multimedia IoT industry will have higher requirements for services, resulting in more video data, such as automatic driving and smart city. The resulting multimedia data brings more tremendous pressure to the uplink and downlink transmission of IoT [6]. In this regard, deploying MEC at the small base station (SBS) can bring services to the edge of the network and enable IoT users to obtain better experience [7, 8]. Due to the large number of users and the diversity of video requirements, it is worth studying how to efficiently deliver videos to IoT users with limited resources.

With layered technology such as H.264/moving Picture Experts Group-4 (MPEG-4) scalable video coding (SVC), a video stream can be divided into different quality levels to provide users with more personalized services [9]. Generally, more multimedia data layers can bring users a better experience, but it also needs to pay more communication costs. When faced with many requests, some additional video enhancement data may squeeze the resources of other vulnerable users and introduce additional interference, resulting in unfair resource allocation [10]. On the other hand, users with similar needs in the network can be divided into the same multicast group through multicast transmission. Multicast transmission of SVC video can make full use of its hierarchical structure, effectively reducing the system energy consumption and significantly reducing the reception delay in the face of many IoT users [11]. With a reasonable beamforming design, SBS can use limited communication resources to serve more users on the premise of reducing resource loss [9]. However, the nature of multicast transmission makes the communication over this medium vulnerable to eavesdropping [12]. In network service, transmission security and privacy issues will directly bring terrible experiences to users. A part of the literature has studied this problem from the perspective of security protocols and encryption algorithms [11, 13, 14]. In addition, there are often eavesdroppers in the process of wireless transmission. Eavesdroppers trying to access multicast services without authorization will cause economic losses to operators. Using beamforming technology and SVC technology [15–18], through more accurate beamforming design, artificial noise can be introduced to reduce the channel quality of potential eavesdroppers as much as possible, making it difficult for them to obtain complete transmitted video [19]. On the other hand, if the eavesdropper cannot obtain the basic layer data, the transmission security of the complete video can be guaranteed.

In the literature, researchers usually design optimization strategies to improve the performance in the network from two directions, that is, secure transmission and cached video delivery. On the security of transmission studies, it is often assumed that there are potential eavesdroppers in the network. The transmission signals of actual users are designed from the perspective of the physical layer to improve confidentiality and prevent data leakage [20, 21]. However, most studies regard security as the main goal rather than a prerequisite to optimizing network performance. An effective active caching strategy can filter out popular data from a large amount of data for caching to reduce the delay of users obtaining content and improve the user experience. However, this often depends on the screening of a large number of historical data, and the improvement of performance depends too much on the accuracy of cache. Using reasonable resource allocation strategy, the improvement of network performance will often have greater guarantee [22, 23]. Many efforts have been devoted to the efficient transmission of cached content at the edge of the network. However, how to jointly consider the cost of content cache location and limited network resources to deliver video while ensuring transmission security still lacks understanding.

Motivated by this, we study a video delivery scheme to maximize the weighted sum of QoE in a multimedia IoT network. The multicast transmission model with eavesdroppers, cache cost model, and QoE model are first introduced, based on which a network QoE optimization problem is formulated. By applying the SCA technique, cooperative beamforming and video quality selection are jointly optimized to guide how to deliver videos to different IoT users. Furthermore, extensive simulation experiments are carried out to verify the QoE enhancement performance of our proposed scheme.

The contributions of this paper are summarized as follows:

(i) In the multimedia IoT network scenario, the corresponding transmission model and the user QoE model are designed according to the existence of eavesdroppers. The design of the beamforming model is used to prevent insecurity in the multicast process, which is reflected in the design of the optimization problem.

(ii) An alternating iterative scheme considering system beamforming design and user acquired video quality selection strategy is designed with SCA technology. The QoE optimal solution under the condition of fixed video quality is found step by step through alternating updating.

(iii) A user selection scheme for video quality enhancement is designed by adding penalty parameters. The simulation results show that, combined with an iterative alternating algorithm, this mechanism can ensure that the system can use limited resources to obtain the best weighted QoE.

The rest of the paper is organized as follows. The related works are reviewed in Section 2. Section 3 introduces the transmission model and the optimization problem. The algorithm is designed in Section 4. Section 5 provides and discusses the simulation results. Section 6 concludes the paper.

2. Related Work

In the literature, the service experience enhancement is usually studied through proactive caching and content delivery. When seeking the best experience, the security of transmission also needs to be considered [24]. Hence, the related works are reviewed based on the lines of security of multicast transmission and cached-enabled video delivery, respectively.

On the security of transmission study, the focus is mainly on the impact of the insecure channel model and preventing content from being eavesdropped on during multicast transmission. In [20], two cognitive single-group multicast secure beamforming (SGMC-S-BF) schemes were proposed for a scenario where there exists one eavesdropper who is actually a regular user of the legitimate communication system. However, it attempts to access unauthorized multicast services. In [25], a task-oriented user selection incentive mechanism was proposed, which clusters the similarity of users by jointly considering the security and fairness of served users to realize efficient user dynamic selection. In [26], the constant modulus (CM) signaling was studied, and beamforming is designed using the semidefinite relaxation (SDR) technique and a custom-build nonconvex alternating direction method of multipliers (ADMM) algorithm, respectively. In [27], power minimization and secrecy rate maximization were considered in the secrecy network. A closed-form solution of transmit beamforming is given by exploiting the Bernstein-type inequality and the S-Procedure to convert the probabilistic secrecy rate constraint into the determined constraint. Literatures [20–27] focus on improving security performance but ignore the improvement of user performance indicators. In [21], a SDP-based secure layered video transmission scheme was proposed, but it is not suitable for the case of multiple base stations. Hence, the improvement of video delivery performance considering a nonsecure environment still needs further study in multigroup multicast scenarios.

Research on video delivery focuses on limited resource allocation and delay optimization. In [22], a dynamic interest capture model in the Industrial Internet was proposed to mine the individual user interest, based on which a group interest aggregation algorithm is then studied to determine the content caching strategies for edge nodes. In [28], an adaptive active cache scheme combined with reinforcement learning was proposed to improve user QoE of content-centered edge cache IoT and reduce cache cost. However, [22–28] mainly focus on the active cache scheme, ignoring system performance optimization by transmission strategy. In [29], the authors predicted video content requests from the perspective of the traces collected over a big city, and a joint active caching, power allocation, and user association scheme was designed to optimize the QoE of user content delivery. In [30], an active cache and user association scheme based on belief propagation was proposed to maximize the revenue of operators on the premise of ensuring the quality of service (QoS). In [31], a social-aware spectrum sharing and caching helper selection (SSC) strategy was proposed to share and cache downlink spectrum resources of multicast storage resources and unload multimedia content. In [32], the authors addressed the impact of cache on the transmission design, and a robust joint optimization strategy is designed for the case of incomplete channel information to minimize the transmission cost. In [33], a multiquality video transmission scheme based on SBS clusters division was proposed to maximize the economic efficiency (ECE) of network transmission. In summary, [29–33] optimized video delivery based on only unicast or multicast transmissions under the scenario with single base station. In addition, the above literature ignores the personalized demands of users, and multiquality video transmission is often more accurate to serve each user and save limited network resources.

Due to the increasingly personalized user demands and the complexity of the network environment, SVC technology is usually applied to meet the diverse requirements of video delivery in IoT network. In [34], the author used the stochastic geometry tool to increase the probability of successful transmission of multiquality video and proposed a two-stage transmission optimization algorithm based on the convex optimization and the packing problem. In [35], a matching sensing scheme considering relay selection was proposed to optimize the cooperative transmission performance of SVC video and improve the QoE of users. In [36], a multicast video transmission strategy based on the multicast subgrouping and SVC technology is proposed, and the transmission efficiency and fairness between users were discussed. In [37], the author addressed the application of multiquality video in multicast transmission and described the optimal beamforming as a power minimization problem and user experience maximization problem, respectively. In [23], cache-assisted data rate (CADR) was proposed as a performance index to measure the SVC video transmission performance in nonorthogonal channel D2D scene, and a two-stage joint optimization scheme is proposed. However, [34–37] mainly discussed the perspective of multiquality video transmission, but the cost of cache location and transmission interference in the case of multigroup multicast has not been considered. The D2D auxiliary transmission strategy proposed by [23] is difficult to apply directly in multicast transmission based on multi-SBS.

In summary, on the premise of the existence of eavesdroppers, there is still lack of deep understanding on how to meet the personalized user demands through jointly optimizing the multiquality video and multicast transmission beamforming and ensure transmission security, which motivates this paper.

3. System Model

In this section, we present a multimedia IoT network architecture, transmission model with eavesdroppers, cache model, and delay cost and QoE model, and the modeling of the optimization problem is introduced.

3.1. Transmission Model. As depicted in Figure 1, we focus on a cached-enabled radio access network that includes multiple densely deployed SBSs and multiple IoT users and eavesdroppers indicated by $\mathscr{K} = \{1, \ldots, k, \ldots, K\}$, $\mathscr{U} = \{1, \ldots, u, \ldots, U\}$, and $\mathscr{U}_E = \{1, \ldots, u_E, \ldots, U_E\}$, respectively, where videos can be transmitted through nonorthogonal multicast mode. In addition, each of SBS is configured with A_k antennas and can cache popular content in the equipped MEC storage unit at the network edge, so that IoT user requests can be responded to quickly. Based on the content and the quality level of a video request, the IoT users are divided into several multicast groups $\mathscr{G} = \{1, \ldots, g, \ldots, G\}$. In this regard, wireless resources can be saved by multicasting the videos to users with identical content and quality request in a group.

In this paper, SBS k can be associated with multiple groups k_g, and the user in such group is represented by g_u. The channel power gain between IoT user u, eavesdroppers j, and SBS k is denoted by $\mathbf{h}_u = [\mathbf{h}_{1,u}^H, \ldots, \mathbf{h}_{K,u}^H]^H \in \mathbb{C}^{A_n K \times 1}$ and $\mathbf{e}_u = [\mathbf{e}_{1,u_E}^H, \ldots, \mathbf{e}_{K,u_E}^H]^H \in \mathbb{C}^{A_n U_E \times 1}$, respectively. The beamforming vector of SBS k to multicast group g is denoted by $\mathscr{W}_k = \left\{\mathbf{w}_{k,g} \in \mathbb{C}^{A_k \times 1}\right\}$. On the other hand, $\mathbf{v}_k = [\mathbf{v}_1^H, \ldots, \mathbf{v}_K^H]^H \in \mathbb{C}^{A_n K \times 1}$ represents the artificial noise generated by SBS by beamforming, which is used to further reduce the channel conditions of eavesdroppers. Besides, $P_k > 0$ is used to denote the maximum transmission power of SBS k; there holds

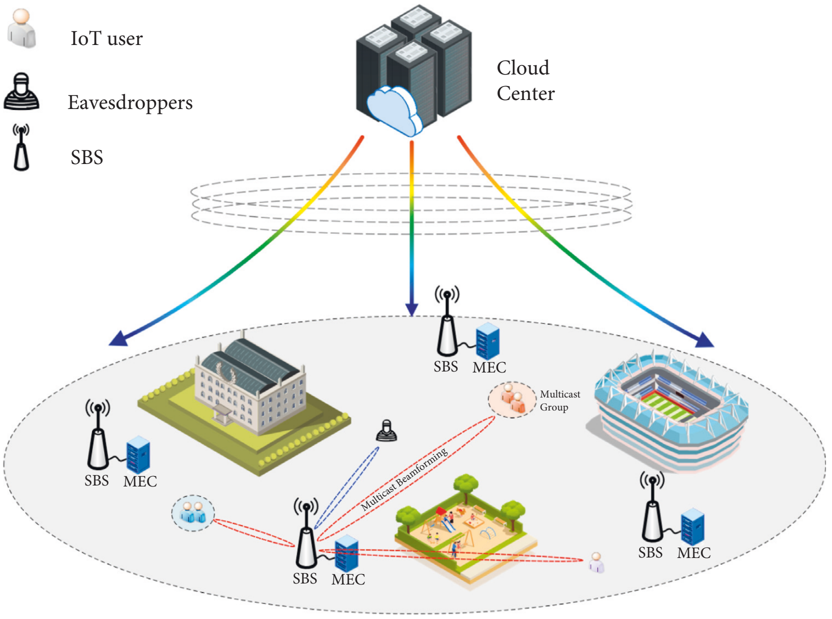


Figure 1: A video streaming multicast transmission scenario with eavesdroppers.

$$\sum_{g\in\mathcal{G}} w_{k,g2}^{2} + \mathbf{v}_{k2}^{2} \le P_k, \quad \forall k \in \mathcal{K}. \tag{1}$$

For multicast group g, we use $\mathbf{w}_g = [\mathbf{w}_{1,g}^H, \ldots, \mathbf{w}_{K,g}^H]^H$ to represent the beamforming vector group it receives. For simplicity, the SVC technology encodes a video into two layers, and SBSs will store different quality files of a video in the cache space. To deal with the quality level of each video, the base layer (BL) and the enhancement layer (EL) are used to characterize the basic video quality and the enhancement video quality, respectively. The multimedia contents quality requested by IoT user u is represented by $l_u = \{1, 2\}$, where $l_u = 1$ represents that IoT user requests low-quality multimedia contents and $l_u = 2$ represents that IoT user requests high-quality multimedia contents. Besides, the requested video content and the video quality of the multicast group g are represented as g_f and g_l, respectively. Note that high-quality multimedia contents include both BL data and EL data at the same time. A video content cannot be decoded with only EL data. Let l_u^{req} denote the quality level requested by user u; there holds

$$l_u \ge l_u^{\text{req}}, \quad \forall u \in \mathcal{U}. \tag{2}$$

In order to improve the multimedio content delivery efficiency, a user can be assigned to at most two groups where one group only receives BL data and the other receives EL data, as depicted in Figure 2. The network only needs to allocate a multicast beamforming vector to provide BL data to users requesting low-quality video and users requesting high-quality video simultaneously. On the other hand, when SBS transmits data to multicast group users, eavesdroppers can receive broadcast information and obtain unauthorized video content. By reducing its signal-to-interference-to-noise ratio (SINR), it cannot restore the complete video, so as to prevent the content from being intercepted. In this sense, the network only needs to assign only one multicast beamforming vector to provide BL version video to the users requesting low-quality videos and those requesting high-quality videos at the same time. The received signals of IoT user u and eavesdropper j are given by, respectively,

$$\begin{aligned} y_u &= \sum_{g\in\mathcal{G}_u} \mathbf{h}_u^H \mathbf{w}_g s_g + \sum_{i\in G\setminus\mathcal{G}_u} \mathbf{h}_u^H \mathbf{w}_i s_i + n_u, \quad \forall u \in \mathcal{U}, \\ y_{u_E} &= \sum_{g\in\mathcal{G}} \mathbf{e}_{u_E}^H \mathbf{w}_g s_g + \sum_{k\in\mathcal{K}} \mathbf{e}_{u_E}^H \mathbf{v}_k s_g + n_j, \quad \forall u_E \in \mathcal{U}_E, \end{aligned} \tag{3}$$

where $\mathcal{G}_u$ represents the multicast group set assigned by the user u. s_g with $\mathbb{E}|s_g|^2 = 1$ represents the precoding binary symbols of the video requested by group g, and $n_u \sim CN(0, \sigma_u^2)$ denotes the additive white Gaussian noise (AWGN). Applying Shannon's theory, the transmission rate of u when requesting a video f with quality level l holds as

$$R_{u,f_l} = B\log_2\left(1 + \Gamma_{u,f_l}\right). \tag{4}$$

The SINR is denoted by

$$\Gamma_{u,f_l} = \frac{\left|\mathbf{h}_u^H \mathbf{w}_{g,f_l}\right|^2}{I_{u,f_l}}, \quad \forall u \in g,\ g \in \mathcal{G}, \tag{5}$$

where

$$\begin{aligned} I_{u,f_l} = &\sum_{l>l_{\text{req}}} \left|\mathbf{h}_u^H \mathbf{w}_{g,f_l}\right|^2 + \sum_{i\in\setminus\{g\}} \left|\mathbf{h}_u^H \mathbf{w}_i\right|^2 + \sigma^2, \\ &\forall u \in g,\ g \in \mathcal{G}. \end{aligned} \tag{6}$$

In (5), $\mathbf{w}_{g,f_l}$ denotes the multicast group beamforming vector which provides content f with quality level l. In (6), the first term denotes higher-quality version signal

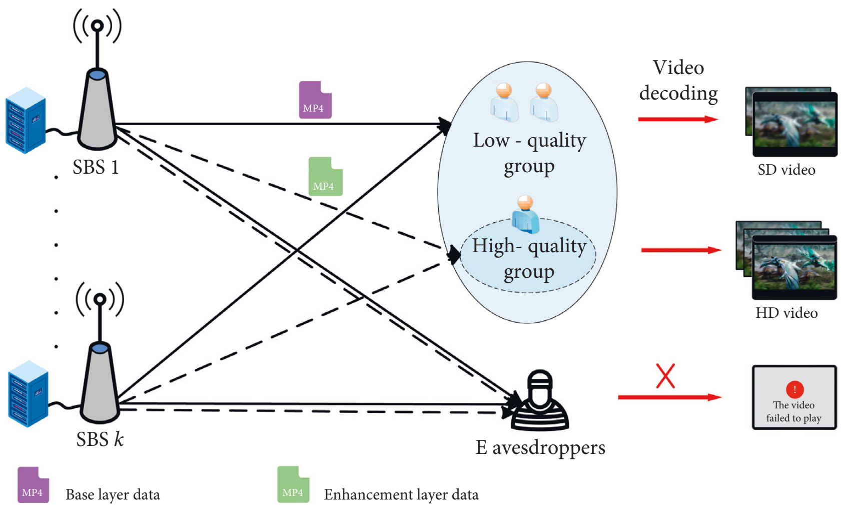


Figure 2: Cooperative video multicast transmission model.

interference to the same video content, and the second term denotes the other group signal interference. Since SBS actively generates artificial noise, the coded information can be transmitted to the user in advance, so it will not be regarded as noise. In addition, to ensure the user fairness in group g, the transmission rate should depend on the user with the worst channel condition in the group, and the group transmission rate thus holds

$$R_{g,f_l} = \min_{u\in g} R_{u,f_l}. \tag{7}$$

On the other hand, the transmission rate of eavesdropper j can be obtained as

$$\begin{gathered} R_{u_E,f_l} = B\log_2\left(1 + \Gamma_{u_E,f}\right), \\ \Gamma_{u_E,f} = \frac{\left|\mathbf{e}_{u_E}^H \mathbf{w}_{g,f_l}\right|^2}{I_{u_E}}, \forall u_E \in \mathcal{U}_E, \end{gathered} \tag{8}$$

where

$$I_{u_E} = \sum_{i>1} \left|\mathbf{e}_u^H \mathbf{w}_i\right|^2 + 4\sum_{k\in\mathcal{K}} \left|\mathbf{e}_u^H \mathbf{v}_k\right|^2 + \sigma^2, \forall u_E \in \mathcal{U}_E. \tag{9}$$

In (9), the noise can inform the authorized user in advance, and it will not cause additional interference to the target user. Due to the characteristics of SVC, we can ensure the security of multicast video transmission only by ensuring that the video of BL layer is not completely eavesdropped. Therefore, we give a maximum tolerance $\Gamma_{u_E,f} < \Gamma_{\text{toler}}$ of the eavesdropper, so that it cannot obtain the complete video.

3.2. Cache Model. The limited video content library is denoted by $\mathcal{F} = \{1, \ldots, f, \ldots, F\}$, where F is the library size. The size of a unit video block with quality l of different content f is denoted by $S_{f,l}$. In order to facilitate the experiment, different video blocks of the same quality are assumed to have the same size S_l, which can be realized by adaptive slicing of video. The popularity of all contents is assumed to follow the Zipf distribution [38], that is, the probability that content $f - th$ being requested is given by

$$\begin{gathered} P(f) = \frac{f^{-\alpha}}{\sum_{i=1}^{F} i^{-\alpha}}, \\ f = 1, 2, \ldots, F, \end{gathered} \tag{10}$$

where α denotes the Zipf parameter. The larger α, the more significant the popularity skewness between different contents. Suppose that each SBS will cache contents according to the popularity ranking until the cache space is full, and the cache state of content f at SBS k can be obtained as

$$c_{k,f} = \begin{cases} 1, & \text{if } f \text{ is cached by SBS } k, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

3.3. Delay Cost and QoE Model. As the cache capacity of each MEC is finite, only some popular videos can be precached. If a video is not cached in the MEC of the local SBS, the content must be obtained from the neighbor SBS or cloud center, which will introduce the extra delay in the video transmission process. Since multiple SBSs cooperate to transmit multimedia content to IoT users simultaneously, let D_{f_u} denote the extra delay; there holds

$$D_{f_u} = \begin{cases} 0, & \prod_{k\in\mathcal{K}} c_{k,f} = 1, \\ d_f, & \prod_{k\in\mathcal{K}} c_{k,f} = 0,\ \exists c_{k,f} = 1,\ \forall k \in K, \\ d_0, & \sum_{k\in\mathcal{K}} c_{k,f} = 0. \end{cases} \tag{12}$$

Equation (12) indicates that there is no extra delay if the video is stored in all MEC cache equipped with SBS. If the video is only cached in the MEC of a part of SBSs, the extra delay equals a fixed value d_f. It is caused by the video content sharing between SBSs using Xn interface in the 5G networks [39]. A fixed extra delay $d_0\,(d_0 \gg d_f)$ is needed if the video can only be obtained in the cloud center [40]. Furthermore, taking the video transmission delay from the local SBS to the user into account, the total video delivery delay is given by

$$D_u = \frac{S_l}{R_{g,f_l}} + D_{f_u}, \quad \forall u \in g, \forall g \in \mathscr{G}. \tag{13}$$

Compared with other factors, the startup delay of the video often directly affects the willingness of the user to play the video. In addition, from the practical experience, the benefits of reducing the startup delay meet the feature of diminishing margins rather than a fixed value. The lower the delay is, the lower the QoE improvement that can be achieved. Hence, we apply a logarithm to characterize the impact of startup delay of the video on the QoE in this paper [41]. On the other hand, a higher-quality level of video is able to improve the QoE compared to the one with low-quality level. Hence, the QoE of user u can be modelled by

$$QoE_u = (1-\eta)\frac{l_u}{l_{\mathrm{EL}}} + \eta\left(\log_2\left(1 + \frac{\beta}{\max\limits_{l \le l_u}\{D_u\}}\right)\right), \tag{14}$$

where $\eta \in [0,1]$ is a weight factor that characterizes the importance of the video quality and that of the delay on the QoE. Besides, β is a positive parameter used to control the marginal benefit of the startup delay. The first term indicates the impact of the definition of the user u-requested video on the overall QoE. The second term indicates the impact of the transmission delay on the overall QoE, where $\max_{l \le l_u}\{D_u\}$ represents the maximum transmission delay of all the video data received by user u.

3.4. Problem Formulation. In this paper, the aim is to maximize the sum of the QoE of each user that is called network QoE through jointly optimizing the video version selection and multicast group beamforming; there holds

$$Q = \sum_{u \in \mathscr{U}} QoE_u. \tag{15}$$

In order to improve the QoE of the IoT user, an efficient beamforming strategy can effectively reduce the transmission delay in the wireless transmission process. At the same time, the eavesdropper should provide the poor channel conditions designed to hinder its action. On the premise of ensuring the basic demands of users, providing users with video quality enhancement services at a low enhancement cost also helps to improve the QoE of the network. However, the improvement of video quality will bring additional interference to the same frequency transmission. Therefore, video quality selection and multicast group beamforming are jointly optimized. The optimization problem is formulated as follows:

$$\mathscr{P}_1 \max_{\mathbf{w},\mathbf{l}} Q, \tag{16a}$$

$$s.t.\; R_{g,f_l} \le R_{u,f_l}, \forall u \in g, \forall g \in \mathscr{G}, \tag{16b}$$

$$R_l^{\mathrm{req}} \le R_{g,f_l}, \quad \forall g \in \mathscr{G}, \tag{16c}$$

$$\sum_{g \in \mathscr{G}} \left\|\mathbf{w}_{k,g}\right\|_2^2 + \mathbf{v}_{k2}^2 \le P_k, \quad \forall k \in \mathscr{K}, \tag{16d}$$

$$\Gamma_{u_E,f} < \Gamma_{\mathrm{toler}}, \forall u_E \in \mathscr{U}_E, \tag{16e}$$

$$l_u \ge l_u^{\mathrm{req}}, \quad \forall u \in \mathscr{U}. \tag{16f}$$

In problem$\mathscr{P}_1$, (16b) is the fairness constraint within the multicast group, (16c) represents the video bit rate constraint to ensure the QoS requirements of each, where $R_l^{\mathrm{req}}$is the bit rate threshold corresponding to the video quality, (16d) is to the power constraint of each SBS, (16e) is SINR constraint of eavesdropper, (16f) means that the quality level of the video obtained by each user should be not lower than the corresponding requested one.

4. Algorithm Design

Since $\mathscr{P}_1$ is an MINLP problem, it is difficult to solve by convex technique directly. Therefore, it is promising to decouple the process of video quality selection and that of multicast group beamforming. First, the video quality selection parameter is fixed as **l**. The optimization problem $\mathscr{P}_1$ can then be transformed as

$$\mathscr{P}_{1,1} \max_{\mathbf{w},\mathbf{v}} \sum_{u \in \mathscr{U}} \left(\log_2\left(\left(\frac{1+\beta}{\max\limits_{l \le l_u}}\right)\{D_u\}\right)\right), \tag{17a}$$

$$\text{s.t. } (16b), (16c), (16d), (16e). \tag{17b}$$

However, the quadratic term of $\mathbf{w}$ in constraint (16b) appears in both the numerator and the denominator, resulting in its nonconvex nature. On the other hand, since the right side of (16e) is a constant term, it is still a convex term, and the artificial noise part $\mathbf{v}$ can be solved directly. To extract the fractional part, we introduce auxiliary variables in the SINR part of (16b) and modify the original constraint into

$$\Gamma_u \le \frac{\left|\mathbf{h}_u^H \mathbf{w}_{g_u}\right|^2}{I_u}, \quad \forall u \in g, \forall g \in \mathscr{G}, \tag{18}$$

$$\sum_{l > l_{req}} \left|\mathbf{h}_u^H \mathbf{w}_{g,f_l}\right|^2 + \sum_{i \in G \setminus \{g\}} \left|\mathbf{h}_u^H \mathbf{w}_i\right|^2 + \sigma^2 \le I_u, \quad \forall u \in g, \forall g \in \mathscr{G}. \tag{19}$$

Consequently, the nonconvex constraint (16b) of the original problem is replaced by (18) and (21). Nevertheless, constraint (18) is still a nonconvex constraint. In this regard, we design an approximate convex lower bound to relax constraint (18). Let $f(\mathbf{w}_{g_u}, I_u) = \|\mathbf{h}_u^H \mathbf{w}_{g_u}\|_2^2 / I_u$ and perform

Taylor's first-order expansion of $f(\mathbf{w}_{g_u},\gamma_u)$ at feasible points $\mathbf{w}_{g_u}^{(t)}$ and $\gamma_u^{(t)}$; constraint (18) can then be replaced by an auxiliary function as

$$\psi^{(t)}\left(\mathbf{w}_{g_u},\gamma_u\right) \triangleq \frac{2\mathscr{Re}\left(\left(\mathbf{w}_{g_u}^{(t)}\right)^H \mathbf{h}_k\mathbf{h}_k^H\mathbf{w}_{g_u}\right)}{I_u^{(t)}} - \left(\frac{\left|\mathbf{h}_u^H\mathbf{w}_{g_u}^{(t)}\right|}{I_u^{(t)}}\right)^2\gamma_u, \tag{20}$$

where $\mathscr{Re}(\cdot)$ represents the real part and t represents the number of iterations. For any $(\mathbf{w}_{g_u}^{(t)}, I_u^{(t)})$ that satisfies the constraint, there are $f(\mathbf{w}_{g_u}^{(t)}, I_u^{(t)}) = \psi(\mathbf{w}_{g_u}^{(t)}, I_u^{(t)})$ and $\nabla f(\mathbf{w}_{g_u}, I_u) = \nabla\psi(\mathbf{w}_{g_u}, I_u)$. Therefore, constraint (18) is transformed into a second-order cone (SOC) constraint as

$$\Gamma_u \le \psi^{(t)}\left(\mathbf{w}_{g_u}, I_u\right). \tag{21}$$

Since the second derivative of (21) is more than zero, the transformed constraint has the convex property. Through iteration, the problem can be solved using the CVX tool and MATLAB. To substitute $\mathbf{w}$ with $\mathbf{w}^{(t)}$ in problem $\mathscr{P}_{1,1}$, parameters $I^{(t)}$, $\Gamma^{(t)}$, and $\psi^{(t)}$ can be obtained. In this way, all constraints are still satisfied since $\mathbf{w}^{(t)}$ is a feasible point. Further applying $I^{(t)}$, $\Gamma^{(t)}$, and $\psi^{(t)}$ to solve problem $\mathscr{P}_{1,1}$, the solution $\mathbf{w}^{(t+1)}$ can be further regarded as the input of the $(t+1)$-th iteration. In the iterative process, the construction of $\psi^{(t)}$ requires the same gradient value as the original function, the objective function value of the $t-th$ iteration must not be greater than that of the $(t+1)-th$ iteration, there holds $\sum_{u\in\mathscr{U}} Q_u^{(t+1)} \ge \sum_{u\in\mathscr{U}} Q_u^{(t)}$, and the iterative process is monotonically decreasing. In addition, considering that the system power is limited, the convergence of the iterative solution can be guaranteed according to the monotone boundedness theorem. When $\mathbf{w}^{(t)} = \mathbf{w}^{(t+1)}$ holds, the iteration converges and the optimal multicast group beamforming can be achieved. Generally, the convergence speed of the optimization problem is strongly related to the initial beamforming value in the first iteration, that is, $\mathbf{w}^{(0)}$. In what follows, an auxiliary problem is further formulated to guide how to determine $\mathbf{w}^{(0)}$:

$$\mathscr{P}_{1,2} \max_{\mathbf{w},\mathbf{v}} \delta, \tag{22a}$$

$$\text{s.t. } R_l^{\text{req}}\delta \le B\log_2\left(1+\varphi_u\right), \quad \forall u \in g, \forall g \in \mathscr{G}, \tag{22b}$$

$$(16d), (16e), (19), (21), \tag{22c}$$

where $\delta = \min_{u\in\mathscr{U}}\left\{R_{u,l}/R_l^{\text{req}}\right\}$, which is called the rate satisfaction. $\delta \ge 1$ indicates that the corresponding solution $\mathbf{w}^*$ satisfies (16c) and (16d). The solution approach of problem $\mathscr{P}_{1,2}$ is summarized in Algorithm 1. By setting $\mathbf{w}^0 = \mathbf{w}^*$, problem $\mathscr{P}_{1,1}$ can be solved iteratively by convex tools.

In step 3, Algorithm 1 solves a second-order cone programming problem, calculated by the interior point method, and its computational complexity can be expressed as $\mathscr{O}(((G+3)(A_kK+4))^{3.5})$. In addition, the computation cost in each iteration is bounded by $\mathscr{O}(T_1((G+3)(A_kK+4))^{3.5})$, where T_1 is the number of iterations in Algorithm 1.

In the video quality selection process, in order to degrade the interference from the signal of EL video to other groups, a heuristic algorithm is designed as follows. For a given user u, an auxiliary variable z_u is introduced to measure the cost of quality enhancement, which can further help to select the video quality level. The QoS constraint (16c) is then transformed into $(R_{g,l} - R_{u,l}) \le z_u$, where larger z_u results in more costs to enhance the video quality for user u, and vice versa. Therefore, we further formulate the following optimization problem to minimize the sum of z_u of each user:

$$\mathscr{P}_{1,3} \min_{\mathbf{w},\mathbf{v}} \sum_{u\in\mathscr{U}_{\text{BL}}} z_u, \tag{23a}$$

$$\text{s.t. } R_{g,f_l} \le B\log_2\frac{\left(1+\varphi_u\right)}{R_l^{\text{req}}} + z_u, \quad \forall u \in g, \forall g \in \mathscr{G}, \tag{23b}$$

$$0 \le z_u, \quad \forall u \in \mathscr{U}, \tag{23c}$$

$$(16b), (16c), (16e), (19), (21), \tag{23d}$$

where $\mathscr{U}_{\text{BL}}$ represents the user who initially requested the low-quality multimedia content. In (23c), the corresponding enhancement cost of IoT users requesting high-quality multimedia content are all set to 0. By solving problem $\mathscr{P}_{1,3}$, the video quality enhancement cost set $\mathbf{z}$ can be obtained, which is further used to select the user in $\mathscr{U}_{BL}$ with the smallest $\mathbf{z}$ to provide EL videos to improve the QoE performance.

Based on Algorithm 1, the complexity of Algorithm 2 is mainly determined by the user selection process in step 5 and the maximization process of QoE in step 10. T_2 and T_3 represent the iteration times of the above two steps, respectively, and the computational complexity of Algorithm 2 can be expressed as $\mathscr{O}((T_1 + U_{BL}(T_2+T_3))((G+3)(A_kK+4))^{3.5})$, where U_{BL} is the number of IoT users requesting low-quality video.

5. Simulation Results

In this section, simulation results are presented and discussed. Without other highlights, the simulation parameters are set as follows. The simulation scenario includes 7 SBSs with $A_k = 2$ that are distributed on the vertex and center of a regular hexagon with a side length of 100 m, and 40 IoT users and 3 eavesdroppers that are randomly distributed on a circle with a radius of 200 m. The channel gain from SBS k to the user u is defined as $\mathbf{h}_{k,u} = \sqrt{1/(1+d_{k,u}/d_0)^{\rho}}\tilde{h}_{k,u}$, where the path loss factor is set as $\rho = 3$, the standard distance is set as $d_0 = 50m$, and the noise power is set as 1. According to the transmission unit size requirements in IEEE 802.11, we set the EL data to 2 Mbit, corresponding to 1080 p definition video, and set the BL data to 1 Mbit, corresponding to 720 p video. The remaining simulation parameters are shown in Table 1. The network performance is evaluated in terms of average network rate, multicast group rate, and network QoE.

Figure 3 shows the convergence performance of Algorithm 1 under four different random channel realizations

```
Input: Channel condition and QoS threshold of multi-quality video.
Output: Optimal beamforming vector and rate satisfaction.
Step:
(1) Set the $t=0$, beamforming vector $\mathbf{w}^{(0)}, \mathbf{v}^{(0)}$ satisfies constraint (16d);
(2) Calculate $I_u^{(t)}$ of each user as follows:
        $I_u^{(t)} = \sum_{l>l_{\text{req}}} |\mathbf{h}_u^H \mathbf{w}_{\mathbf{g},f_l}^{(\mathbf{t})}|^2 + \sum_{i\in G\setminus\{g\}} |\mathbf{h}_u^H \mathbf{w}_i^{(\mathbf{t})}|^2 + \sigma^2, \forall u \in g, \forall g \in \mathcal{G}$
(3) Solve $\mathcal{P}_{1,2}$;
(4) Update $\mathbf{w}^{(\mathbf{t+1})}, \mathbf{v}^{(\mathbf{t+1})}, \mathbf{r}^{(\mathbf{t+1})}, \Gamma^{(\mathbf{t+1})}$;
(5) $\delta^* \leftarrow \delta^{(t)}$;
(6) $\leftarrow \mathbf{v}^{(\mathbf{t})}$;
(7) $\mathbf{w}^* \leftarrow \mathbf{w}^{(\mathbf{t})}$;
(8) If $\delta^{(\mathbf{t+1})} - \delta^{(\mathbf{t})} \leq \epsilon$ iteration stop. Otherwise, set $t \leftarrow t+1$ and go to Step 2.
(9) Output $\mathbf{w}^*$, $\mathbf{v}^*$ and $\delta^*$.
```

Algorithm 1: Maximize rate satisfaction δ algorithm.

```
Input: Channel condition, cache transmission delay, delay satisfaction factor and QoS threshold of multi-quality video.
Output: Optimal beamforming vector, optimal version selection set and maximum network QoE.
Step:
(1) Set the $t=0$, feasible beamforming vector $\mathbf{w}^{(0)}, \mathbf{v}^{(0)}$, only require low-quality video user set $\mathcal{U}_{BL}$;
(2) Calculate the optimal user rate satisfaction $\delta^*$ and optimal beamforming vector $\mathbf{w}^*, \mathbf{v}^*$ by solving $\mathcal{P}_{1,2}$.
(3)If $\delta^* > 1$
(4)     Calculate current optimal QoE $Q^*$ by solve $\mathcal{P}_{1,1}$;
(5)     Calculate the enhancement cost set $\mathbf{z}$ by solve $\mathcal{P}_{1,3}$;
(6)     Obtain the user index $\mathbf{u}^*$ corresponding to the minimum value in $\mathbf{z}$;
(7)     $\mathcal{U}_{BL} \leftarrow \mathcal{U}_{BL} \setminus u^*$;
(8)     $l_{u^*} \leftarrow 2$;
(9)     Calculate network QoE $Q\prime$ and beamforming vector $\mathbf{w}\prime$ by solve $\mathcal{P}_{1,1}$;
(10)    If the problem is unsolved or the network QoE drops $(Q' \leq Q^*)$
(11)        Break.
(12)    Else
(13)        Update $Q^* \leftarrow Q'$, $\mathbf{w}^* \leftarrow \mathbf{w}\prime$, $\mathbf{v}^* \leftarrow \mathbf{v}\prime$, $\mathbf{l}^* \leftarrow \mathbf{l}'$;
(14)    End if
(15)    If $\mathcal{U}_{BL} = \varnothing$ iteration stop. Otherwise, set $t \leftarrow t+1$ and go to Step 5.
(16) Else
(17)    The solution is not feasible, the current resources cannot satisfy all users.
(18) End if
(19) Output $\mathbf{w}^*$, $\mathbf{v}^*$, $\mathbf{l}^*$ and $Q^*$.
```

Algorithm 2: Maximize QoE algorithm.

Table 1: Simulation parameters.

Symbol	Value
Transmission power of SBS	40 dBm
Bandwidth	10 MHz
Number of video contents	100
Cache ratio in each MEC	0.6
Video block size	1 Mbit (BL), 2 Mbit (EL)
Maximum tolerance SINR of eavesdropper	−10 dB
Delay satisfaction factor	0.2
Marginal effect factor	5
Transmission delay	0.2 (core network)/0.05 (MEC)
Zipf coefficient	1

(RCR) from the average transmission rate. All the simulation results are under the condition of secure communication. Under the interference of artificial noise, the eavesdropper cannot obtain the content at the lowest decoding rate. The beamforming of the multicast group is initialized to $v_{k,a}^H, w_{k,g,a}^H = \text{rand} \times \sqrt{P_k/((G+1) \times A_k)}$, where $ran\ d$ is a random factor within [0, 1]. It is observed that, after the first 5 iterations, the average rate in four different RCR can reach the 96.72% level obtained in the 20th iteration, which verifies the convergence performance of Algorithm 1. Moreover, the convergence performance of Algorithm 1 proposed is insensitive to the channel conditions, which verifies the robustness of the proposed scheme.

In Figure 4, four multicast groups are randomly selected, and the rate performance of multicast groups with different delay costs and intra group video request quality is compared.

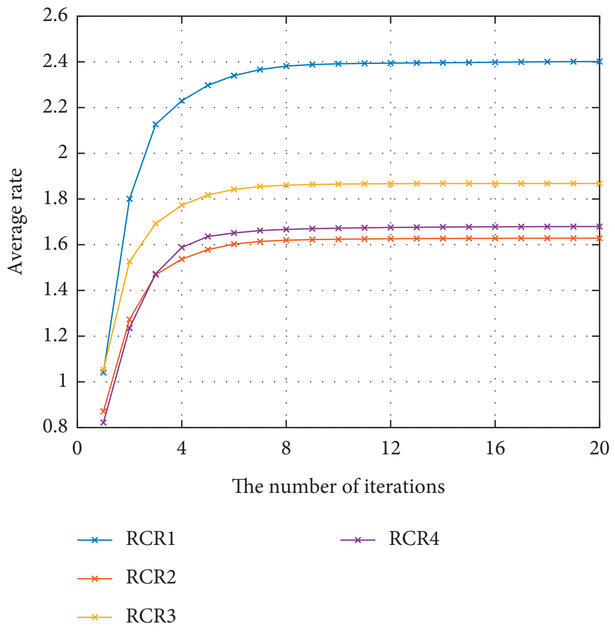


Figure 3: Average rate under different RCR.

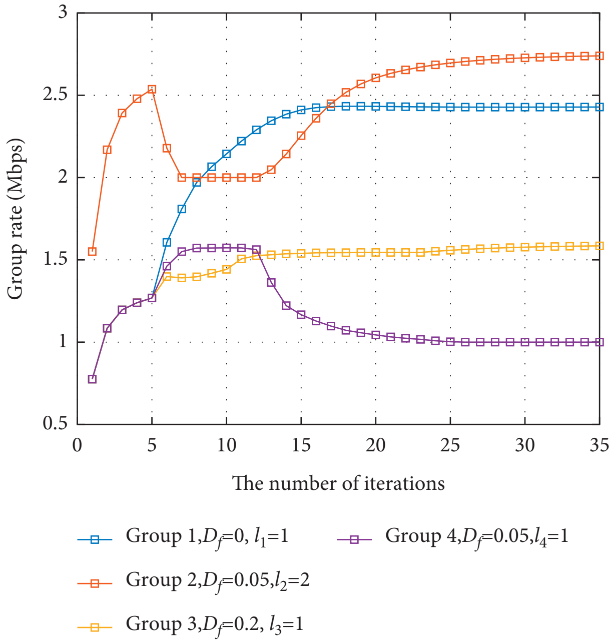


Figure 4: Multicast group rate performance in different groups.

In the first five iterations, the proposed scheme tries to find out the appropriate initial parameters for subsequent optimization. After that, the QoE is maximized iteratively. First, group 2 and group 4 changed significantly near the fifth and 12th iterations. This is because they have high video quality requirements. Allocating additional resources helps improve the previous item of QoE, which is more advantageous than the optimization of delay. For group 4, because there are few users in the group and the channel conditions are poor, it is only necessary to maintain a minimum QoS requirement at the end to save limited resources. Group 1 selects the multicast group with the most members in the group and allocates resources to it, helping to improve the efficiency of beamforming and optimize network QoE. For group 3, although the delay cost $D_f = 0.2$, this makes the QoE delay part have a large room for improvement, so it is still in a slow rising state. The above shows that, with the optimization of network QoE, the resource scheduling of different groups will still be adjusted adaptively according to iteration.

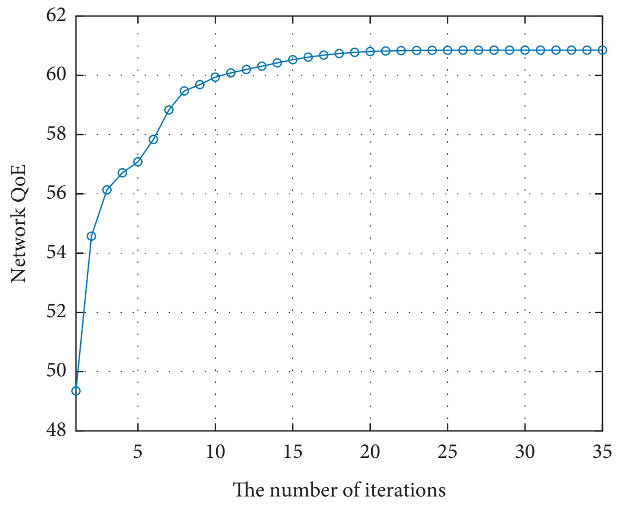


Figure 5: Trajectory of network QoE weighted sum.

Figure 5 shows the trend of network QoE weighted sum. Although the allocation of resources between different multicast groups fluctuates in Figure 4, the QoE of the whole network increases monotonically. The allocation of resources will only change the growth rate of network QoE, which shows the effectiveness of the proposed algorithm for network QoE optimization.

In Figure 6, the network QoE performance was depicted under different cache capabilities of SBSs. In order to validate the effectiveness of the proposed scheme, three baseline schemes, namely, QoE-Max, QoS-Only, and Unicast, are introduced. The QoE-Max scheme maximizes the network QoE based on multicast group beamforming optimization only, without considering the enhancement of video quality. The QoS-Only scheme only guarantees the basic QoS requirement of video quality of each user. The Unicast-QoE algorithm will transmit multimedia content to users through unicast cooperative transmission, producing additional interference. The cache ratio in Figure 5 indicates the proportion of the content library that an MEC can cache. The network QoE of the four schemes shows an increasing trend, but there are still some fluctuations due to the uncertainty of users' requests for popular content. In addition, as the MEC cache ratio increases, more space is used to store infrequently used content. Furthermore, compared with the Unicast-QoE algorithm with the worst performance, the proposed algorithm improves the network QoE by 23.59% on average. And compared with the QoE-Max algorithm, which ignores video quality enhancement, the proposed algorithm improves by 6.68% because the additional video quality brings users a better experience. During the experiment, the Unicast-QoE algorithm needs to design

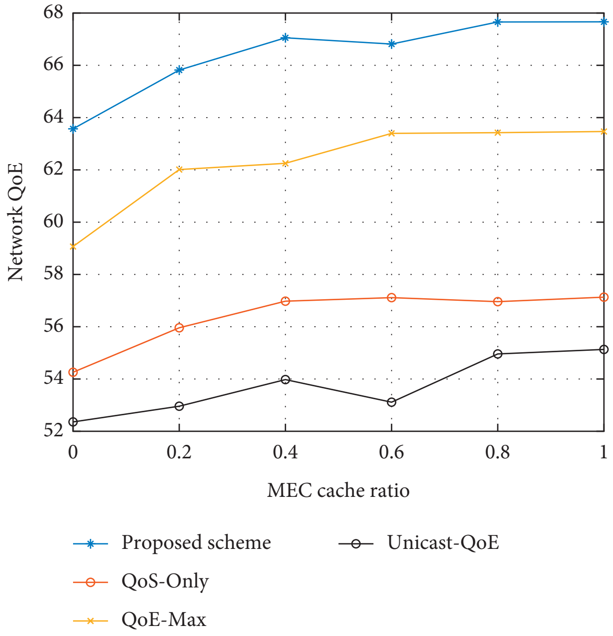


Figure 6: Network QoE performance under different cache capabilities of SBSs.

beamforming separately for all users, which will produce additional resource consumption and interference. In addition, among the four algorithms, the proposed algorithm also has the best performance because the proposed algorithm is optimized for video quality and beamforming simultaneously.

6. Conclusions

In this paper, a QoE-aware video delivery scheme was studied in multimedia IoT network with potential eavesdroppers. A joint video quality selection and multicast group beamforming scheme were proposed to maximize the network QoE while preventing data eavesdropping as much as possible. Extensive simulation results validated the effectiveness of the proposed scheme in terms of user satisfaction, multicast group rate, and network QoE [42].

Data Availability

The relevant data used to support the results 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 in part by the National Natural Science Foundation of China (61871062, 61771082, 61901071, and U20A20157), the Science and Technology Research Program of Chongqing Municipal Education Commission (KJQN202000626), Natural Science Foundation of Chongqing, China (cstc2020jcyj-zdxmX0024), and University Innovation Research Group of Chongqing (CXQT20017).

References

[1] B. Jedari, G. Premsankar, G. Illahi, M. D. Francesco, A. Mehrabi, and A. Ylä-Jääski, “Video caching, analytics, and delivery at the wireless edge: a survey and future directions,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 431–471, 2021.

[2] C. Long, Y. Cao, T. Jiang, and Q. Zhang, “Edge computing framework for cooperative video processing in multimedia IoT systems,” *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1126–1139, 2018.

[3] A. Nauman, Y. A. Qadri, M. Amjad, Y. B. Zikria, M. K. Afzal, and S. W. Kim, “Multimedia Internet of things: a comprehensive survey,” *IEEE Access*, vol. 8, pp. 8202–8250, 2020.

[4] Ericsson, “Ericsson mobility report,” https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/november-2021.

[5] V. Cisco, “networking index: Global mobile Data Traffic Forecast Update,” https://www.cisco.com.

[6] S. He and W. Wang, “Multimedia upstreaming cournot game in non-orthogonal multiple access Internet of things,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 398–408, 2020.

[7] 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.

[8] Z. Li, Y. Zhou, D. Wu, T. Tang, and R. Wang, “Fairness-aware federated learning with unreliable links in resource-constrained Internet of things,” *IEEE Internet of Things Journal*, p. 1, 2022.

[9] Y. Fallah, H. Mansour, S. Khan, P. Nasiopoulos, and H. Alnuweiri, “A link adaptation scheme for efficient transmission of H.264 scalable video over multirate WLANs,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 7, pp. 875–887, 2008.

[10] H. Zhu, Y. Cao, T. Jiang, and Q. Zhang, “Scalable NOMA multicast for SVC streams in cellular networks,” *IEEE Transactions on Communications*, vol. 66, no. 12, pp. 6339–6352, 2018.

[11] S. Pizzi, C. Suraci, A. Iera, A. Molinaro, and G. Araniti, “A sidelink-aided approach for secure multicast service delivery: from human-oriented multimedia traffic to machine type communications,” *IEEE Transactions on Broadcasting*, vol. 67, no. 1, pp. 313–323, 2021.

[12] S. Goel and R. Negi, “Guaranteeing secrecy using artificial noise,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2180–2189, 2008.

[13] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, “Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles,” *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.

[14] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, “A blockchain-based machine learning framework for edge services in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, 2022.

[15] D. Hwang, J. Yang, K. Kwon, J. Joung, and H.-K. Song, “Equalization-based beamforming for secure multicasting in multicast wiretap channels,” *IEEE Access*, vol. 9, pp. 33826–33835, 2021.

[16] H. Luo, Q. Li, L. Yang, and J. Qin, "A fast algorithm for fractional QCQP and applications to secure beamforming in cognitive nonorthogonal multiple access networks," *IEEE Transactions on Signal Processing*, vol. 69, pp. 6237–6250, 2021.

[17] P. Huang, Y. Hao, T. Lv, J. Xing, J. Yang, and P. T. Mathiopoulos, "Secure beamforming design in relay-assisted Internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6453–6464, 2019.

[18] P. V. Tuan, T. Trung Duy, and I. Koo, *Multiuser MISO Beamforming Design for Balancing the Received Powers in Secure Cognitive Radio Networks*, in *Proceesings of the in 2018 IEEE Seventh International Conference on Communications and Electronics (ICCE)*, pp. 39–43, Hue, Vietnam, July2018.

[19] M. R. A. Khandaker and K. K. Wong, "Masked beamforming in the presence of energy-harvesting eavesdroppers," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 40–54, 2015.

[20] S. Fan and J. Xu, *Single-Group Multicast Secure Beamforming via Learning the Eavesdropper's Channel Correlation*, in *Proceedings of the in 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, Dublin, Ireland, July2020.

[21] D. W. K. Ng, R. Schober, and H. Alnuweiri, "Power efficient MISO beamforming for secure layered transmission," in *Proceedings of the in 2014 IEEE Wireless Communications and Networking Conference*, pp. 422–427, Istanbul, Turkey, April2014.

[22] Z. Li, X. Gao, Q. Li, J. Guo, and B. Yang, "Edge caching enhancement for industrial Internet: a recommendation-aided approach," *IEEE Internet of Things Journal*, p. 1, 2022.

[23] J. Ma, L. Liu, H. Song, R. Shafin, B. Shang, and P. Fan, "Scalable video transmission in cache-aided device-to-device networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4247–4261, 2020.

[24] Y. Hong, X. Jing, and H. Gao, "Programmable weight phased-array transmission for secure millimeter-wave wireless communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 2, pp. 399–413, 2018.

[25] J. Xiong, X. Chen, Q. Yang, L. Chen, and Z. Yao, "A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2347–2360, 2020.

[26] Q. Li, C. Li, and J. Lin, "Constant modulus secure beamforming for multicast massive MIMO wiretap channels," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 264–275, 2020.

[27] Z. Chu, H. Xing, M. Johnston, and S. Le Goff, "Secrecy rate optimizations for a MISO secrecy channel with multiple multiantenna eavesdroppers," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 283–297, 2016.

[28] D. Huang, X. Tao, C. Jiang, S. Cui, and J. Lu, "Trace-driven QoE-aware proactive caching for mobile video streaming in metropolis," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 62–76, 2020.

[29] X. He, K. Wang, and W. Xu, "QoE-driven content-centric caching with deep reinforcement learning in edge-enabled IoT," *IEEE Computational Intelligence Magazine*, vol. 14, no. 4, pp. 12–20, 2019.

[30] L. Liu, Y. Zhou, J. Yuan, W. Zhuang, and Y. Wang, "Economically optimal MS association for multimedia content delivery in cache-enabled heterogeneous cloud radio access networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 7, pp. 1584–1593, 2019.

[31] N.-S. Vo, T.-M. Phan, M.-P. Bui, X.-K. Dang, N. T. Viet, and C. Yin, "Social-aware spectrum sharing and caching helper selection strategy optimized multicast video streaming in dense D2D 5G networks," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3480–3491, Sept, 2021.

[32] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Cache-aware multicast beamforming design for multicell multigroup multicast," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 11681–11693, 2018.

[33] X. Zhang, T. Lv, Y. Ren, W. Ni, N. C. Beaulieu, and Y. J. Guo, "Economical caching for scalable videos in cache-enabled heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 7, pp. 1608–1621, 2019.

[34] D. Jiang and Y. Cui, "Analysis and optimization of caching and multicasting for multi-quality videos in large-scale wireless networks," *IEEE Transactions on Communications*, vol. 67, no. 7, pp. 4913–4927, 2019.

[35] Z. Xu, Y. Cao, W. Wang, T. Jiang, and Q. Zhang, "Incentive mechanism for cooperative scalable video coding (SVC) multicast based on contract theory," *IEEE Transactions on Multimedia*, vol. 22, no. 2, pp. 445–458, 2020.

[36] A. De La Fuente, J. J. Escudero-Garzás, and A. García-Armada, "Radio resource allocation for multicast services based on multiple video layers," *IEEE Transactions on Broadcasting*, vol. 64, no. 3, pp. 695–708, 2018.

[37] C. Guo, Y. Cui, D. W. K. Ng, and Z. Liu, "Multi-quality multicast beamforming with scalable video coding," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5662–5677, 2018.

[38] Y. Jin, Y. Wen, and C. Westphal, "Optimal transcoding and caching for adaptive streaming in media cloud: an analytical approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 12, pp. 1914–1925, 2015.

[39] S. Chen, Z. Yao, X. Jiang, J. Yang, and L. Hanzo, "Multi-agent deep reinforcement learning-based cooperative edge caching for ultra-dense next-generation networks," *IEEE Transactions on Communications*, vol. 69, no. 4, pp. 2441–2456, 2021.

[40] S.-E. Elayoubi and J. Roberts, "Performance and cost effectiveness of caching in mobile access networks," in *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, pp. 79–88, 2015.

[41] D. Zegarra Rodríguez, R. Lopes Rosa, E. Costa Alfaia, J. Issy Abrahão, and G. Bressan, "Video quality metric for streaming service using DASH standard," *IEEE Transactions on Broadcasting*, vol. 62, no. 3, pp. 628–639, 2016.

[42] R. O. Afolabi, A. Dadlani, and K. Kim, "Multicast scheduling and resource allocation algorithms for OFDMA-based systems: a survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 240–254, 2013.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 2427155, 10 pages
https://doi.org/10.1155/2022/2427155

Research Article

Entity Relationship Modeling for IoT Data Fusion Driven by Dynamic Detecting Probe

Ye Tao,[1] **Shuaitong Guo**,[1] **Hui Li**,[1] **Ruichun Hou**,[2] **Xiangqian Ding**,[2] **and Dianhui Chu**[3]

[1]*College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, China*
[2]*College of Information Science and Engineering, Ocean University of China, Qingdao, China*
[3]*School of Computer Science and Technology, Harbin Institute of Technology (Weihai), Weihai, China*

Correspondence should be addressed to Ye Tao; ye.tao@qust.edu.cn

Received 13 March 2022; Accepted 6 May 2022; Published 8 July 2022

Academic Editor: Qi Li

To solve the problem of integrating and fusing scattered and heterogeneous data in the process of data space construction, we propose a novel entity association relationship modeling approach driven by dynamic detecting probes. By deploying acquisition units between the business logic layer and data access layer of different applications and dynamically collecting key information such as global data structure, related data, and access logs, the entity association model for enterprise data space is constructed from three levels: schema, instance, and log. At the schema association level, a multidimensional similarity discrimination algorithm combined with semantic analysis is used to achieve the rapid fusion of similar entities; at the instance association level, a combination of feature vector-based similarity analysis and deep learning is used to complete the association matching of different entities for structured data such as numeric and character data and unstructured data such as long text data; at the log association level, the association between different entities and attributes is established by analyzing the equivalence relationships in the data access logs. In addition, to address the uncertainty problem in the association construction process, a fuzzy logic-based inference model is applied to obtain the final entity association construction scheme.

1. Introduction

Data become an important resource in the information era. For different application scenarios, data can be stored in either centralized or distributed environment. It becomes particularly important to discover the association and correlation among heterogenous data source from multiple domains [1, 2]. After data are collected intensively, there are problems such as low sharing of original information, disconnection between information, and business processes and applications, which can easily lead to the formation of information silos [3]. In particular, the IoT industry requires huge technical data support to realize the procedural industrial processes and technologies, such as the construction of smart cities. This needs to solve the problem of information silos to achieve data sharing and fusion of centrally collected data under the premise of ensuring data security [4, 5]. To explore the correlation between data, some enterprises have started to build data space to integrate the data collected centrally, to eliminate information silos.

From the early days of data warehouses, data lakes, to the today's data fabric and data space systems, connecting data entities plays a vital role in data analysis. In the past, data association operations usually required cooperation between business-related personnel and database administrators to complete, which usually meant a lot of labor, material, and time, and its scalability was poor, and once the data changed, the data association information needed to be generated again. There is also a lot of research in academia on how to generate correlation information between data quickly and accurately. Current research results mainly focus on discovering associations between entities or attributes through the semantic matching of dictionaries or semantic libraries, using data representation, or content similarity judgments [6], and then using plain Bayesian learning algorithms to calculate the probability of similarity between data. Many of these methods have poor generalizability, slow response, and low accuracy when attempting to discover the existence of associations from a large amount of data.

In this paper, we propose a new approach to discover entity association relationships in big data. First, this approach obtains business logic information and database data through dynamic probes deployed between the business logic layer and data access layer of different systems. Then, it portrays the similarity degree among entities in three dimensions, schema, instance, and log and gives the similarity values among entities in these different dimensions. Finally, based on the fuzzy logic inference method [7, 8], the similarity values among entities in different dimensions are converted into normalized values that can be uniformly measured to obtain the best matching results of entity association. Thus, heterogeneous data from multiple source databases are integrated into a comprehensive enterprise data space through entity matching.

2. Related Work

In academic research, entity association is mainly divided into two types: schema matching [9] and instance analysis [10]. Schema matching extracts structural features from data sources as metadata and analyzes them to achieve association matching between data with fewer resources; matching based on instance analysis analyzes the data itself to obtain matching information, which usually consumes more resources but can obtain more accurate and comprehensive analysis results.

For schema matching academic research [11], Gomes dos Reis et al. used Structured Query Language (SQL) to extract features such as the name of the database, name of the schema, and type of column as metadata sets from each selected dataset. Then, they joined all metadata from each dataset into a metadata database. Finally, the correlation between the metadata was calculated by different methods to establish an association between the source data.

In addition to building a database through metadata [12], Berlin et al. use plain Bayesian learning to populate the attribute dictionary with example values provided by domain experts. To make efficient use of the attribute fields stored in the database, they employ statistical feature selection techniques to learn an efficient representation of the examples. With some columns of operations, the optimization process, which is based on a minimum cost maximum flow network algorithm, finds the overall optimal match between the two customer schemas based on the sum of the individual attribute matching scores.

For the ontology semantic similarity problem in schema matching, Meng et al. [13] studied the semantic similarity model based on distance, information content, and attributes and discovered that converting words to concept words in ontologies and performing semantic similarity calculations that can deliver the precise and effective measurement of targeted ontology semantics in a domain, which improves the accuracy of ontology semantic analysis in schema matching.

We can find that schema matching can effectively distinguish the association between data according to the analyzed information when processing a small amount of data, and the processing speed does not change significantly with the change of data volume because the analyzed elements are fixed, and the association matching between data can be achieved with less resources. However, when the amount of data grows exponentially, the probability that model information of different categories of data is similar or identical increases sharply because the amount of pattern information is certain, leading to a weakening of the differentiation effect of pattern matching analysis data.

In academic research on instance analysis, the preprocessing that mines associations between data include categorized data. For example, for the data conflict problem in data fusion, in [14], the conflict can be divided into two categories: uncertain conflict and contradictory conflict, and then the duplicate data of the same representation are fused, thus solving problems such as the possible conflict between different values for the same attribute. Reference [15] proposes that solutions such as name and description matching in schema matching can be used for element-level instance analysis. Addressing the problem of different data types in instances, the authors in [16, 17] propose an approach for classifying instance data and present a systematic theoretical framework for establishing data associations for different classes of data. Instance analysis can maintain a better differentiation of data fusion when dealing with large amounts of data, but this often takes a long analysis time. In addition, instance analysis often takes a lot of time and operational resources to correct data association relationships when data change, especially when new data are added.

Academics are also studying the integration of deep learning with logs, for example, using deep learning to replace statistical methods in logs that portray associations between users and certain types of items or certain things. Mohanty et al. [18] cleaned the web log files collected by the IoT, built user profiles, saved similar information, and proposed a recommendation system based on rough fuzzy clustering to recommend e-commerce shopping sites to users. The logs contain correlations among the data, but they are generated by manipulating the data, and only part of the data are involved compared to the overall data, leading to a lack of completeness, and their analysis is unable to explore the correlations that exist in all the data. In this paper, we offer a proposal for extracting the data association information in logs and using it as a basis for matching entity associations in multisource data to construct entity associations in enterprise data space; this approach builds on the feature that logs contain association information between data [19].

The constantly increasing amount of data accumulating in the development of enterprises leads to an increasing size and number of categories of data, and methods such as schema matching, instance analysis, and log mining to analyze data from a single dimension may have problems such as not making full use of the diversity of data or incomplete analysis. Addressing the above issues, this paper analyzes the data from multiple dimensions by integrating schemas, instances, and logs to make full use of the diversity of data to establish entity associations.

3. Our Customized Framework

The entity association model in Figure 1 shows the mapping relationship between multiple sources of data from different departments in the enterprise business system and the data space. According to the multidimensional analysis framework proposed in this paper, normalized similarity values between data that can be compared are obtained to establish the association relationship between entities. As shown in Figure 1, R_1 indicates a similarity value of 1 between its associated entities a_{13} and n_{11}.

In the middle of the business logic layer and data access layer of each business system, such as enterprise resource planning (ERP), customer relationship management (CRM), and software configuration management (SCM), we deploy probes to obtain data. Then, the business logic layer of the data is stored as logs, and the rest of the data are stored in a relational database. To overcome the problems of large size and a variety of data types, the model preclassifies the data based on their characteristics and nature, which improves the data processing and increases the accuracy of matching between entities. The structure and content of the data are divided into two categories: schema and instance, while logs as a carrier of business logic are grouped into a separate category. The similarity values between the data are analyzed and calculated in three dimensions: schema, instance, and log. The schema matching analysis includes both attribute names and constraints, and the instance analysis is divided into three analysis methods according to data type: numeric, character, and long text. Based on the attribute association information contained in SQL, the log analysis calculates the similarity values among the data. Finally, based on the fuzzy logic analyzer, a normalization calculation is performed based on similar values for the data in different dimensions to obtain the effective association values in the data space. The corresponding schema is shown in Figure 2.

3.1. Schema Similarity Model. Many different databases are developed by database designers to fit application scenarios, naming conventions, and other factors, but database designs generally contain table and field names, table structures, and data types. As such, the attribute names and constraints of the schema information in the database are extracted as the analysis content of the schema similarity model to measure the similarity between the data.

3.1.1. Name of Attribute. Attribute name analysis is divided into two types: plain text similarity and text semantic similarity analysis. The text similarity between attribute names is calculated by the edit distance algorithm, and text semantic similarity is calculated through a semantic library.

Edit distance is a way of quantifying how similar two strings are; it takes two words w_1 and w_2 and finds the minimum number of operations required to convert w_1 to w_2. The plain text similarity value is defined according to the minimum number of edits, as shown in the following equation:

$$S_{\text{plain}}(w_1, w_2) = 1 - \frac{D(w_1, w_2)}{\text{Max}(l_1, l_2)}, \tag{1}$$

where l_1 and l_2 are the character lengths of w_1 and w_2 and D is the edit distance of w_1 and w_2.

Different expressions may be used for the description of the same entity. For example, if the information of an upstream company is recorded in the enterprise database, its attribute name can be named CompanyID and SupplierID based on different scenarios. To address the fact that plain text analysis cannot resolve the semantics between words, a semantic-based similarity analysis method is proposed. In particular, a tree semantic hierarchy is established for the attribute names, as shown in Figure 3, and the similarity between words is calculated by the corresponding positions of the attribute names in the tree diagram.

Therefore, the equation calculating the semantic-based similarity is

$$S_{\text{sema}}(w_1, w_2) = \frac{2H}{N_1 + N_2 + 2H}, \tag{2}$$

where N_1 and N_2 denote the shortest paths from words w_1 and w_2 to the nearest common parent word w, respectively, and H denotes the shortest path from w to the root node.

S_{name} is defined as the maximum of the plain text similarity and the semantic similarity of the text, as shown in the following equation:

$$S_{\text{name}} = \text{Max}\left(S_{\text{plain}}, S_{\text{sema}}\right). \tag{3}$$

3.1.2. Constraint. Designers follow certain principles when programming columns in a database, such as the appropriate data type and whether it is empty. The representative constraints selected from these rules can be used to explore the similarity among columns. Constraints listed in Table 1 are extracted as features: type of each column, if the column is a primary or foreign key or not if the column has constraint of null or not null if the column has comments.

In the following equation, we assume that the two columns requiring constraint similarity discrimination are A and B, and a_i and b_i are the values of the ith candidate constraint corresponding to the attributes of the two columns, respectively, such that

$$v_i = \begin{cases} 1 & a_i = b_i \\ 0 & \text{otherwise} \end{cases}, \quad i = 1, 2, \ldots, n, \tag{4}$$

where n is the number of candidate constraints. Therefore, the attribute constraint similarity between column A and column B is calculated by

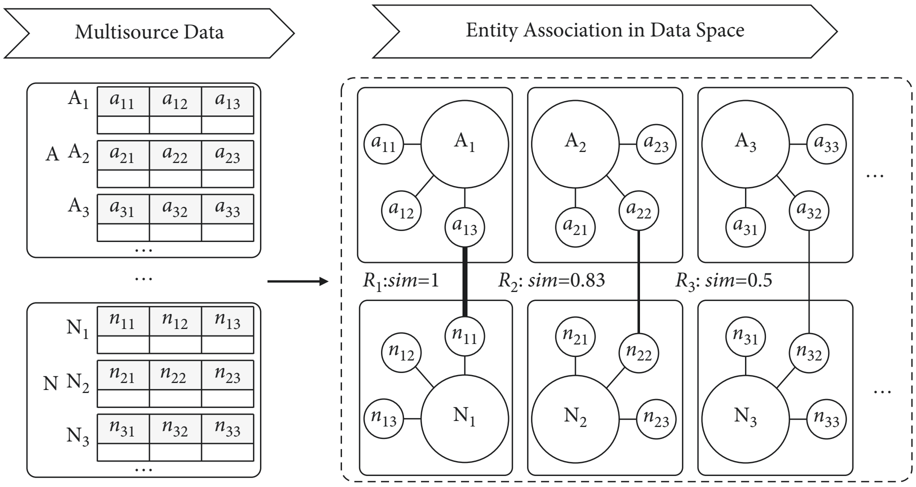

Multisource Data
Entity Association in Data Space
A
A_1 a_{11} a_{12} a_{13}
A_2 a_{21} a_{22} a_{23}
A_3 a_{31} a_{32} a_{33}
N
N_1 n_{11} n_{12} n_{13}
N_2 n_{21} n_{22} n_{23}
N_3 n_{31} n_{32} n_{33}
R_1:sim=1
R_2: sim=0.83
R_3: sim=0.5
…

Figure 1: Entity association mapping for multisource data.

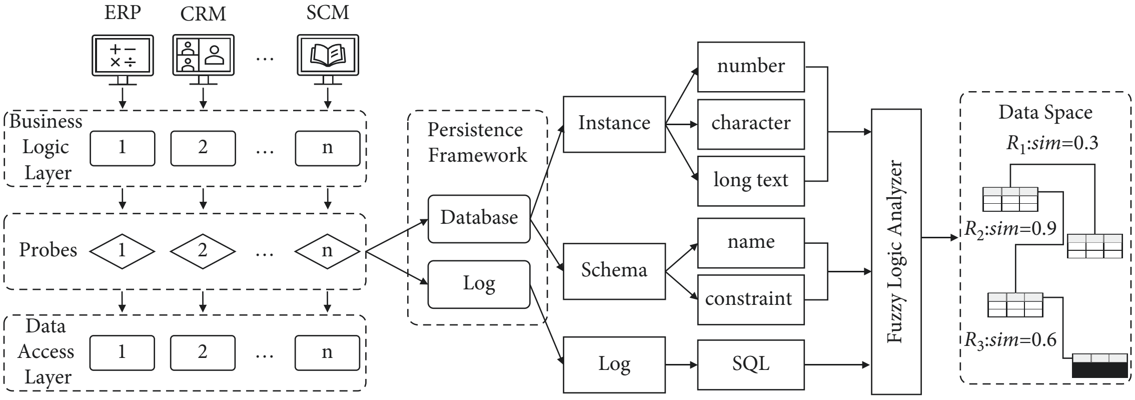


Figure 2: A logical framework for multisource data analysis.

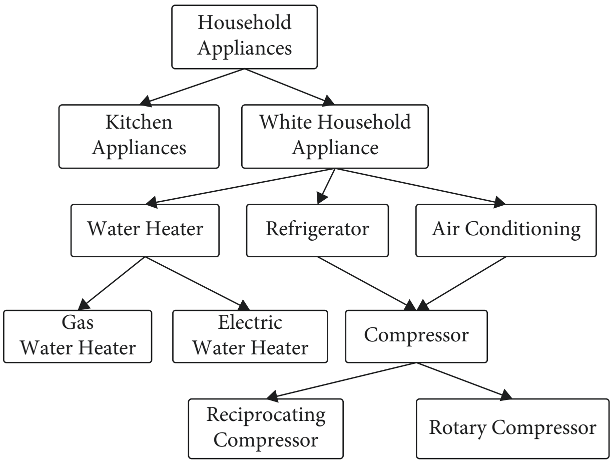


Figure 3: Attribute name tree semantic hierarchy diagram.

Table 1: Constraint features.

$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
Type of column	Null	Primary key	Foreign key	Comments

$$S_{\text{cons}} = \frac{\sum_i v_i}{n}. \tag{5}$$

3.1.3. Schema Similarity. S_{schema} includes attribute names and constraint analysis of similar values by weighting, as shown in the following equation:

$$S_{\text{schema}} = \alpha S_{\text{name}} + (1-\alpha) S_{\text{cons}} \, (\alpha \in [0,1]). \tag{6}$$

3.2. Instance Similarity Model. Since there are similarity trends in datasets representing similar entities, such as value intervals, extreme values, and keywords. Similarity relationships between data can be established by the main content of the dataset. It is obvious that data categories have distinctive features of a dataset, and differences in data categories lead to variability in the attributes chosen to characterize the dataset. Establishing differentiated feature extraction schemes for different classes of datasets can improve the accuracy of data association matching. The data types in the database are categorized, and different categories of data correspond to different processing schemes; generally, if the data categories are different, there is no similar relationship.

According to the different data types, instance analysis can be divided into the following three types: numeric, character, and long text. The numeric type refers to the exact numeric data type and the approximate numeric data types in Table 2. The string data types are divided into two categories, character, and long text, according to the length of the text. After classifying and clustering the data, the similarities between the data are analyzed according to the process shown in Figure 4.

3.2.1. Number. For scalar data, the similarity between columns can be evaluated from the perspective of numerical distribution, e.g. the median, mean, variance and etc. In order to reflect the characteristics of numerical scalars from different aspects, the selection of features is focused on the following three aspects, the maximum and minimum values that can define the range of data, the mean, arithmetic median and plural that reflect the main distribution of data, the sample standard deviation that can reflect the degree of dispersion of data, these indicator elements are not sensitive to the change of data volume and can be used as the feature elements for calculating column similarity, while the number of non-null values and the cumulative sum of the data do not change significantly with the change of data volume, but are not suitable as feature elements. Finally, the feature vector corresponding to each column is calculated, substituted into the cosine similarity formula, and the result is used as the numerical similarity value.

3.2.2. Character. Character is short textual content, and it uses the term frequency-inverse document frequency as the similarity calculation algorithm. First, the content of the columns that need to determine similarity is combined as a separate dataset. Then, the vectors for each column are found. Finally, the feature vectors are substituted into the cosine similarity formula to calculate the similarity value.

3.2.3. Long Text. Long text is long text content, where the records in the columns are mapped as vectors, a model is built using an autoencoder, and the similarity values among columns are calculated based on the model. Assuming that A and B are the two columns in the database, and they share the long text data type (Figure 5). The overfitting problem of the model due to the large difference in the number of datasets is solved by randomly selecting k records in columns A and B as the sample data sets S_1 and S_2. Since vectors are required as input for the autoencoder, the text in the sample data sets is transformed into vectors $\overrightarrow{U}$ and $\overrightarrow{V}$. Then, the vectors are divided into a training set and test set, the autoencoder model is built using the training set, and the similarity of columns A and B is calculated according to the accuracy of the test set.

Table 2: Data type categorization.

Data type	Members
Exact numeric data type	Smallint, mediumint, int, bigint
Approximate numeric data type	Float, double, decimal
String data types	Char, varchar, blob, text

The autoencoder model calculates similarity, as shown in Algorithm 1. For input, x is divided into a training set and a test set according to a custom scale, y is used as the test set, and ω is the custom text similarity threshold. On output, λ and θ are the percentages of the test set evaluated as similar. For autoencoder 1, x and y for the input in Algorithm 1 are $\overrightarrow{U}$ in vector space and the test dataset of $\overrightarrow{V}$ in vector space, and the output is λ_1 and θ_1. For autoencoder 2, x and y for the input in Algorithm 1 are $\overrightarrow{V}$ in vector space and the test dataset of $\overrightarrow{U}$ in vector space, and the output is λ_2 and θ_2. According to the results obtained from the autoencoder, S_{long} represents two columns of similar values, as shown in the following equation:

$$S_{\text{long}} = \text{Min}\left(\frac{\theta_1}{\lambda_1}, \frac{\theta_2}{\lambda_2}, 1\right). \tag{7}$$

3.3. Log Similarity Model. The business logic layer in the layered architecture mainly packages the attributes and behaviors of entities. Although the representation of entities varies across different business logics and similar entities have similar attributes and behaviors. The SQL commands recorded in the logs contain correlation relationships among columns, which can be used as a basis of analysis for measuring column similarity. The column-to-column similarity can be obtained by counting the number of equivalence relations in the log file.

In the following equation, we assume that A and B are columns in the database, and the log similarity value of columns A and B is calculated by

$$S_{\log} = \frac{N_{ab}}{N_a + N_b}, \tag{8}$$

where a and b are the names of columns A and B, N_a and N_b are the number of SQL commands containing a and b in the log, and N_{ab} is the number of SQL commands containing both a and b in the log.

3.4. Fuzzy Logic Similarity. According to the previous section, the calculation of the data with the proposed model can obtain similar values in three dimensions: pattern, instance, and log, which need to be unified into directly comparable values since similar values on different dimensions are not directly comparable. The methods that can generally be used to convert multidimensional values into a single value are the Delphi method [20], weighted average, and fuzzy logic. Delphi method relies on domain-specific knowledge, and when the data source is not regular, it cannot be well adapted to the data, while weighted average, due to its fixed form, is

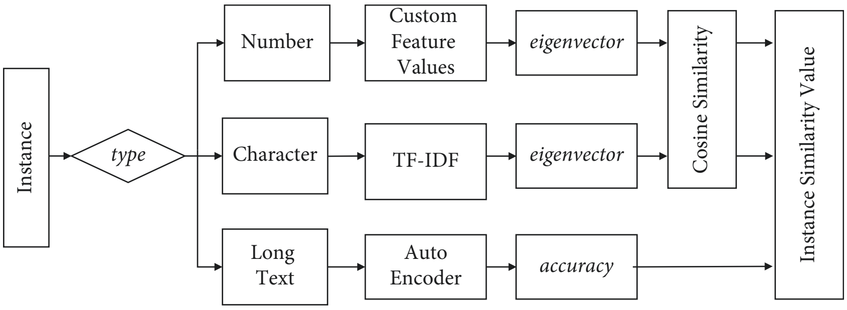


Figure 4: Instance analyzer.

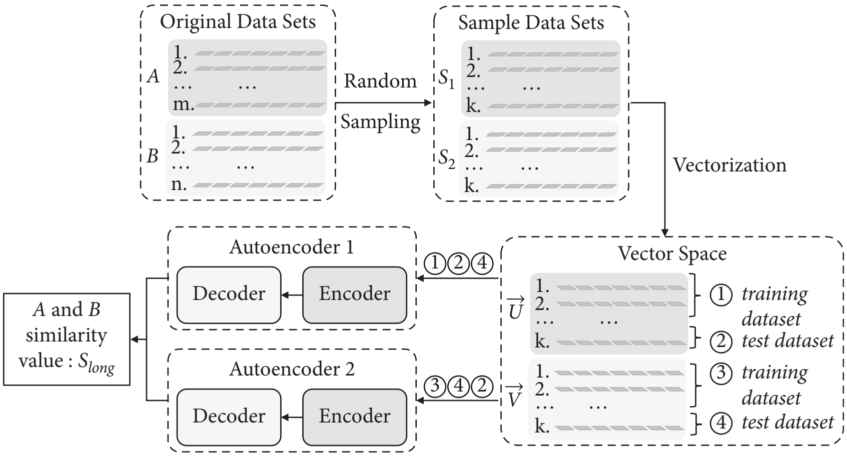


Figure 5: The long text analysis process.

```
Input : $x, y, \omega$
Output : $\lambda, \theta$
$a$_train, $a$_test←train_test_split$(x)$
$b$_test←$y$
$a$_num, $b$_num←len($a$_test), len($b$_test)
input←$a$_train
encoded = Dense(input)
decoded = Dense(encoded)
autoencoded = Model(input, decoded)
$a$_test_predict = autoencoded($a$_test)
$b$_test_predict = autoencoded($b$_test)
    For$a, b$in$a$_test, $a$_test_predict
        $s$_$a$_num + +←similarity$(a, b) \geq \omega$
    $\lambda$←num/$s$_$a$_num
    For$a, b$in$b$_test, $b$_test_predict
        $s$_$b$_num + +←similarity$(a, b) \geq \omega$
    $\theta$←num/$s$_$b$_num
```

Algorithm 1: Long text similarity calculation method.

more homogeneous for data processing and cannot make full use of the characteristics of the data. In contrast, fuzzy logic can contain expert domain knowledge [21] and its ability to use multiple functions for data fitting when processing data. Its adaptability is relatively good, so fuzzy logic is chosen to normalize the similar values of multiple dimensions.

The similarity values obtained from the above calculation by schema, instance, and log similarity models are processed using fuzzy logic for standardization. While A and B are the columns in the database, the similarity values obtained from the above three-dimensional analysis are substituted into the affiliation function to obtain the affiliation values. The values that meet the fuzzy rules are aggregated according to the rules and defuzzified to obtain a normalized measure of column-to-column similarity. In Figure 6, for example, A and B have similar values of 0.6, 0.7, and 0.8 in the schema, instance, and log dimensions. Through a series of fuzzy operations, the similarity value between A and B is 0.71.

4. Experiment

To verify the feasibility of the proposed framework, this paper uses data from all business systems of a company and stores them in a unified manner. The hardware environment for the experiments is an Intel(R) Xeon(R) Silver 4210 CPU @ 2.20 GHz, 64 GB RAM, and RTX2080Ti*4. The results are the average of three replicated experiments. The dataset consists of Haier, the upstream and downstream of Haier's supply chain, and public data set available on the Internet [22]. It mainly includes the following categories: product data, enterprise operation data, value chain data, and external data.

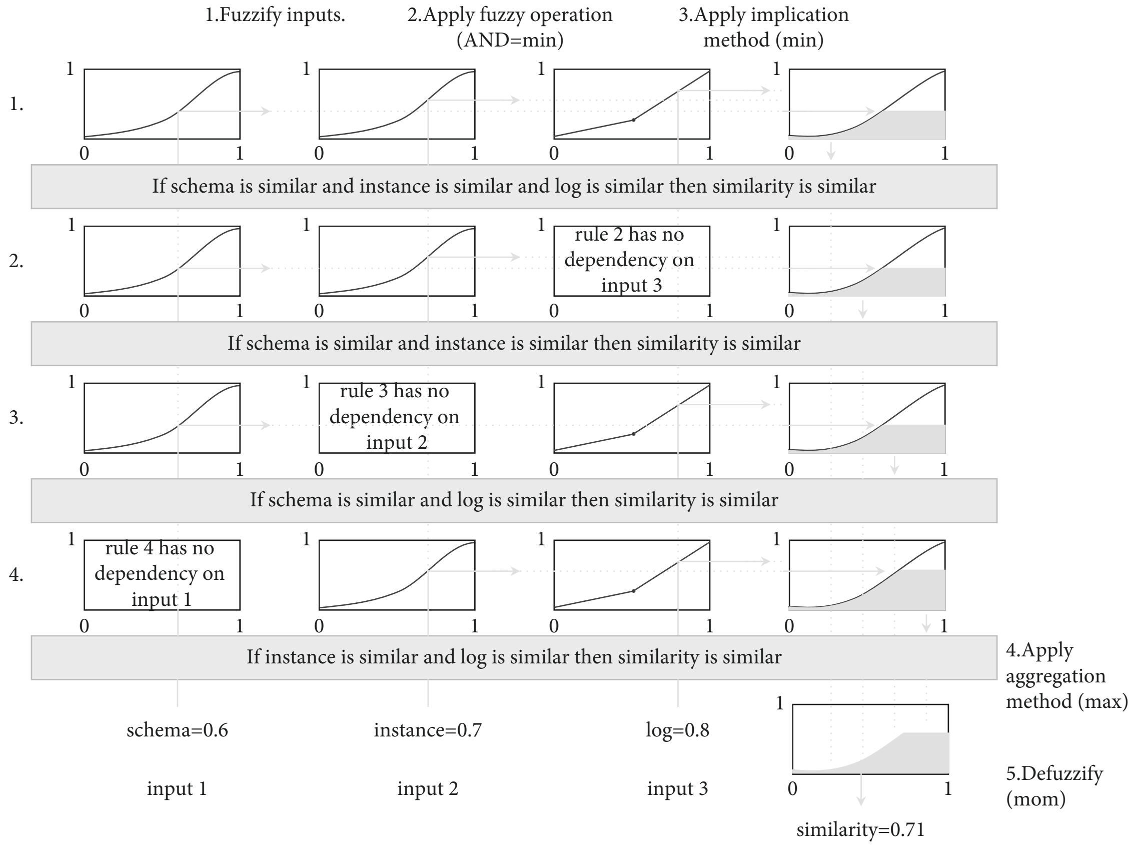

1.Fuzzify inputs.
2.Apply fuzzy operation (AND=min)
3.Apply implication method (min)
1.
If schema is similar and instance is similar and log is similar then similarity is similar
2.
rule 2 has no dependency on input 3
If schema is similar and instance is similar then similarity is similar
3.
rule 3 has no dependency on input 2
If schema is similar and log is similar then similarity is similar
4.
rule 4 has no dependency on input 1
If instance is similar and log is similar then similarity is similar
4.Apply aggregation method (max)
schema=0.6
instance=0.7
log=0.8
input 1
input 2
input 3
5.Defuzzify (mom)
similarity=0.71
0
1

Figure 6: Fuzzy logic instance diagram.

4.1. Data Aggregation Matching Experiments. Data aggregation matching experiment is as follows: the data provided by the suppliers are analyzed for correlations between the data using the model designed in this section, and the data are automatically imported into the summary table (the attributes to be imported and their corresponding partial data need to exist in the summary table in advance). Table 3 shows the matching results of 3000 records in the selected supplier data, respectively, where each row is the matching result information of each supplier. The total number of attributes refers to the total number of data attributes provided by suppliers, the total number of valid attributes refers to the data that can correspond to a column attribute in the summary file, and the total number of correctly associated attributes refers to the number of columns that are correctly integrated into the summary file after matching each supplier's data through the model. The correct rate is the ratio of the number of correctly associated attributes to the total number of valid attributes.

From the results, we can see that the best performance of data matching accuracy can reach about 89%, which can be well used as an auxiliary tool for data matching, while the analysis of the results of the lower accuracy of data matching for no. 2 reveals that more proprietary names and abbreviations are used in the data provided by its suppliers, and because its business involves relatively single, the content similarity is high, which leads to the low accuracy of pattern matching results in model analysis, thus leading to unsatisfactory results, and the accuracy can be subsequently improved by optimizing the semantic analysis in pattern matching.

4.2. Comparison of Experiments for the Different Solutions of Data Space Entity Association. A certain number of columns are randomly selected as samples from all data, and experiments are conducted using schema matching (see Section 3.1), instance analysis (see Section 3.2), and the fuzzy logic-based model proposed in this paper to compare them in two ways: running time and accuracy.

The Design of Experiment. (1) Runtime with different methods: from a total of 3702 columns, randomly select 400, 600, . . ., 2400 columns, 11 groups in total, record the running time of each set of data under the three methods, and repeat the above operation three times.Figure 7(a) shows the average runtime with different methods. (2) Accuracy with different methods: a total of 3702 columns are grouped according to numbers, characters, and long text; and 50 pairs of related columns are randomly selected from each group. 5000, 10000, . . ., 55000 rows were selected from the selected columns; 11 groups in total; and the similarity value of each group of data under the above three models was calculated to determine whether the prediction was correct according to the threshold value w. The percentage of

Table 3: Data matching results.

No.	Number of attributes	Number of valid attributes	Number of valid attributes correctly associated	Accuracy rate (%)
1	1847	942	749	79.51
2	2084	642	411	64.02
3	1974	762	647	84.91
4	1639	849	758	89.40

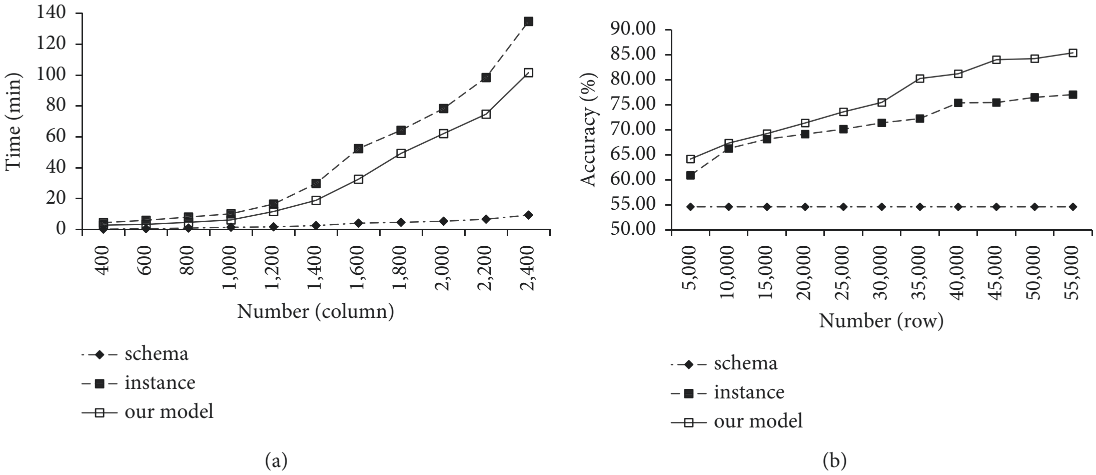


Figure 7: Comparison of experiments for different methods: (a) running time of different methods and (b) accuracy of different methods.

correct prediction is calculated, and Figure 7(b) shows the average prediction accuracy of the above three methods.

Analysis of the Experiment. Experiments based on the schema take less time, as shown in Figure 7(a). The instance-based method takes significantly more time for the same amount of data due to the comprehensive content analysis, while the method proposed in this paper includes instance analysis but takes less time than the instance-based method because the data are analyzed in categories during the instance analysis.

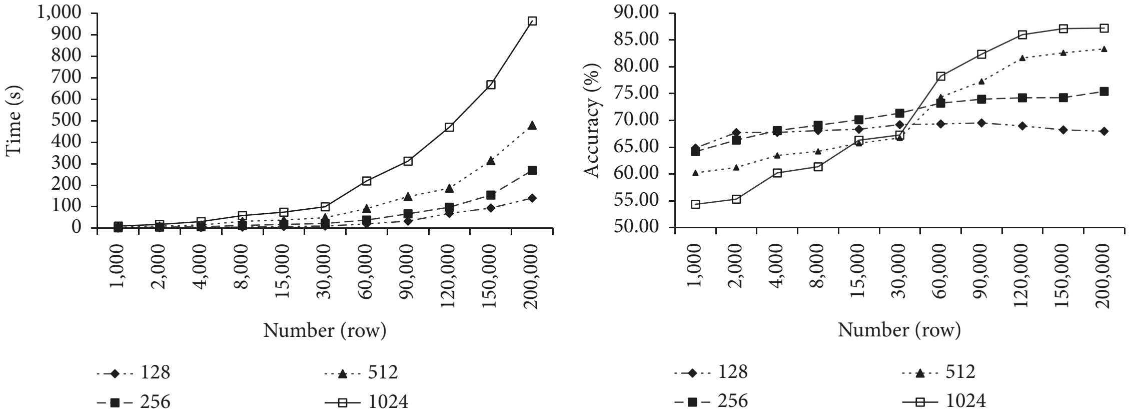
The accuracy of the schema-based method was highest when the experimental sample was below 600, as seen in Figure 7(b). The method proposed in this paper maintained the highest accuracy after 800 columns, the schema-based method was limited after the data volume was 1400 columns due to the limited analysis elements, and the data matching accuracy decreased due to the increase of homogeneous data caused by easily mismatching events when the data size became larger. Overall, with the increase of data volume, the data matching accuracy of all analysis methods tends to increase, which is due to the fact that, in equal proportion sampling, when the sample is small, the number of similar data corresponding to the suppliers is smaller, which leads to the possibility of mismatching; and when the proportion of sampled data covering the overall data increases, the mismatching situation decreases significantly, and thus the correct rate of data matching gradually increases.

As shown in Figure 7, the proposed method in this paper can obtain a high accuracy rate in a short time with a moderate amount of data.

4.3. Long Text Validation Experiments. To study the performance of the autoencoder on the long text case in the instance analysis, two columns of associated long text are selected, and the performance of the model proposed in this paper is observed in different cases by changing the vector dimension.

The Design of Experiment. From the existing long text columns, 10 pairs with suitable amount of data and correlation were selected, and the running time and prediction accuracy under long text analysis were recorded by changing their vectorized coding length to 128, 256, 512, and 1024, and the results are shown in Figure 8.

Analysis of the Experiment. The higher the dimensionality is, the more time the experiment takes for the same amount of data as shown in Figure 8(a). Figure 8(b) shows that in the case of a small volume of data, if the dimensionality is too high, it will reduce the accuracy. The reason for this performance can be found by analyzing the principle of the autoencoder. The autoencoder model reduces the dimension of data to extract key information, and when the data size is small, the compressed extracted data features in the long text are limited, so the high-dimensional feature vector will be mixed with a large amount of noisy data, which results in a low accuracy rate. As the volume of data increases, more data features can be extracted from long text, and the high-dimensional feature vector can represent the text better and therefore obtain higher accuracy.



(a)

(b)

Figure 8: Comparisons of experiments for long text columns: (a) running time of different vector dimensions; (b) accuracy of different vector dimensions.

5. Conclusion

This paper proposes a hybrid data matching model based on schema, instances, and logs. The model consists of four main components: the front probe to acquire the analysis data, the analysis data, the three-dimensional outputs, and the normalized metric based on fuzzy logic. Experimental results show that the model provided in this paper has better results in terms of accuracy and efficient handling of mass data compared to previous single matching methods based on schema or instances. For further research, the focus is on how to establish a mapping relationship between data and weights and on establishing a guidance scheme for weight assignment to better address the impact of the randomness of multisource heterogeneous data on the accuracy of the results.

Data Availability

The data are generated by the actual operation process of the enterprise and involve private data that cannot be desensitized and can only be used internally for the time being.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1702902 and in part by the Shandong Province Colleges and Universities Young Talents Initiation Program under Grant 2019KJN047.

References

[1] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, “A blockchain-based machine learning framework for edge services in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, 2022.

[2] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, “Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles,” *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.

[3] J. Lv, K. Shen, S. Johnson, F. Chen, and G. Li, “Application on information island with information visualization and software engineering,” in *Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI)*, pp. 598–603, Nanjing, China, November 2018.

[4] Y. Tian, Z. Zhang, J. Xiong, L. Chen, J. Ma, and C. Peng, “Achieving graph clustering privacy preservation based on structure entropy in social IoT,” *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2761–2777, 2022.

[5] J. Xiong, R. Ma, L. Chen et al., “A personalized privacy protection framework for mobile crowdsensing in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[6] M. Nordin, A. Alzeber, and A. Zaid, “A survey of schema matching research using database schemas and instances,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 10, 2017.

[7] X. Li, H. Wen, Y. Hu, and L. Jiang, “A novel beta parameter based fuzzy-logic controller for photovoltaic MPPT application,” *Renewable Energy*, vol. 130, pp. 416–427, 2019.

[8] Z. Roumila, D. Rekioua, and T. Rekioua, “Energy management based fuzzy logic controller of hybrid system wind/photovoltaic/diesel with storage battery,” *International Journal of Hydrogen Energy*, vol. 42, no. 30, pp. 19525–19535, 2017.

[9] W. Tan and A. Mapforce, *Approximation Algorithms for Schema-Mapping Discovery*, vol. 42, 2017.

[10] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, “Multi-instance learning with discriminative bag mapping,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1065–1080, 2018.

[11] D. Gomes dos Reis, M. Ladeira, M. Holanda, and M. de Carvalho Victorino, “Large database schema matching using data mining techniques,” in *Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW) 2018*, pp. 523–530, Singapore, November 2019.

[12] J. Berlin and A. Motro, “Database schema matching using machine learning with feature selection,” *Notes on Numerical*

Fluid Mechanics and Multidisciplinary Design, pp. 452–466, Springer, Berlin, Germany, 2002.

[13] L. Meng, R. Huang, and J. Gu, "A review of semantic similarity measures in WordNet," *Int. J. Hybrid Inf. Technol.*vol. 6, pp. 1–12, 2013.

[14] A. Bakhtouchi, "Data reconciliation and fusion methods: a survey," *Applied Computing and Informatics*, vol. 18, no. 3/4, pp. 182–194, 2022.

[15] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," *The VLDB Journal*, vol. 10, no. 4, pp. 334–350, 2001.

[16] X. Xu and W. Wang, "Attribute identification between spatial datasets based on instance statistical similarities," in *Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing 2008*, pp. 1–5, Dalian, China, October 2008.

[17] E. Sutanta, R. Wardoyo, K. Mustofa, and E. Winarko, "A hybrid model schema matching using constraint-based and instance-based," *International Journal of Electrical and Computer Engineering*, vol. 6, no. 3, pp. 1048–1058, 2016.

[18] S. N. Mohanty, J. Rejina Parvin, K. Vinoth Kumar, K. C. Ramya, S. Sheeba Rani, and S. K. Lakshmanaprabu, "Optimal rough fuzzy clustering for user profile ontology based web page recommendation analysis," *Journal of Intelligent and Fuzzy Systems*, vol. 37, no. 1, pp. 205–216, 2019.

[19] Y. Tao, S. Guo, C. Shi, and D. Chu, "User behavior analysis by cross-domain log data fusion," *IEEE Access*, vol. 8, pp. 400–406, 2020.

[20] I. Belton, A. MacDonald, G. Wright, and I. Hamlin, "Improving the practical application of the Delphi method in group-based judgment: a six-step prescription for a well-founded and defensible process," *Technological Forecasting and Social Change*, vol. 147, pp. 72–82, 2019.

[21] C.-L. Wu, T.-W. Ke, and T.-H. Meen, "Evaluation of intensified colorectal cancer treatment using model based on Delphi method, fuzzy logic, and analytical hierarchy process (DFAHP)," *Sensors and Materials*, vol. 33, no. 10, pp. 3499–3512, 2021.

[22] Industrial-Datasets, "Haier's internal dataset and publicly available datasets on the Internet," 2021, https://github.com/forgstfree/Industrial-Datasets.

[23] Y. Tao, S. Guo, R. Hou, X. Ding, and D. Chu, "Entity relationship modeling for enterprise data space construction driven by a dynamic detecting probe," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, J. Xiong, S. Wu, C. Peng, and Y. Tian, Eds., pp. 185–196, Springer International Publishing, New York, NY, USA, 2021.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 2206917, 14 pages
https://doi.org/10.1155/2022/2206917

Research Article

MSAAM: A Multiscale Adaptive Attention Module for IoT Malware Detection and Family Classification

Changguang Wang ,[1,2] Ziqiu Zhao,[2] Fangwei Wang ,[1,2] and Qingru Li [1,2]

[1]*College of Computer & Cyber Security, Hebei Normal University, Shijiazhuang 050024, China*
[2]*Key Laboratory of Network & Information Security of Hebei Province, Hebei Normal University, Shijiazhuang 050024, China*

Correspondence should be addressed to Fangwei Wang; fw_wang@hebtu.edu.cn and Qingru Li; qingruli@hebtu.edu.cn

Received 15 February 2022; Revised 16 May 2022; Accepted 6 June 2022; Published 21 June 2022

Academic Editor: Jinbo Xiong

Nowadays, the attack and defense of malware have presented asymmetric characteristic threats, which has disrupted the pace of IoT research. Traditional detection and family classification methods based on feature extraction, as well as the classical machine learning algorithms, have been afflicted with the problems of high time consuming and unbalanced numbers of malware samples. This paper designs a universal and effective Multiscale Attention Adaptive Module called MSAAM that can combine local and global feature information. It can automatically adjust the arrangement and proportion of channel and spatial submodules by auxiliary classifiers according to actual tasks. The traditional CliqueNet uses a circular feedback structure to improve the DenseNet, optimizes the information flow in a deep network, enhances the utilization of its parameters, and uses a multiscale strategy to prevent a sharp increase of its parameters. As a result, it shows a good effect in the study of image classification. By replacing the attention module in the traditional CliqueNet with the designed MSAAM, we present a new method to process the produced gray-scale images converted from the malware and thus get better results in malware processing. The improved CliqueNet runs on the benchmark datasets of MalImg and Microsoft's BIG 2015 to verify our presented method. After validation on the experimental benchmark datasets, the detection accuracy reaches 99.8%, while the family classification accuracy reaches 99.2% and 98.2% on the above two datasets, respectively. The presented method can solve the problem of unbalanced samples in malware family classification and is also effective against obfuscation attacks.

1. Introduction

Malware refers to a computer code that is written or set up deliberately to pose a threat or potential threat to a network or system. Malware families consist of crypto miners, viruses, ransomware, worms, and spyware, whose purposes are mainly an illegal gathering of information, obstruction of services, or espionage. Nowadays, the popularization of IoT, 5G, and cloud computing increases not only the number and types of malware, but also their threat scope and targets. The McAfee Labs 2021 Threats Report [1] states that Q3 and Q4 of 2020 averaged 588 and 648 malware-related security issues per minute. The two-quarters increased by 169 (40%) and 60 (10%) per minute, respectively, compared to the last quarter. From the third quarter to the fourth quarter, Office malware surged by 199%. Attackers only need to focus on one vulnerability to achieve their goals, but defenders need to patch all vulnerabilities in time to ensure IoT security. This asymmetry is the difficulty faced by malware processing research. And with the increasing application fields affected by malware [2–4], the pressure on IoT security research increases dramatically. Although cybersecurity research continues to grapple with malware threats, malware developers continue to develop new ways to bypass the existing defenses.

Traditional malware processing techniques include static analysis and dynamic analysis. Static analysis means intercepting and analyzing the features such as opcodes, system calls, and Application Programming Interfaces (APIs) from the entire software without executing it. A new malware identification system is proposed according to the frequency of opcodes in portable executable files [5]. The API call sequence can represent the target's actions and then be used to classify the malware families [6]. Based on API intimacy analysis, their system selects the graph method to analyze the network process and can detect Android malware with high efficiency and precision [7]. The dynamic analysis method enables and controls the running behavior of the target in a closed environment to intercept the behavior characteristics of the target, such as registry modification, system call, file operation, and so on. From five-minute API activities, features are input to a CNN for malware family differentiation [8]. Obfuscated malware is flagged with hooks placed in the system to calculate the time length consumed by the kernel and users [9]. Several feature sets are configured for dynamic malware analysis by extracting features from API calls, and statistical inference is applied to find a set of feature sets that can correctly characterize samples [10]. Nevertheless, both styles of analysis implicit their disadvantages. It is difficult for static analysis with disassembly as the main method to solve the malware that uses various obfuscation techniques for complex reverse engineering. Dynamic analysis has the risk of affecting the system due to the need to actually enable the target. The disassembly technology used in static analysis and the controllable environment required for dynamic analysis of actual activation of target require a large amount of relevant prior professional knowledge. In addition, they require a lot of time.

The spread of machine learning actively drives research in malware detection and family classification [11–14]. But malware defense relying on classic machine learning still has its shortcomings. Their essence lies in feature engineering involving a wide range of fields. Once attackers understand the characteristics of the technology used, they can easily avoid detection.

The deep learning algorithm imitates the learning process of the brain by establishing an artificial neural network with a hierarchical structure and realizing artificial intelligence in a computing system. The outstanding advantages of deep learning are the ability to adapt to the rules of a large amount of data, to extract and filter the input information layer by layer, and to have the ability to learn from mistakes. Due to its powerful feature learning capabilities, many studies have applied deep learning to malware identification [15]. Malware family with imbalanced data is optimized by scheme using convolutional neural network (CNN) and Bat algorithm [16]. Screening the advantages of deep learning architecture and visualization is done to achieve robust and intelligent zero-day malware detection in a big data environment using a hybrid approach of two basic analytics and image processing [17]. In addition, deep learning is widely used for countering malware [18, 19].

In this paper, a new malware processing approach is presented based on Multiscale Attention Adaptive Module (MSAAM) and CliqueNet. The CliqueNet optimizes the information flow in a deep network with a cyclic feedback structure, improves the utilization of parameters in the network, and uses a multiscale strategy to prevent the explosive growth of parameters. MSAAM can combine local and global multiscale feature information in the spatial domain and can automatically adjust the arrangement and proportion of channel and spatial attention submodules by auxiliary classifiers according to actual tasks. This method improves the problem of unbalanced samples of malware family classification while reducing feature engineering and is also effective for obfuscation attacks.

1.1. Contributions of This Study.

1.1. Contributions of This Study. Multiscale Attention Adaptive Module (MSAAM) which is a new and general module is designed to optimize the expression of CNNs. It contains two parts which are Improved Efficient Channel Attention (IECA) and Multiscale Spatial Attention (MSA), respectively. The proposed MSAAM can automatically adjust the arrangement and proportion of channel and spatial attention submodules by auxiliary classifiers according to actual tasks. Adopting a multiscale strategy combined with Depthwise Convolution and the original spatial attention block of the Convolutional Block Attention Module (CBAM), this study constructs a new attention mechanism called Multiscale Spatial Attention (MSA) that can combine the key information of the local and global attention.

For the first time, we study malware detection and family classification using CliqueNet to expand the information flow and implement feature filtering that in turn enhances the expression of malware processing research.

Without complex feature engineering, our model has good performance on MalImg and BIG 2015 datasets. It deals effectively with the unbalanced samples of malware family classification, and it can also resist malware obfuscation attacks.

The remaining work: Section 2 explores research connected with the model and the method of gray-scale imaging of malware. Section 3 presents our proposed method and introduces MSAAM. Section 4 tests application of this study. Section 5 generalizes our research. Section 6 discusses the limitations of this paper and proposes plans.

2. Related Work

2.1. Malware Visualization. Malware visualization has been an effective technique in malware research in recent years. Nataraj et al. [20] did not focus on the invisible features of malware detection, but a method to detect malware based on visible components is proposed. Transform every byte in the PE file into a pixel with a value in the range of [0, 255] to convert the malware into a visible gray-scale image. They extracted wavelet decomposition-based GIST features from gray-scale images for malware detection. After the conversion, the gray-scale images of malware are visually different for diverse malware families. And as shown in Figure 1, this diversity also lies in benign and malware. And after converting the malware PE file into a gray-scale image, the slight modification made by the malware author to the binary file in the new variant cannot affect the overall structure of the malware gray-scale image. This visualization technique is very effective in detecting malware and its variants. Nowadays, the visualization of malware images has become a routine method. Venkatraman et al. [21] proposed and studied the application of image-based hybrid methods

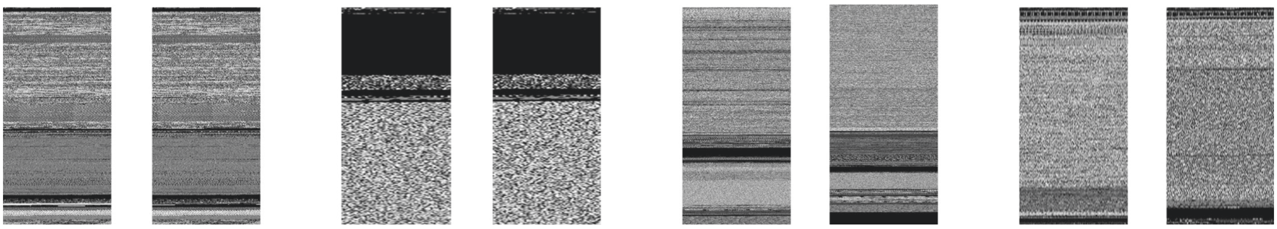


Figure 1: Malware image samples belonging to different families of various malware datasets.

and deep learning architecture to achieve effective malware family classification. Based on image texture similarity, a file-independent deep learning malware family classification method was proposed [22]. Verma et al. [23] demonstrate the idea of using texture analysis to process malware executables. This obfuscated and unbalanced malware discrimination technique combines first-order features extracted on the visualized malware and second-order statistical texture features computed on a gray-scale cooccurrence matrix (GLCM).

However, although these methods solve the problem of code confusion to a certain extent, most of them are based on texture similarity. They require high computational cost feature engineering to extract the complex texture features of malware images (GLCM, Speed-Up Robust Features (SURF), Generalized Search Trees (GIST), Scale-Invariant Feature Transform (SIFT), and Local Binary Pattern (LBP)). When dealing with rapidly iterative malware, they consume a lot of time and are not conducive to processing large datasets. Therefore, reducing the cost of feature engineering and directly using original gray-scale images of malware are the key directions we need to consider. In addition, [24] points out the difficulties faced by texture similarity analysis at this stage. Texture similarity-based analysis is challenging to apply to real-world scenarios with complex malware families, and there are still many practical problems in solving code obfuscation techniques. But [24] considers only the most basic convolutional neural networks in the analysis. We refer to the ideas given in [20] to change target software into gray-scale images. The malware detection and family classification system combined with MSAAM and CliqueNet is used to directly process the original malware gray-scale images to reduce feature engineering and improve efficiency. Through the more robust feature representation capabilities of the new CNNs and the new attention module, the problem that texture similarity analysis is difficult to apply to real-world scenarios with complex malware families is improved.

2.2. Attention Mechanism. Attention mechanisms originating from vision research not only are a focus of recent natural language processing [25–27] but have also been carried forward to research in the image domain [28–30]. It shows strong staying power in the direction of deep learning by imitating the human visual system's attention to important features and eliminating irrelevant information. After SENet [31] demonstrated that attention can strengthen neural network expressive ability by screening significant information areas, the attention mechanism became a conventional means to improve the feature representation of CNNs. The gray-scale images converted from the malware were put into the malware analysis network combining CNN and SENet [32]. By using max and average pooling methods to aggregate features, then creating a network structure combining channel and spatial attention mechanisms, CBAM [33] laid the foundation for the current development of a convolutional neural network attention mechanism. Besides, some other works use adjusted attention blocks to enhance the efficiency of CNNs [34, 35].

After that, the development of the attention module is divided into two aspects: lightweight structure and enhanced feature aggregation. In terms of structural lightweight, ECANet [36] modified SENet based on the idea of lightweight and not reducing the dimensionality, obtaining high efficiency but fewer parameters. In ADCM [37], dropout is integrated into CBAM. We used ECANet and Depthwise Convolution to improve CBAM by the idea of a lightweight and then proposed a new general lightweight convolutional neural network attention module DEAM [38]. Afterward, the DEAM and DenseNet are used to build an effective malware detection and family classification scheme. Some attention modules improved according to the idea of lightweight pay too much attention to the computational efficiency, resulting in the severe lack of feature information used to calculate the attention map, and can not achieve satisfactory results in the vast and complex task model. As a result, their scope of application can only be limited to small task models.

In terms of feature aggregation enhancement, Chen et al., who are dedicated to dynamic scene deblurring research, designed an attention module AAM [39] that can autonomously adjust the position of submodules. Coordattention [40] uses two 1D convolutions with different orientations to aggregate feature information of different dimensions, which enables spatial location information to be embedded into channel attention. The lightweight network using Coordattention can pay attention in a larger region. Multiscale strategies that have performed well in the field of target detection and semantic segmentation are also regarded as an excellent method to improve the feature aggregation effect of the attention mechanism [41–43]. Although the pursuit of feature aggregation will improve the effect of attention mechanisms, sometimes it will significantly increase the computational cost, which is not suitable for small networks. In particular, the self-attention module developed based on the idea of self-attention mechanism, such as transformer [44], realizes the global reference of each pixel-level prediction, which has great requirements for hardware. Users of small task models often do not have the hardware requirements to use self-attention. At present, some researchers have begun to develop local self-attention in order to reduce the hardware requirements of self-attention mechanism.

Due to the good results shown by DEAM [38], here we use a multiscale strategy to create an effective spatial attention mechanism based on the overall framework of DEAM. The auxiliary classifier is used to automatically adjust the arrangement and proportion of the channel and spatial attention submodules. Multiscale Attention Adaptive Module (MSAAM) which is a new and general module is designed.

2.3. Development of the CNNs. The emergence of convolutional neural networks (CNNs) has brought rapid development to extensive computer vision fields. For strengthening the performance of CNNs, the exploration of network architecture has always been part of the research of CNNs. The original convolutional neural network LeNet [45] consists of five layers. VGGNet [46] has 19 layers and proves that its final performance can be influenced by increasing its depth. GoogLeNet [47] has 22 layers and proves the width is another crucial factor in determining model representation. The three aspects of depth, width, and cardinality have gradually become the most critical factors in determining network architecture. The scale of CNNS has also been increasing with the deepening of exploration. However, the blindly enlarged deep network will make it difficult for the latter layer to obtain gradient information from the previous layer, resulting in the problem of gradient disappearance and parameter redundancy [48]. The emergence of this problem restricts the development of the network architecture in terms of depth, width, and cardinality.

Some experts have explored new paths from the constantly enlarged differences in network architecture, and they have moved towards the exploration of connection modes. ResNet [49] introduces a bypass path so that the top-level network can obtain information from the bottom-level network, strengthens the correlation of the gradients between the network layers, and alleviates the problem of gradient disappearance. It also simplifies network training. After ResNet has shown excellent performance in computer vision orientation, the bypass path is considered to be a key factor to facilitate the work of training these deep networks and solving the problem of gradient disappearance. DenseNet [50] further deepens the idea of ResNet, applying the bypass path to entirety, realizing complex connection, and exploiting the potential of the network through feature reuse. A class-balanced loss is added to the last layer of the DenseNet model for classification for sample imbalanced malware, and this loss is reweighted [51]. But as the dense connection path in DenseNet increases linearly, its parameters will increase sharply. Compared with DenseNet, CliqueNet [52] uses a cyclic feedback structure to further optimize feature flow in deep networks and improve the utilization of parameters in the network. And it also introduces a multiscale feature strategy on the model output to avoid the problem of a sharp increase in parameters in DenseNet.

But as the model continues to expand in the network architecture and connection mode, the hardware burden it brings to researchers also increases dramatically. If researchers want to make a better choice between model performance and requirements, building a general enhancement module in a deep learning model has more room for development than accumulating more nonlinear layers. Therefore, in this paper, new malware processing model is constructed by combining MSAAM and CliqueNet.

3. Proposed Methodology

Our proposed model is composed of CliqueNet and Multiscale Attention Adaptive Module (MSAAM). We obtain CliqueNet suitable for the proposed model based on CliqueNet-S0. Multiscale Spatial Attention (MSA) and Improved Effective Channel Attention (IECA) are two important submodules that make up MSAAM. First, the framework of our proposed malware processing method is described. After that, the architecture of CliqueNet is introduced. Finally, we show our proposed MSAAM and describe MSA.

3.1. Method Overview. Figure 2 depicts the whole framework of our malware detection and family classification approach. The proposed malware processing model uses MSAAM and CliqueNet to automatically extract features at various levels of abstraction from gray-scale images of malware. These features can express the image comprehensively and clearly, and the extracted features are used to train the model. This model can directly process the original malware gray-scale images to reduce feature engineering and improve efficiency.

The following describes the process that the incoming malware samples go through to process them. First, the input malware executable file samples are turned into gray-scale images using the same method as Nataraj et al. [20]. The converted gray-scale images are sent to the trained malware detection model combining MSAAM and CliqueNet to effectively identify benign software and malware. After distinguishing benign software and malware, the malicious samples are sent to the trained malware family classification model combining MSAAM and CliqueNet to effectively identify the family to which each malware belongs.

3.2. Structure of the CliqueNet. To maximize the information flow between layers and solve the problem of a sharp increase in parameters in DenseNet, CliqueNet [52] was created. The most prominent feature of CliqueNet is the use of a cyclic feedback structure with spatial attention effects, so that the model not only has a forward propagation part, but

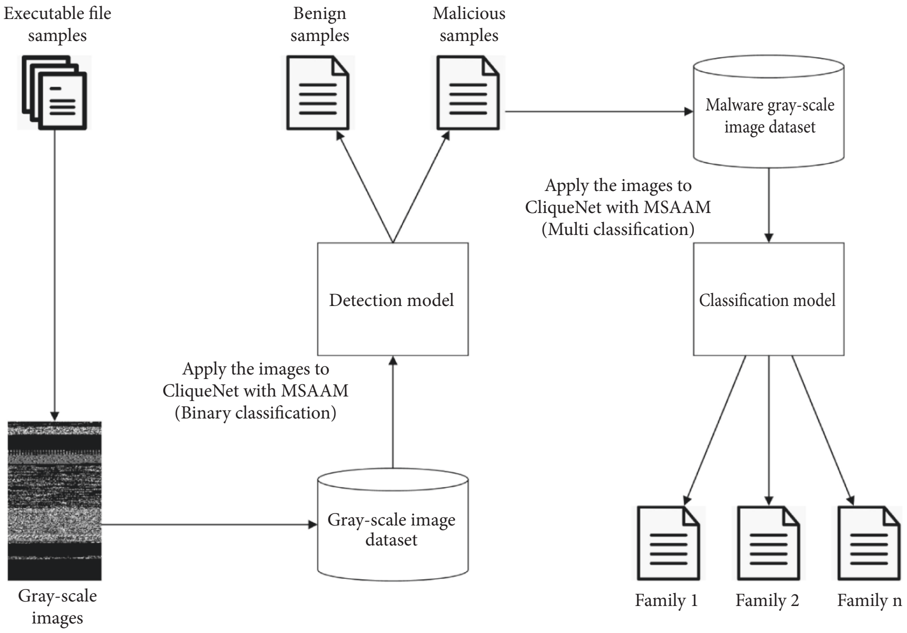

Executable file samples
Benign samples
Malicious samples
Malware gray-scale image dataset
Apply the images to CliqueNet with MSAAM (Multi classification)
Detection model
Classification model
Apply the images to CliqueNet with MSAAM (Binary classification)
Gray-scale image dataset
Gray-scale images
Family 1
Family 2
Family n

Figure 2: Malware processing method flowchart.

also optimizes the feature map of the previous level based on the output of the next level. Therefore, we can utilize the feature map output in the convolution repeatedly. The modified feature map will consider more important information. Moreover, for avoiding the increased hardware requirements and redundant parameters caused by the explosive growth of parameters in DenseNet, CliqueNet introduces a multiscale feature strategy on the model output. Figure 3 depicts the basic functional module of CliqueNet.

CliqueBlock is the part of CliqueNet that implements the loop feedback structure, as shown in Figure 3. It can be divided into stage-I updated forward and stage-II updated backward. Stage-I is like DenseNet's forward densely connected propagation; the input of each layer will contain the refined feature information of all previous layers. Stage-II realizes the reverse refinement of the model. The input of each convolution operation not only includes the final results of all previously updated layers, but also includes the output feature maps of the subsequent levels. In each step of the stage-II update, the last few feature maps are used to refine relatively earliest feature information, because final feature maps contain relatively higher-level visual information. In this way, the cyclic feedback structure realizes the refinement of the feature maps of each level to achieve the effect of spatial attention. For i-th layer and k-th stage in stage-II, the alternately updated expression is

$$X_i^{(k)} = g\left(\sum_{l<i} W_{li} * X_l^{(k)} + \sum_{m>i} W_{mi} * X_m^{(k-1)}\right), \tag{1}$$

where $k \geq 2$, k denotes the number of stages, and two stages complete a loop. $W * X$ means that the convolution kernel performs convolution operation on the input feature map, and g is nonlinear activation function.

To solve the problem of the sharp increase of parameters in DenseNet, CliqueNet adopts a multiscale feature strategy in the overall structure, as shown in Figure 3. The output of each CliqueBlock consists of two parts. One part is the combination of the output of each layer after reverse refining is called transit_feature, and the other is the combination of input layer and output of each layer after reverse refining is called block_feature. The transit_feature is transmitted to the next CliqueBlock through the transition with the attention mechanism. The block_feature is compressed into a feature vector after global average pooling. Since only the output of stage-II of each CliqueBlock will be used as the input of the next CliqueBlock, the dimension of the feature map of CliqueBlock will not increase super linearly, which has the advantages of parameter amount and calculation amount. And there is no need to use a bottleneck structure like DenseNet to prevent parameter explosion. So the basic structure of CliqueBlock in CliqueNet is BN + ReLU +3 ∗ 3 Conv + Dropout.

We obtain CliqueNet in our method by adapting CliqueNet-S0. Table 1 describes the details of our CliqueNet and CliqueNet-S0 parameters.

3.3. Multiscale Attention Adaptive Module. Here, our purpose is to apply attention block to enhance the effect of malware gray-scale images detection and family classification. For this reason, MSAAM aims to further strengthen the feature extraction capabilities of CliqueNet. The novelty of MSAAM is that it can combine local and global multiscale key information in the spatial domain and can automatically adjust the arrangement and proportion of channel and spatial attention submodules by auxiliary classifiers according to actual tasks to obtain better feature expression ability.

The proposed MSAAM inherits the overall design of CMBA [33], and its construction can be divided into two parts: Improved Efficient Channel Attention (IECA) which

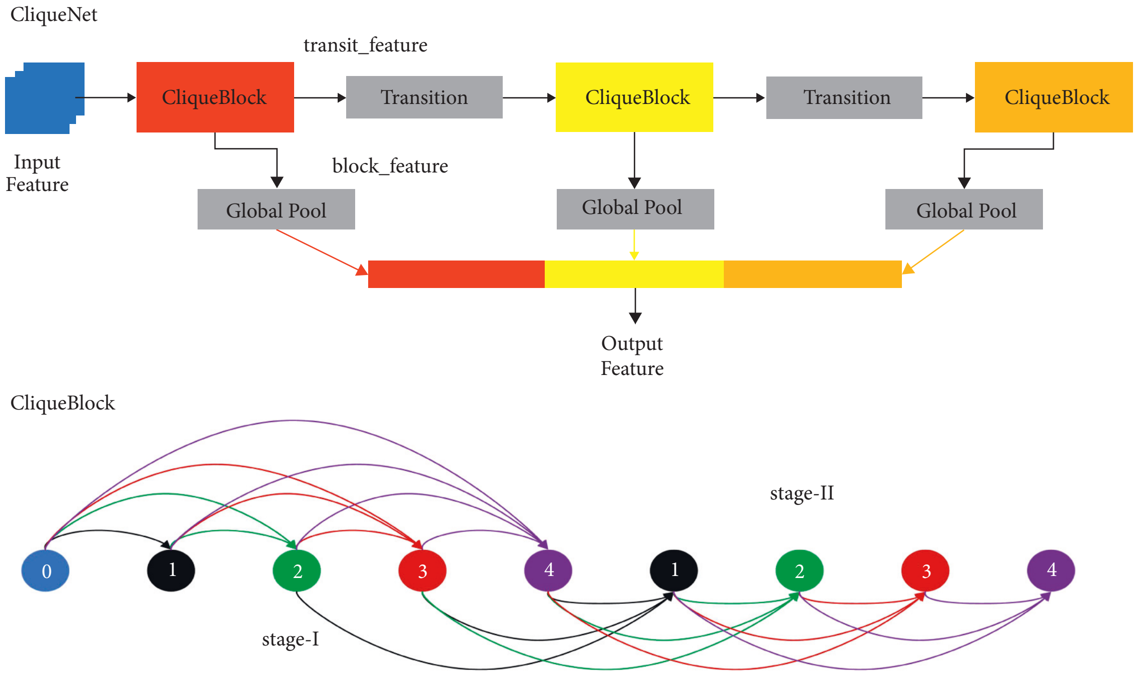


Figure 3: Basic functional modules of CliqueNet [52] and CliqueBlock.

Table 1: The details of our CliqueNet and CliqueNet-S0 parameters. The two numbers in CliqueBlock represent the number of convolutional filters in the block and the number of convolutional layers in the block.

Layers	Our CliqueNet	CliqueNet-S0
Convolution	7 * 7 conv, 32, stride 1	7 * 7 conv, 64, stride 2
Pooling	2 * 2 max pool, stride 2	3 * 3 max pool, stride 2
CliqueBlock (1)	36 * 4	
Transition	1 * 1 conv 2 * 2 ave pool, stride 2	36 * 5
CliqueBlock (2)	36 * 4	
Transition	1 * 1 conv 2 * 2 ave pool, stride 2	64 * 6
CliqueBlock (3)	36 * 4	
Transition	1 * 1 conv 2 * 2 ave pool, stride 2	100 * 6
CliqueBlock (4)	36 * 4	80 * 6

is from [38] and Multiscale Spatial Attention (MSA). Figure 4 shows the overall framework of MSAAM. We use an adaptive method in the comprehensive framework to automatically adjust the arrangement and proportion of the channel and spatial attention submodules. We define the input feature map for MSAAM as $M \in R^{C\times H\times W}$. A one-dimensional channel attention map $M_C \in R^{C\times 1\times 1}$ is obtained by IECA in MSAAM, which can highlight essential feature information in the channel dimension of the feature map. The three-dimensional local space attention map $M_{S1} \in R^{C\times H\times W}$ and the two-dimensional global space attention map $M_{S2} \in R^{1\times H\times W}$ of the feature map are calculated by MSA. The three-dimensional space attention map $M_S \in R^{C\times H\times W}$ is obtained by combining the local and global multiscale key information to highlight more meaningful spatial feature information. Different channel and spatial attention submodule placements in different actual scenes will make the attention mechanism exert different levels of effects.

In this study, we apply learnable matrices to allow the module to adaptively select the placement of attention submodules suitable for the current scene, as well as the proportion of channel and spatial attention submodules that affect the results. The learnable matrices $W_1, W_2 \in R^{C\times H\times W}$ are auxiliary classifiers used to implement the adaptive permutation selection method. We set $W_1 = 0$ means all elements in W_1 are 0, $W_1 = 1$ means all elements in W_1 are 1, and W_2 also has the same setting. The serial placement of channel and space attention submodules is the special case of $W_2 = 0$, and the parallel placement of channel and space attention submodules is the special case of $W_1 = 0$. The calculation process of MSAAM:

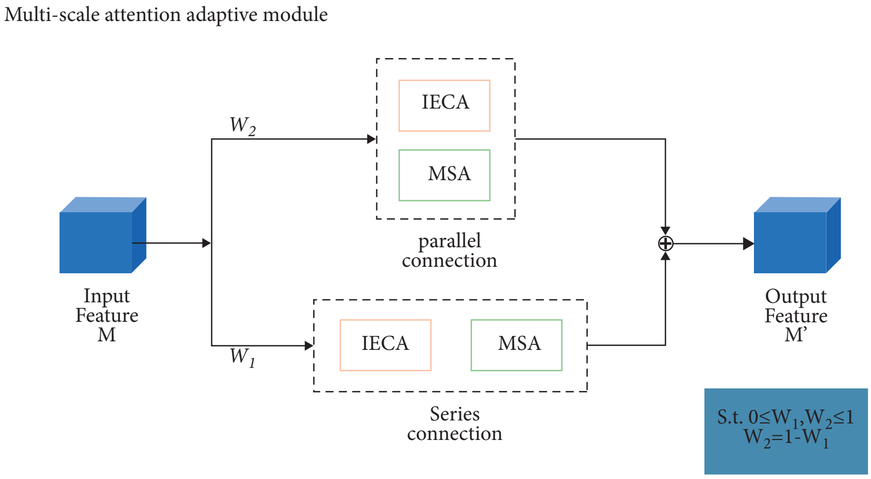


Figure 4: The overall architecture of MSAAM.

$$M' = M_S(M_C(M) \otimes M) \otimes M_C(M) \otimes M \otimes W_1 + M_C(M) \otimes M \otimes M_S(M) \otimes W_2, \tag{2}$$

where $\otimes$ denotes elementwise multiplication, + denotes element summation, and M' denotes the output feature map.

3.3.1. Multiscale Spatial Attention. Differing from the channel attention, the spatial one emphasizes the spatial features. The spatial one computes the likelihood of spatial feature information on the feature map to highlight more meaningful spatial feature information. This study uses a multiscale strategy combined with the spatial attention mechanism of CBAM [33] and Depthwise Convolution to construct a new spatial attention mechanism MSA that can combine the critical information on local attention and global attention, as shown in Figure 5. By collecting feature information on multiple scales, multiscale feature extraction and feature fusion can significantly enhance the information aggregation ability and expand the receptive field of the model. Since spatial information is more fragmented than channel information, more feature information is lost when aggregating. And these two strategies need to consider the application cost when using, so using them on spatial attention can lead to better optimization.

Depthwise Convolution uses convolution and Sigmoid functions on each channel to obtain a spatial attention map without dimensionality reduction. Compared with the spatial attention mechanism in CMBA that uses the max and average pools to compress information, it can obtain better spatial information. However, since Depthwise Convolution is a separate operation for each channel, it focuses on the local spatial information relationships within different regions of the feature map. After visualizing the gray-scale image, the operation of Depthwise Convolution on a single channel of the feature map will divide the image into regions, and the obtained relationship is only the local information relationship in each region. Just using Depthwise Convolution lacks a vision of the global information relationship of the feature map. Figure 6 depicts the visualization of the local and global information relationship of the feature map. Therefore, considering the multiscale strategy, fusion of the key information of local attention extracted by Depthwise Convolution and the key information of global attention extracted by the spatial attention mechanism in CMBA can better pay attention to the spatial information of the features.

In MSA, Depthwise Convolution is applied to the input feature map $M \in R^{C\times H\times W}$, and the Sigmoid function is used for the output feature descriptor $F_2 \in R^{C\times H\times W}$ to obtain a three-dimensional local spatial attention map $M_{S1} \in R^{C\times H\times W}$. The max pool and average pool compression are used for the input feature map $M \in R^{C\times H\times W}$. These two spatial feature descriptors ($F^S_{\text{avg}}$and$F^S_{\max}$) indicate the average pool space information and the max pool space information, respectively. A 7 $*$ 7 Conv and the Sigmoid function are used for the feature descriptor $F_3 \in R^{2\times H\times W}$ obtained after the two spatial context descriptors are spliced to bring a 2D global spatial attention map $M_{S2} \in R^{1\times H\times W}$. Finally, this attention adds the three-dimensional local space attention map $M_{S1} \in R^{C\times H\times W}$ and the two-dimensional global space attention map $M_{S2} \in R^{1\times H\times W}$ element by element to get the three-dimensional space attention map $M_S \in R^{C\times H\times W}$. $W_3, W_4 \in R^{C\times H\times W}$ are learnable matrices used to determine the proportion of local key information and global key information. The calculation formula of MSA is as follows:

$$\begin{aligned} M_S(M) &= \sigma(\text{DepthwiseConv2D}(M)) \otimes W_3 \\ &\quad + \sigma\left(\text{conv}^{7\times 7}\left(\left[F^S_{\text{avg}}; F^S_{\max}\right]\right)\right) \otimes W_4 \\ M_S(M) &= M_{S1} \otimes W_3 + M_{S2} \otimes W_4, \end{aligned} \tag{3}$$

where DepthwiseConv2D denotes Depthwise Convolution, and [;] denotes tensor splicing.

4. Experiments and Analysis

4.1. Experimental Setting and Datasets. We use MalImg [20] and Microsoft's BIG 2015 to assess the proposed approach. Tables 2 and 3 give a detailed introduction of the two benchmark datasets. The MalImg dataset, consisting of 25

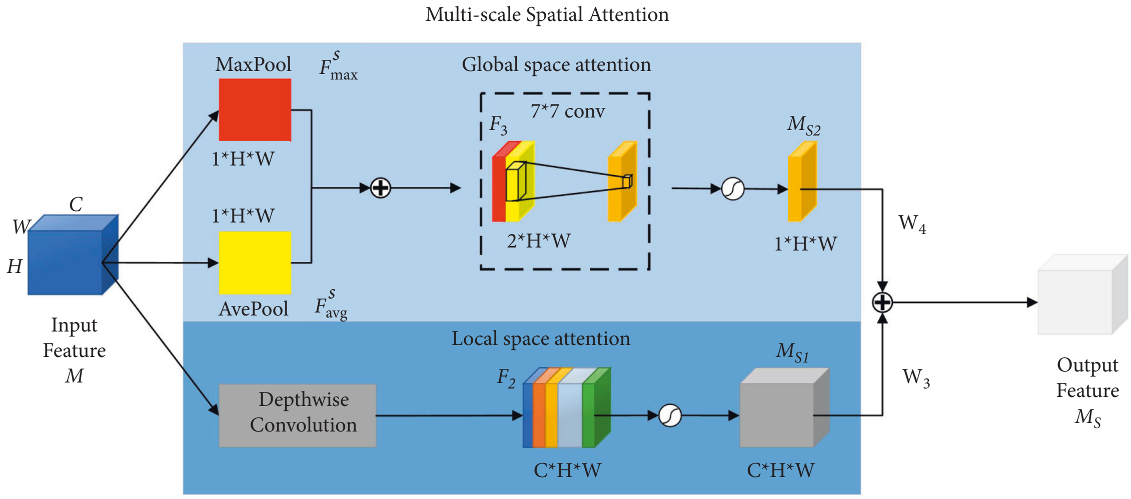
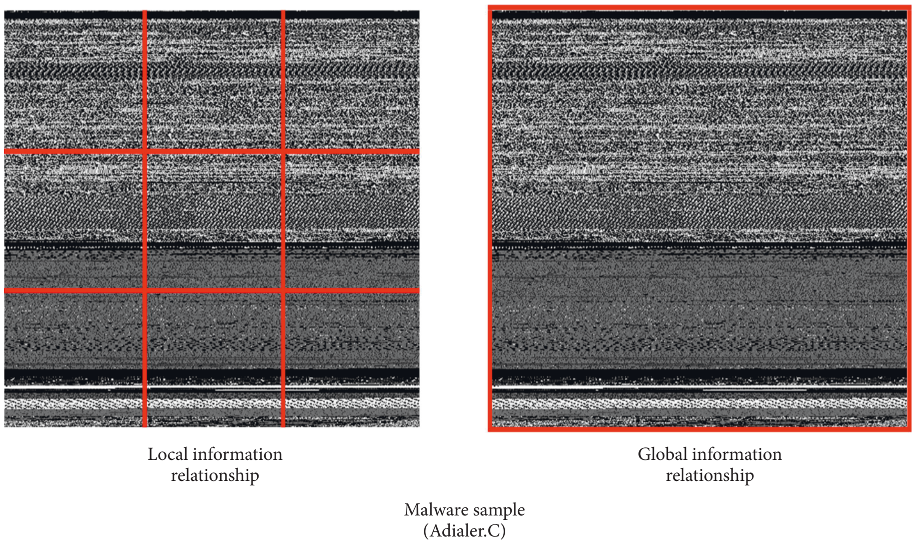


FIGURE 5: The detailed process of Multiscale Spatial Attention.


Local information relationship

Global information relationship

Malware sample (Adialer.C)

FIGURE 6: Convolutional receptive field gray-scale images visualization.

TABLE 2: Sample distribution of MalImg dataset.

No.	Family	Number of samples	No.	Family	Number of samples
1	Adialer.C	122	14	Lolyda.AA2	184
2	Agent.FYI	116	15	Lolyda.AA3	123
3	Allaple.A	2949	16	Lolyda.AT	159
4	Allaple.L	1591	17	Malex.gen!J	136
5	Alueron.gen!J	198	18	Obfuscator.AD	142
6	Autorun.K	106	19	Rbot!gen	158
7	C2LOP.gen!g	200	20	Skintrim.N	80
8	C2LOP.P	146	21	Swizzor.gen!E	128
9	Dialplatform.B	177	22	Swizzor.gen!I	132
10	Dontovo.A	162	23	VB.AT	408
11	Fakerean	381	24	Wintrim.BX	97
12	Instantaccess	431	25	Yuner.A	800
13	Lolyda.AA1	213			9339

Table 3: Sample distribution of BIG 2015 dataset.

No.	Family	Number of samples	No.	Family	Number of samples
1	Ramnit	1541	6	Tracur	751
2	Lollipop	2478	7	Kelihos_ver1	398
3	Kelihos_ver3	2942	8	Obfuscator.ACY	1228
4	Vundo	475	9	Gatak	1013
5	Simda	42			10868

malware families and 9339 malware samples in total, is a massive and unbalanced malware dataset. The BIG 2015 dataset contains 21741 malware samples, consisting of 9 malware families. The training set contains 10868 samples and the test set has 10873 ones. The test set of the BIG 2015 dataset, however, did not give the corresponding label. Thus this paper only uses the training set part. The bytes files containing the original hexadecimal code of the files in the dataset are used to generate gray-scale images of the malware.

To classify malware families, we directly use the MalImg and BIG 2015 as the evaluation benchmark. For the malware detection part, we randomly selected a total of 1087 malware samples out of the 34 malware families contained in the MalImg and BIG 2015 datasets. Using the filtered malicious samples and an equal number of benign samples to create a new malware detection dataset, it contains rich malware families to ensure the experimental scalability.

To effectively evaluate the performance of our model, the dataset splits into three parts where the data is used for training, validation, and test at a ratio of 6 : 2 : 2. Moreover, to abate errors, we repeat every experiment 5 times. Our experiment uses the Adam optimizer and categorical_crossentropy, the batch size is 16, and the learning rate is 0.0005. The environment equipment is as follows: Windows 10, Intel(R) Core (TM) i7-1165G7 CPU @ 2.80 GHz 2.80 GHz and NVIDIA GeForce GTX 1060. The proposed malware processing method is built on the Python framework and the tensorflow2.3 framework.

Malware loses different degrees of information when it converts to gray-scale images of different sizes. However, an immense gray-scale image size will have high requirements on the physical equipment and will bring a huge burden to the training of the model. In [24], the effects of image sizes on malware family classification based on texture feature analysis are experimentally verified. To balance performance and cost, and to guarantee the validity of the experiment, we adjust the gray-scale image size of the model input to $256 * 256$, which is recommended in [24].

The $N * N$ confusion matrix is used to calculate four types of indicators in order to examine our proposed model for detecting malware. These indicators are as follows: *accuracy*, *precision*, *recall*, and *F1 score*. The accuracy rate used alone can only show the expressive ability at the macro level and cannot sensitively reflect the prediction level of the class with a small number of samples. *Precision* reflects how many of all the samples marked as positive are expressed correctly. Recall reflects the ability of the model to identify the target. *F1 score* determines the accuracy and robustness of the model. For the two-class detection task, we directly obtain these values. For the multiclass family classification task, we will enumerate three metrics for each family in detail. Finally, the indicators of each family are combined to calculate macro-precision, macro-recall, and macro-F1.

Indicators other than accuracy are calculated as follows:

$$\begin{aligned}\text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}},\\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}},\\ \text{F1} &= \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}},\end{aligned} \tag{4}$$

where TP is the target sample expressed as positive, FP is the target sample expressed as negative, and FN is the nontarget sample expressed as negative.

4.2. Performance of Detecting Malware.

4.2. Performance of Detecting Malware. Table 4 shows the results in a $2 * 2$ confusion matrix. Our accuracy, precision, recall, and F1 score on malware dataset are 99.8%, 99.8%, 99.8%, and 99.8%, respectively. For the test set with a total number of samples of 421, only one benign sample was classified incorrectly. After validation on the experimental malware detection dataset, our model can effectively distinguish between benign software and malicious software. Table 5 demonstrates that our model outperforms existing methods in detection ability.

4.3. Performance of Classifying Family

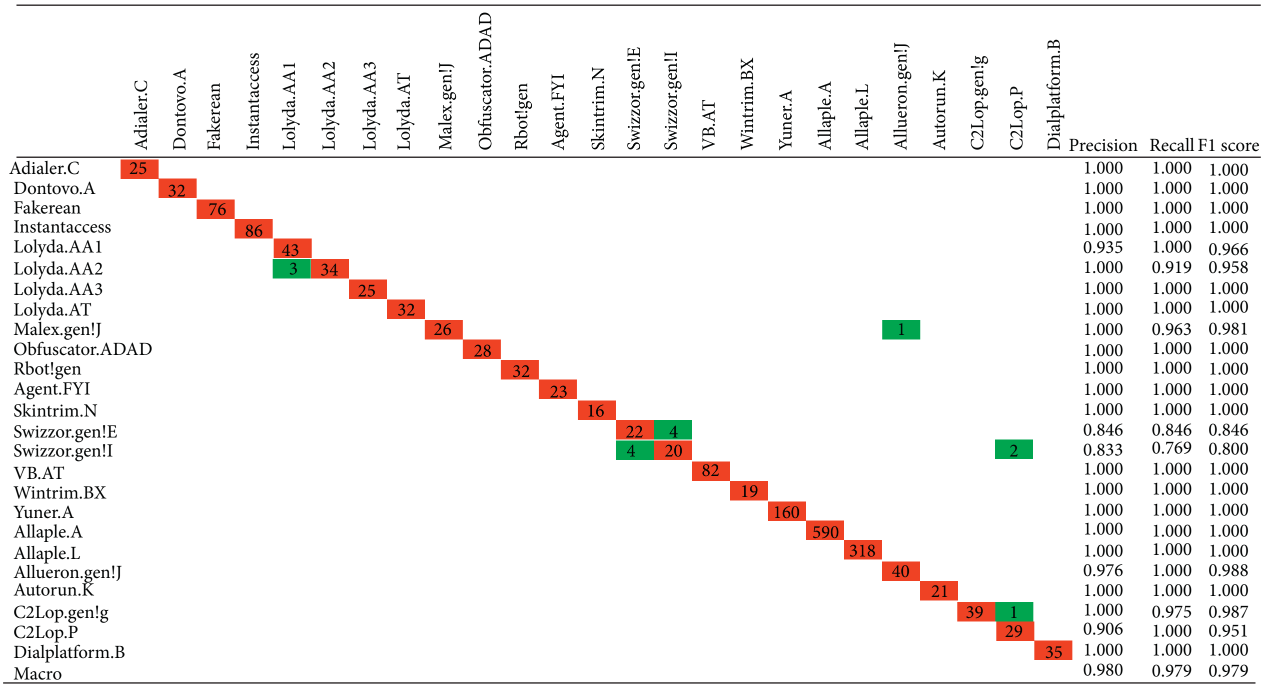
4.3.1. MalImg Dataset. The $25 * 25$ confusion matrix on Figures 7 and 8 illustrates our classifying results of the proposed method and CliqueNet on MalImg dataset. Our accuracy, precision, recall, and F1 score on the MalImg are 99.2%, 98.0%, 97.9%, and 97.9%, respectively. Without MSAAM on MalImg dataset, the sequential metrics are 98.6%, 96.7%, 96.3%, and 96.5%, respectively. Table 6 illustrates the comparison results on the MalImg dataset. After validation on the experimental MalImg dataset, our model is equivalent to [23, 51] in precision, recall, and F1 score, but it surpasses these two tasks in terms of accuracy. Compared with other work, our model has achieved a comprehensive surpass in all four evaluation indicators. Table 7 shows the influence of our model on families with smaller samples in two datasets. It achieved F1 scores of 100%, 100%, and 87.5% on Skintrim.N, Wintrim.BX, and Simda families, respectively. These show that our model can effectively complete the classification of malware families

Table 4: Our classification performance on malware dataset.

	Malware	Benign	Precision	Recall	F1 score
Malware	204	0	0.995	1	0.998
Benign	1	216	1	0.995	0.998
Macro			0.998	0.998	0.998

Table 5: The comparison of the binary classification effect between our model and others.

Models	Accuracy (%)
RF [5]	97.0
García and DeCastro-García [10]	99.4
TELM [12]	99.7
SVM [13]	98.5
Zhang et al. [18]	95.1
CRNN [19]	96.2
DenseNet + DEAM [38]	99.3
Hemalatha et al. [51]	97.6
Proposed method	99.8

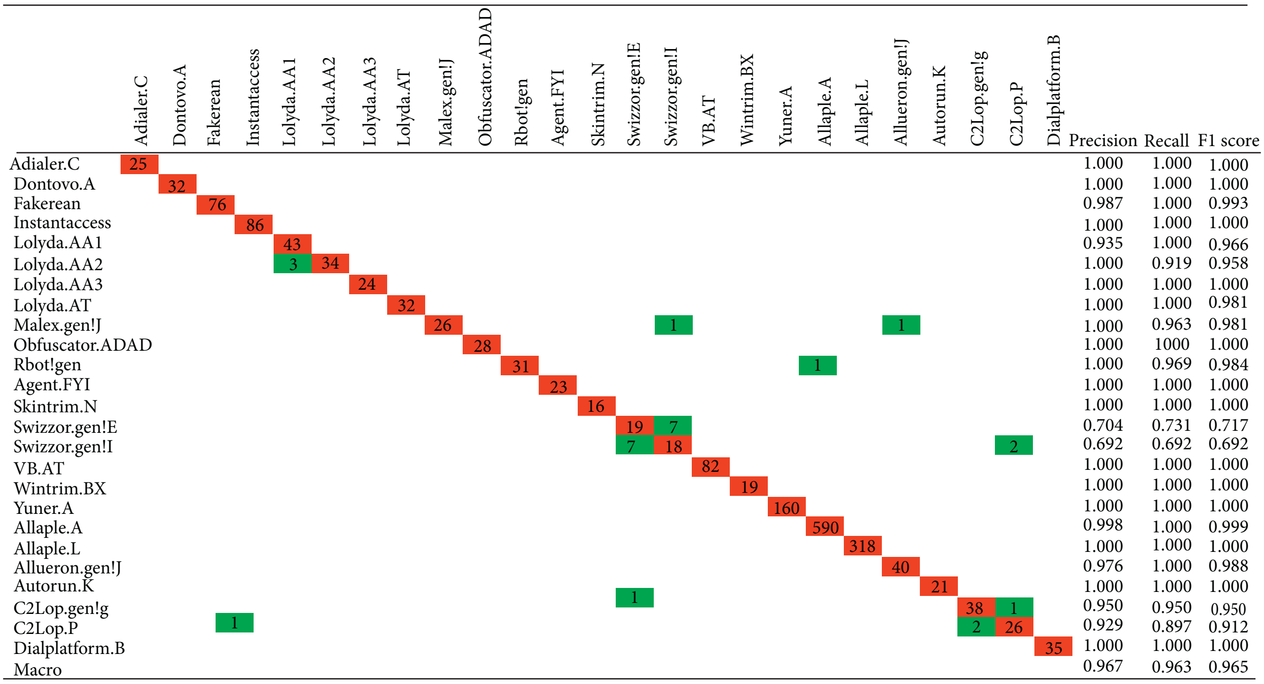


	Adialer.C	Dontovo.A	Fakerean	Instantaccess	Lolyda.AA1	Lolyda.AA2	Lolyda.AA3	Lolyda.AT	Malex.gen!J	Obfuscator.ADAD	Rbot!gen	Agent.FYI	Skintrim.N	Swizzor.gen!E	Swizzor.gen!I	VB.AT	Wintrim.BX	Yuner.A	Allaple.A	Allaple.L	Allueron.gen!J	Autorun.K	C2Lop.gen!g	C2Lop.P	Dialplatform.B	Precision	Recall	F1 score
Adialer.C	25																									1.000	1.000	1.000
Dontovo.A		32																								1.000	1.000	1.000
Fakerean			76																							1.000	1.000	1.000
Instantaccess				86																						1.000	1.000	1.000
Lolyda.AA1					43																					0.935	1.000	0.966
Lolyda.AA2					3	34																				1.000	0.919	0.958
Lolyda.AA3							25																			1.000	1.000	1.000
Lolyda.AT								32																		1.000	1.000	1.000
Malex.gen!J									26												1					1.000	0.963	0.981
Obfuscator.ADAD										28																1.000	1.000	1.000
Rbot!gen											32															1.000	1.000	1.000
Agent.FYI												23														1.000	1.000	1.000
Skintrim.N													16													1.000	1.000	1.000
Swizzor.gen!E														22	4											0.846	0.846	0.846
Swizzor.gen!I														4	20									2		0.833	0.769	0.800
VB.AT																82										1.000	1.000	1.000
Wintrim.BX																	19									1.000	1.000	1.000
Yuner.A																		160								1.000	1.000	1.000
Allaple.A																			590							1.000	1.000	1.000
Allaple.L																				318						1.000	1.000	1.000
Allueron.gen!J																					40					0.976	1.000	0.988
Autorun.K																						21				1.000	1.000	1.000
C2Lop.gen!g																							39	1		1.000	0.975	0.987
C2Lop.P																								29		0.906	1.000	0.951
Dialplatform.B																									35	1.000	1.000	1.000
Macro																										0.980	0.979	0.979

Figure 7: Multiclassification performance of proposed method on MalImg.

and is robust to the problem of imbalance in the classification of malware families. Compared with CliqueNet without MSAAM, it proves that the proposed MSAAM can strengthen the attention to the characteristics of malware. Compared with our previous work [38] and CliqueNet + DEAM, it proves that MSAAM improves the performance of the attention module based on the DEAM.

The comparison between Figures 7 and 8 illustrates that the classification difficulties of the MalImg dataset are concentrated on two families. They are Swizzor.gen!E and Swizzor.gen!I, respectively. The addition of MSAAM has greatly improved the classification effect on these two key families, while the classification effect on other families has also been slightly improved. MSAAM raises the F1 score of Swizzor.gen!E from 71.7% to 84.6% and raises the F1 score of Swizzor.gen!I from 69.2% to 80.0%. This reflects that our proposed MSAAM effectively improves the feature expression ability of CNN. And for the entire network architecture of deep learning, the addition of MSAAM will hardly bring about an increase in computational consumption. There are many samples processed by obfuscation techniques in the MalImg data set. Our model achieves 100% classification on 17 out of 25 families. The families using the packaging technology UPX which makes them



	Adialer.C	Dontovo.A	Fakerean	Instantaccess	Lolyda.AA1	Lolyda.AA2	Lolyda.AA3	Lolyda.AT	Malex.gen!J	Obfuscator.ADAD	Rbot!gen	Agent.FYI	Skintrim.N	Swizzor.gen!E	Swizzor.gen!I	VB.AT	Wintrim.BX	Yuner.A	Allaple.A	Allaple.L	Alluerón.gen!J	Autorun.K	C2Lop.gen!g	C2Lop.P	Dialplatform.B	Precision	Recall	F1 score
Adialer.C	25																									1.000	1.000	1.000
Dontovo.A		32																								1.000	1.000	1.000
Fakerean			76																							0.987	1.000	0.993
Instantaccess				86																						1.000	1.000	1.000
Lolyda.AA1					43																					0.935	1.000	0.966
Lolyda.AA2					3	34																				1.000	0.919	0.958
Lolyda.AA3							24																			1.000	1.000	1.000
Lolyda.AT								32																		1.000	1.000	0.981
Malex.gen!J									26						1						1					1.000	0.963	0.981
Obfuscator.ADAD										28																1.000	1000	1.000
Rbot!gen											31								1							1.000	0.969	0.984
Agent.FYI												23														1.000	1.000	1.000
Skintrim.N													16													1.000	1.000	1.000
Swizzor.gen!E														19	7											0.704	0.731	0.717
Swizzor.gen!I														7	18									2		0.692	0.692	0.692
VB.AT																82										1.000	1.000	1.000
Wintrim.BX																	19									1.000	1.000	1.000
Yuner.A																		160								1.000	1.000	1.000
Allaple.A																			590							0.998	1.000	0.999
Allaple.L																				318						1.000	1.000	1.000
Allueron.gen!J																					40					0.976	1.000	0.988
Autorun.K																						21				1.000	1.000	1.000
C2Lop.gen!g														1									38	1		0.950	0.950	0.950
C2Lop.P			1																				2	26		0.929	0.897	0.912
Dialplatform.B																									35	1.000	1.000	1.000
Macro																										0.967	0.963	0.965

Figure 8: Multiclassification performance of CliqueNet on the MalImg dataset.

Table 6: Comparative analysis of proposed method with others on MalImg.

Models	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
Cui et al. [15]	94.5	94.6	94.5	94.5
Vinayakumar et al. [17]	96.3	96.3	96.2	96.2
Venkatraman et al. [21]	96.3	91.8	91.5	91.6
Gibert et al. [22]	98.5	95.8	96.6	95.8
Verma et al. [23]	98.5	98.0	98.0	98.0
Densenet + DEAM [38]	98.5	96.9	96.6	96.7
Hemalatha et al. [51]	98.2	97.8	97.9	97.9
CliqueNet	98.6	96.7	96.3	96.5
CliqueNet + DEAM	98.8	97.1	96.8	97.0
Proposed method	99.2	98.0	97.9	97.9

Table 7: Influence of our model on families with smaller samples in two datasets.

	Family	Number of samples	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
MalImg	Skintrim.N	80	100	100	100	100
	Wintrim.BX	97	100	100	100	100
Big 2015	Simda	42	87.5	87.5	87.5	87.5

indistinguishable from each other in similar structures in this dataset are Yuner.A, Rbot!gen, Malex.gen!J, VB.AT, and Autorun.K. But our method reaches a 100% classification accuracy in Yuner.A, VB.AT, Autorun.K, and Rbot!gens, and the F1 score on Malex.gen!J is 98.1%. Allaple uses random keys in the code part to encrypt in several layers, but our model makes a perfect distinction between Allaple.A and Allaple.L. Lolyda.AA1 and Lolyda.AA3 belong to the same family variants and are also made a perfect distinction. All these prove that our model is effective for the classification of obfuscated malware.

4.3.2. BIG 2015 Dataset. The $9 * 9$ confusion matrix on Figures 9 and 10 illustrates the classification results of the proposed method and CliqueNet on BIG 2015 dataset. The accuracy, precision, recall, and F1 score of our model on BIG 2015 dataset are 98.2%, 96.6%, 96.3%, and 96.4%, respectively. The accuracy, precision, recall and F1 score of CliqueNet without MSAAM on BIG 2015 dataset are 97.6%, 96.8%, 94.6%, and 95.5%, respectively. After the addition of MSAAM, the evaluation indicators other than precision have been improved. The comparison between Figures 9 and 10 illustrates that MSAAM improves the classification

	Ramnit	Lollipop	Kelihos_ver3	Vundo	Simda	Tracur	kelihos_ver1	Obfuscator:ACY	Gatak	Precision	Recall	F1 score
Ramnit	303	1	1	0	0	0	1	3	0	0.968	0.981	0.974
Lollipop	0	495	0	0	0	0	0	1	0	0.990	0.998	0.994
Kelihos_ver3	0	0	588	0	0	0	0	0	0	0.995	1000	0.997
Vundo	0	1	0	91	1	1	0	0	1	0.968	0.968	0.963
Simda	1	0	0	0	7	0	0	0	0	0.875	0.875	0.875
Tracur	2	1	0	1	0	145	1	0	0	0.967	0.967	0.967
kelihos_ver1	0	0	1	0	0	0	77	1	0	0.975	0.975	0.975
Obfuscator:ACY	2	2	1	2	0	4	0	226	3	0.974	0.922	0.948
Gatak	0	0	0	0	0	0	0	1	201	0.980	0.995	0.988
Macro										0.966	0.963	0.964

Figure 9: Multiclassification performance of proposed method on BIG 2015.

	Ramnit	Lollipop	Kelihos_ver3	Vundo	Simda	Tracur	kelihos_ver1	Obfuscator:ACY	Gatak	Precision	Recall	F1 score
Ramnit	299	1	0	0	0	6	1	2	0	0.955	0.968	0.961
Lollipop	2	492	0	0	0	0	0	0	2	0.984	0.992	0.988
Kelihos_ver3	0	0	588	0	0	0	0	0	0	0.998	1000	0.999
Vundo	0	1	0	91	1	1	0	0	1	0.919	0.958	0.938
Simda	0	0	0	1	6	0	0	1	0	1000	0.750	0.857
Tracur	2	1	0	1	0	144	1	0	1	0.941	0.960	0.950
kelihos_ver1	0	0	0	0	0	0	79	0	0	0.952	1000	0.975
Obfuscator:ACY	10	2	1	5	0	2	1	223	1	0.987	0.910	0.947
Gatak	0	3	0	1	0	0	0	0	198	0.975	0.980	0.978
Macro										0.968	0.946	0.955

Figure 10: Multiclassification performance of CliqueNet on BIG 2015.

Table 8: Comparative analysis of proposed method with others on BIG 2015.

Models	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
ACNN [16]	96.0	95.4	88.3	89.7
Gibert et al. [22]	97.5			94.0
DenseNet + DEAM [38]	97.3	95.3	95.4	95.4
CliqueNet	97.6	96.8	94.6	95.5
Proposed method	98.2	96.6	96.3	96.4

performance on multiple families. Table 8 shows a comparative analysis of the proposed method and others on the BIG 2015. Our approach outperforms other works in malware family classification.

5. Conclusion

An efficient malware processing method is presented by using the newly designed universal and effective Multiscale Attention Adaptive Module (MSAAM) and CliqueNet. MSAAM can combine local and global multiscale feature information in the spatial domain and can automatically adjust the arrangement and proportion of channel and spatial attention submodules by auxiliary classifiers according to actual tasks. This method can directly process the gray-scale images of malware, reducing the feature engineering and improving the problem of unbalanced samples of malware family classification. It is also reliable and effective for obfuscation attacks. After validation on the experimental benchmark datasets, the proposed MSAAM attention module combining adaptive and multiscale strategies achieves the optimization of DEAM attention module in performance. The proposed method can effectively handle malware security issues.

6. Discussion

The proposed method does not perfectly solve the problem that texture similarity-based analysis is difficult to apply to real-world scenarios with complex malware families. In order to deepen the research on the effectiveness and universality of detecting malware, we will work on the deepening of feature engineering and the development of better attention modules. We also plan to optimize the information flow in this network by adding adjustments between different layers of the CliqueNet. The confrontation with code obfuscation technology is also a direction that needs to be studied in the future.

Data Availability

The BIG 2015 dataset can be obtained from https://www.kaggle.com/competitions/malware-classification/overview. The MalImg dataset can be obtained from https://www.researchgate.net/figure/Malimg-dataset_tbl2_323130489.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by NSFC under Grant no. 61572170, Natural Science Foundation of Hebei Province of China under Grant no. F2021205004, Science Foundation of Department of Human Resources and Social Security of Hebei Province under Grant no. 201901028 and no. ZD2021062, Science Foundation of Returned Overseas of Hebei Province of China under Grant no. C2020342, and Science and Technology Foundation Project of Hebei Normal University under Grant no. L2021K06.

References

[1] Trellix, "Trellix Threat Labs Research Report," 2021, https://www.mcafee.com/enterprise/en-us/lp/threats-reports/apr-2021.html.

[2] S. Khan and A. Akhunzada, "A hybrid DL-driven intelligent SDN-enabled malware detection framework for Internet of Medical Things (IoMT)," *Computer Communications*, vol. 170, pp. 209–216, 2021.

[3] M. Dib, S. Torabi, E. Bou-Harb, and C. Assi, "A multi-dimensional deep learning framework for iot malware classification and family attribution," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1165–1177, 2021.

[4] Q. Li, J. Mi, W. Li, J. Wang, and M. Cheng, "CNN-based malware variants detection method for internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 23, pp. 16946–16962, 2021.

[5] Y. Abhijit and M. Singh, "Malware detection based on opcode frequency," in *Proceedings of the 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pp. 646–649, IEEE, Ramanathapuram, India, May 2017.

[6] S. Gupta, H. Sharma, and K. Sarvjeet, "Malware characterization using Windows API call sequences," in *Proceedings of the International Conference on Security, Privacy, and Applied Cryptography Engineering*, pp. 363–378, Delhi, India, December 2016.

[7] D. Zou, Y. Wu, S. Yang et al., "IntDroid," *ACM Transactions on Software Engineering and Methodology*, vol. 30, no. 3, pp. 1–32, 2021.

[8] T. Shun, Y. Yukiko, S. Hajime, and I. Tomonori, "Malware detection with deep neural network using process behavior," in *Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, pp. 577–582, IEEE, Atlanta, GA, USA, June 2016.

[9] D. Javaheri and M. Hosseinzadeh, "A framework for recognition and confronting of obfuscated malwares based on memory dumping and filter drivers," *Wireless Personal Communications*, vol. 98, no. 1, pp. 119–137, 2018.

[10] D. E. García and N. DeCastro-García, "Optimal feature configuration for dynamic malware detection," *Computers & Security*, vol. 105, Article ID 102250, 2021.

[11] A. K. M. A. and J. C. D., "Automated multi-level malware detection system based on reconstructed semantic view of executables using machine learning techniques at VMM," *Future Generation Computer Systems*, vol. 79, pp. 431–446, 2018.

[12] A. Namavar Jahromi, S. Hashemi, A. Dehghantanha et al., "An improved two-hidden-layer extreme learning machine for malware hunting," *Computers & Security*, vol. 89, Article ID 101655, 2020.

[13] R. Sihwail, K. Omar, K. Zainol Ariffin, and S. Al Afghani, "Malware detection approach based on artifacts in memory image and dynamic analysis," *Applied Sciences*, vol. 9, no. 18, p. 3680, 2019.

[14] M. Ali, S. Shiaeles, G. Bendiab, and B. Ghita, "MALGRA: machine learning and N-gram malware feature extraction and detection system," *Electronics*, vol. 9, no. 11, p. 1777, 2020.

[15] Z. Cui, F. Xue, X. Cai, Y. Cao, G.-G. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.

[16] X. Ma, S. Guo, H. Li et al., "How to make attention mechanisms more practical in malware classification," *IEEE Access*, vol. 7, pp. 155270–155280, 2019.

[17] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, and S. Venkatraman, "Robust intelligent malware detection using deep learning," *IEEE Access*, vol. 7, pp. 46717–46738, 2019.

[18] J. Zhang, Z. Qin, H. Yin, L. Ou, and K. Zhang, "A feature-hybrid malware variants detection using CNN based opcode embedding and BPNN based API embedding," *Computers & Security*, vol. 84, pp. 376–392, 2019.

[19] S. Jeon and J. Moon, "Malware-detection method with a convolutional recurrent neural network using opcode sequences," *Information Sciences*, vol. 535, pp. 1–15, 2020.

[20] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the 8th International Symposium on Visualization for Cyber Security—VizSec'11*, pp. 1–7, Pittsburgh, PA, USA, July 2011.

[21] S. Venkatraman, M. Alazab, and R. Vinayakumar, "A hybrid deep learning image-based analysis for effective malware detection," *Journal of Information Security and Applications*, vol. 47, pp. 377–389, 2019.

[22] D. Gibert, C. Mateu, J. Planes, and R. Vicens, "Using convolutional neural networks for classification of malware

represented as images," *Journal of Computer Virology and Hacking Techniques*, vol. 15, no. 1, pp. 15–28, 2019.

[23] V. Verma, S. K. Muttoo, and V. B. Singh, "Multiclass malware classification via first- and second-order texture statistics," *Computers & Security*, vol. 97, Article ID 101895, 2020.

[24] B. Tamy and F. B. Marcus, "A.L(a)yingin(Test)Bed," in *Proceedings of the International Conference on Information Security*, pp. 381–401, New York, NY, USA, September 2019.

[25] L. Yang, P. Wang, H. Li, Z. Li, and Y. Zhang, "A holistic representation guided attention network for scene text recognition," *Neurocomputing*, vol. 414, pp. 67–75, 2020.

[26] M. Liu, L. Li, H. Hu, W. Guan, and J. Tian, "Image caption generation with dual attention mechanism," *Information Processing & Management*, vol. 57, no. 2, Article ID 102178, 2020.

[27] Y. Cheng and Y. Morimoto, "Triple-stage attention-based multiple parallel connection hybrid neural network model for conditional time series forecasting," *IEEE Access*, vol. 9, pp. 29165–29179, 2021.

[28] Y. Yang, C. Xu, F. Dong, and X. Wang, "A new multi-scale convolutional model based on multiple attention for image classification," *Applied Sciences*, vol. 10, no. 1, p. 101, 2019.

[29] Z. Huang, Y. Zhao, X. Li et al., "Application of innovative image processing methods and AdaBound-SE-DenseNet to optimize the diagnosis performance of meningiomas and gliomas," *Biomedical Signal Processing and Control*, vol. 59, Article ID 101926, 2020.

[30] J. Shi, K. Wu, C. Yang, and N. Deng, "A method of steel bar image segmentation based on multi-attention U-net," *IEEE Access*, vol. 9, pp. 13304–13313, 2021.

[31] J. Hu, Li Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7132–7141, IEEE, Salt Lake, UT, USA, June 2018.

[32] H. Yakura, S. Shinozaki, R. Nishimura, Y. Oyama, and J. Sakuma, "Neural malware analysis with attention mechanism," *Computers & Security*, vol. 87, Article ID 101592, 2019.

[33] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, Munich, Germany, September 2018.

[34] G. Huang, Y. Gong, Q. Xu, K. Wattanachote, K. Zeng, and X. Luo, "A convolutional attention residual network for stereo matching," *IEEE Access*, vol. 8, pp. 50828–50842, 2020.

[35] M. Zhu, L. Jiao, F. Liu, S. Yang, and J. Wang, "Residual spectral-spatial attention network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 1, pp. 449–462, 2021.

[36] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Efficient Channel attention for deep convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11531–11539, Francisco, CA, USA, June 2020.

[37] Z. Liu, J. Du, M. Wang, and S. S. Ge, "ADCM: attention dropout convolutional module," *Neurocomputing*, vol. 394, pp. 95–104, 2020.

[38] C. Wang, Z. Zhao, F. Wang, and Q. Li, "A novel malware detection and family classification scheme for IoT based on DEAM and DenseNet," *Security and Communication Networks*, vol. 2021, Article ID 6658842, 16 pages, 2021.

[39] L. Chen, Q. Sun, and F. Wang, "Attention-adaptive and deformable convolutional modules for dynamic scene deblurring," *Information Sciences*, vol. 546, pp. 368–377, 2021.

[40] Q. Hou, D. Zhou, and J. Fengi, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13713–13722, IEEE, Nashville, TN, USA, June 2021.

[41] Y. Jiang, H. Yao, C. Wu, and W. Liu, "A multi-scale residual attention network for retinal vessel segmentation," *Symmetry*, vol. 13, no. 1, p. 24, 2020.

[42] Y. Zhu, R. Yang, Y. He et al., "A lightweight multiscale Attention semantic segmentation algorithm for detecting laser welding defects on safety vent of power battery," *IEEE Access*, vol. 9, pp. 39245–39254, 2021.

[43] A. Sinha and J. Dolz, "Multi-scale self-guided attention for medical image segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 1, pp. 121–130, 2021.

[44] V. Ashish, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[45] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[46] S. Karen and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, IEEE, Kuala Lumpur, Malaysia, November 2014.

[47] S. Christian, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, IEEE, Boston, MA, USA, June 2015.

[48] H. Gao, "Deep networks with stochastic depth," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 646–661, Amsterdam, The Netherlands, October 2016.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, IEEE, Las Vegas, NV, USA, June 2016.

[50] H. Gao, Z. Liu, and L. Van Der Maaten, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, IEEE, Honolulu, HI, USA, July 2017.

[51] J. Hemalatha, S. Roseline, S. Geetha, S. Kadry, and R. Damaševičius, "An efficient DenseNet-based deep learning model for malware detection," *Entropy*, vol. 23, no. 3, p. 344, 2021.

[52] Y. Yang, Z. Zhong, T. Shen, and Z. Lin, "Convolutional neural networks with alternately updated clique," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2413–2422, IEEE, Salt Lake, UT, USA, June 2018.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 7125491, 11 pages
https://doi.org/10.1155/2022/7125491

Research Article

Cryptanalysis and Enhancement of an Authenticated Key Agreement Protocol for Dew-Assisted IoT Systems

Yuqian Ma, Yongliu Ma, and Qingfeng Cheng

State Key Laboratory of Mathematical Engineering and Advanced Computing,
Strategic Support Force Information Engineering University, Zhengzhou 450001, China

Correspondence should be addressed to Qingfeng Cheng; qingfengc2008@sina.com

Received 28 March 2022; Accepted 18 May 2022; Published 15 June 2022

Academic Editor: Qi Li

Real-time and high-efficient communication becomes a vital property for IoT-enabled equipment, since the application range of the Internet of Things has extended widely. At the same time, the centralized characterization of the cloud computing is gradually unable to meet the demand for both low latency and high computing efficiency. To resolve these issues, new computing paradigms have been introduced, such as edge, dew, and fog computing. Recently, Saurabh et al. introduced a mutual authentication protocol, which was claimed to resist various attacks without the requirement of a trusted server, for dew-assisted IoT devices. However, this paper will show that Saurabh et al.'s scheme lacks forward security and user anonymity. Then, a new authenticated key agreement (AKA) protocol, named e-SMDAS, will be put forward and formally proven secure under the eCK security model. Further, the analysis results of BAN logic and Scyther tool will also confirm the security of e-SMDAS. Finally, the comparative analysis of security features and computation efficiency between e-SMDAS and several recent schemes will be demonstrated at the end of this paper.

1. Introduction

Cloud computing, developing swiftly and violently, is gradually unable to satisfy the growing needs in the Internet. Flavio et al. [1] introduced the idea of fog computing. However, with the rapid development of the Internet, fog computing alone could not satisfy the quality of cloud-assisted services. Some other computing paradigms were proposed to meet the growing demand for high-quality cloud services. Tian et al. [2] recently proposed a framework for blockchain-assisted edge services in the Industrial Internet of Things (IIoT). The paradigm of dew computing was put forward by Wang [3, 4] to fully make use of on-premises devices and cloud services. Defined as an on-premises device software-hardware organization paradigm in the cloud computing environment, the dew computing, in which dew servers are independent of cloud servers when offline and collaborative with cloud servers when online, provides the functionality of high information processing and low latency communication. The system architecture of cloud-fog-dew computing is demonstrated in Figure 1.

To build a secure and flexible dew computing paradigm, many security features need to be considered. Besides the basic mutual authentication and session key confirmation features, protocols in this paradigm also require forward security which confirms the leakage of long-term secrets will not influence the session keys. Since communications between servers are closely related to users' privacy, anonymity and untraceability are also vital.

To achieve secure communication in the network driven by fog computing, Hameed et al. [5] proposed a scheme claiming that it could achieve mutual authentication, low consumption, and high efficiency in smart home case. In 2021, Liu et al. [6] proposed a distributed access control system based on the decentralized conception of fog computing and blockchain technology. A similar idea was also thought about by Shukla et al. [7], adopting a signature-based encryption algorithm to maximize the strength of fog computing and blockchain.

The application field of the Internet of Things (IoT) has extended largely in recent years. Aiming at protecting

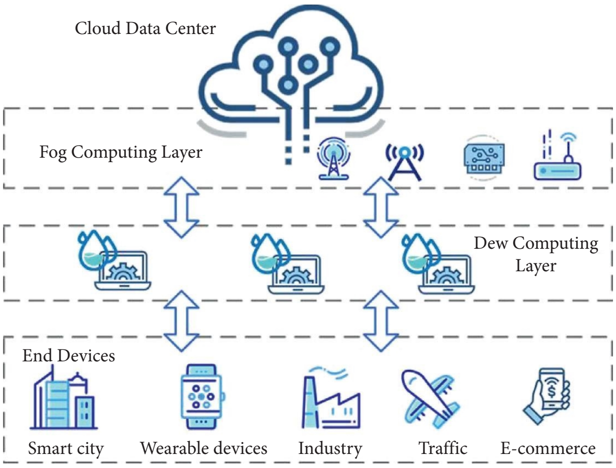


Figure 1: Fog computing system architecture.

the secrecy, integrity, and anonymity of IoT-assisted end devices, Singh and Chaurasiya [8] discussed a possible mutual authentication scheme for the vulnerable fog nodes. A combination of elliptic curve Diffie–Hellman ephemeral key exchange algorithm and preshared key was analyzed by Amanlou et al. [9] to achieve credible communication between the fog gateways and devices located in IoT.

Our contributions in this paper mainly consist of the following four points.

(i) We analyze an authenticated key agreement (AKA) protocol designed for a dew-assisted system by Saurabh et al. [10], referred to as SMDAS protocol below, and point out that their scheme lacks forward security and user anonymity.

(ii) Upon the analysis, we design a new AKA protocol, called e-SMDAS protocol below, remedying SMDAS protocol to achieve the mutual authentication, session key establishment, forward security, user anonymity, and other security features.

(iii) The security of our protocol is formally proven under the eCK security model and also confirmed using the Scyther tool and BAN logic.

(iv) Finally, results of comparison between the enhanced protocol and several recent schemes demonstrate the advantages of our protocol in the aspects of security features and communication efficiency.

The arrangement of this paper is as follows. Related works are first introduced in Section 2. In Section 3, we present some preliminaries used in the analysis of the proposed protocol. After reviewing the process of SMDAS protocol in Section 4, we analyze the security flaws of SMDAS protocol in Section 5. Our newly proposed protocol is described explicitly in Section 6; its formal security proof and security analysis using Scyther tool and BAN logic are provided in Section 7. Comparisons between the proposed protocol and SMDAS protocol are demonstrated in Section 8. Finally, in Section 9, the conclusion is highlighted.

2. Related Work

So far, anonymity and privacy-preserving are vital security features required urgently not only in dew computing paradigm but also in many other applications. To sum up the applications, several relative schemes [11–16] are listed in Table 1. They have been paid much attention to because of the decentralized feature of dew-assisted paradigm [17].

Recently, a lightweight anonymity client authentication scheme was proposed by Gaikwad et al. [18] adopting chaotic hash function. Moreover, Masud et al. [19] proposed a lightweight and physically secure mutual authentication and secret key establishment protocol preserving privacy for COVID-19 patients' care in the Internet of Medical Things. Their protocol used physical unclonable functions to make the network devices distinguish the legitimacy of doctors before acquiring a session key. Xiong et al. [20] proposed a three-party data privacy-preserving mechanism with game theory and machine learning technology. Tian et al. [21] proposed a graph clustering method to protect data privacy sharing in the Social Internet of Things (SIoT).

Besides, forward security is one of the main concerns for AKA protocols. In 2015, Chaudhry et al. [22] proposed a remote user authentication scheme. Regrettably, Ravanbakhsh et al. [23] claimed that Chaudhry et al.'s scheme was unable to achieve perfect forward security and proposed an authenticated communication scheme for Voice over Internet Protocol (VoIP). Later, Nikooghadam and Amintoosi [24] proved that Ravanbakhsh et al.'s scheme did not provide perfect forward security and put forward a two-factor AKA scheme with perfect forward security.

Recently, Saurabh et al. [10] introduced a mutual AKA protocol for the dew-assisted devices. They applied bilinear parings to achieve the mutual authentication and establishment of secure session keys. Formal analysis was presented by the use of AVISPA and the theory of security reduction. However, in this paper, we analyze the security of this protocol and show that it lacks forward security and user anonymity.

3. Preliminaries and Security Model

In this section, we concisely introduce the mathematical definitions and security model used next.

3.1. Mathematical Hard Problems

(i) *Elliptic Curve Discrete Logarithm (ECDL) Problem*: Given an elliptic curve E_p, an additive cyclic group G based on E_p, a generator P of G, and an element $Q = \text{aP}$ from G, it is hard to extract $a \in Z_p^*$ from Q and P.

(ii) *Elliptic Curve Computational Diffie–Hellman (ECCDH) Problem*: Given an elliptic curve E_p, an additive cyclic group G based on E_p, and a generator P of G, considering the elements $S = \text{aP}$ and $T = \text{bP}$ from G, it is hard to compute $U = \text{abP}$.

Table 1: The summary of schemes set in IoT systems.

Scheme	Settings applied in	Limitations
[11]	Wireless sensor networks	Vulnerable to insider attack
[12]		Vulnerable to secret key leakage and forgery attack
[13]	Telecare medicine information systems	Vulnerable to reflection attack
[14]		Vulnerable to replay attack
[15]		Vulnerable to offline password attack
[16]		Vulnerable to impersonation attack and users' identity leakage

3.2. Security Model. LaMacchia et al. [25] proposed the eCK security model in 2007. In this model, each entity owns two secrets, a long-term key x and an ephemeral key r. Assume two entities are A and B; their long-term keys are x_A, x_B; and their ephemeral keys are r_A, r_B, respectively. Besides, each session under the eCK security model has its own identity, denoted as $\mathrm{SID}^i_{A,B}$ if this session's owner is entity A. Then, the abilities of adversary, denoted as $\mathscr{A}$, can be defined through the queries below:

(i) Send(A, M): Through this query, $\mathscr{A}$ can send message M to entity A and get the corresponding message according to the protocol.

(ii) Reveal$(\mathrm{SID}^i_{A,B})$: Through this query, $\mathscr{A}$ can acquire the session key of $\mathrm{SID}^i_{A,B}$ if session $\mathrm{SID}^i_{A,B}$ has been completed. Otherwise, $\mathscr{A}$ will get nothing.

(iii) Ephemeral$(\mathrm{SID}^i_{A,B})$: Through this query, $\mathscr{A}$ can obtain the ephemeral key of the session $\mathrm{SID}^i_{A,B}$.

(iv) Longterm(A): Through this query, $\mathscr{A}$ can obtain the long-term key of entity A.

(v) Test$(\mathrm{SID}^i_{A,B})$: If $\mathscr{A}$ launches this query, session $\mathrm{SID}^i_{A,B}$ will randomly choose b from $\{0, 1\}$. If $b = 0$, $\mathrm{SID}^i_{A,B}$ will choose a random number from the set of keys and send it back to $\mathscr{A}$. If $b = 1$, $\mathrm{SID}^i_{A,B}$ will send the real session key back to $\mathscr{A}$.

To define a secure protocol in the eCK security model, a definition of freshness should be presented first since a secure game through Test$(\mathrm{SID}^i_{A,B})$ is querying toward a fresh session.

Definition 1. A session with identity $\mathrm{SID}^i_{A,B}$ in the eCK model at entity A whose intended partner denoted as B is fresh if the following items are satisfied:

(i) The session has not been asked for a Reveal query.

(ii) If a matching session exists with session identity $\mathrm{SID}^j_{B,A}$, then

  (i) not both Ephemeral$(\mathrm{SID}^i_{A,B})$ and Longterm(A) queries have been asked for;

  (ii) not both Ephemeral$(\mathrm{SID}^j_{B,A})$ and Longterm(B) queries have been asked for.

(iii) If no partner exists, then

  (i) not both Ephemeral$(\mathrm{SID}^i_{A,B})$ and Longterm(A) queries have been asked for;

  (ii) Longterm(B) queries have not been asked for.

Based on this definition, we present the definition of a secure session in the eCK security model.

Definition 2. The advantage of the adversary $\mathscr{A}$ in the secure game with AKA protocol Π is defined as $\mathrm{Adv}^{\mathrm{AKA}}_{\Pi}(\mathscr{A}) = \Pr[A\,\mathrm{wins}] - 1/2$.

If the matching session of Π computes the same session key and no efficient adversary $\mathscr{A}$ has more than a negligible advantage in winning the secure game, then the protocol Π is secure under the eCK security model.

4. Review of SMDAS Protocol

In this section, we review the registration and session key distribution phases of SMDAS protocol [10]. There are three types of entities participating in SMDAS protocol, namely, a sensor node SN_i, a dew server DS_j, and a cloud server S. Notations used in SMDAS protocol are listed in Table 2.

4.1. Registration Phase. Firstly, the cloud server S initializes this system according to the following steps.

(i) S selects an appropriate elliptic curve E over a finite field F_q and then selects G, a subgroup of E, whose order is n. P is a group generator of G.

(ii) S randomly chooses $s \in Z^*_n$ and calculates $X = s\mathrm{P}$, $A = e(P, P)^s$.

(iii) Finally, S publishes the public parameters $\{E, G, A, n, P, X\}$ and keeps s as its own secret key securely.

4.2. Dew Server Registration Phase. Assume that there are m dew servers and each one is denoted as DS_j, $j \in \{1, 2, \ldots, m\}$. These servers select their own identities $\mathrm{ID}_{\mathrm{DS}_j}$. When a dew server registers to the cloud server, it sends its identity $\mathrm{ID}_{\mathrm{DS}_j}$ to S. After receiving DS_j's identity, S will compute $\mathrm{SID}_{\mathrm{DS}_j}$ for DS_j, where $\mathrm{SID}_{\mathrm{DS}_j} = s(X + P \cdot h(\mathrm{ID}_{\mathrm{DS}_j}))$.

4.3. Sensor Node Registration Phase. Every sensor node, denoted as SN_i, has its own identity $\mathrm{ID}_{\mathrm{SN}_i}$ and password $\mathrm{PW}_{\mathrm{SN}_i}$. When the sensor node needs to register to S, it firstly computes $\mathrm{SH}_1 = h(\mathrm{ID}_{\mathrm{SN}_i} \| \mathrm{PW}_{\mathrm{SN}_i})$ and sends message $\mathrm{ID}_{\mathrm{SN}_i}, H_1$ to S. Upon receiving the registration request from SN_i, S verifies $\mathrm{ID}_{\mathrm{SN}_i}$ to confirm SN_i is an unregistered node. Then, S computes $I = h(\mathrm{ID}_{\mathrm{SN}_i} \| s)$, $H_2 = I \oplus H_1$, $\mathrm{SID}_{\mathrm{SN}_i} = s(P + I)$. After computing, S stores SID_i and sends message $H_2, \mathrm{SID}_{\mathrm{SN}_i}$ to SN_i. When SN_i receives message from S, it computes $I = H_2 \oplus H_1$ and stores $\mathrm{SID}_{\mathrm{SN}_i}, I$.

Table 2: Notations applied in SMDAS protocol.

Parameters	Description
S	The cloud/fog server
SN_i	The sensor node i
DS_j	The dew server j
$\mathrm{ID}_{\mathrm{SN}_i}, \mathrm{ID}_{\mathrm{DS}_j}$	The identity of SN_i, DS_j, respectively
S	The secret key of S
$\mathrm{PW}_{\mathrm{SN}_i}$	The password of sensor node i
T_i, T_j	Timestamp generated by SN_i, DS_j, respectively
$h(\ )$	One-way hash function defined from Z_n^* to Z_n^*
$\mathscr{A}$	The adversary

4.4. Session Key Distribution Phase. After DS_j and SN_i register to S, they can establish a session with $\mathrm{SID}_{\mathrm{SN}_i}$ and $\mathrm{SID}_{\mathrm{DS}_j}$. The detailed steps are described below.

(i) SN_i randomly chooses $r_u \in Z_n^*$ and computes the corresponding public key $R_u = r_u P$ and $Z = A^{r_u}$. Then, SN_i calculates the elements of message as follows: $M = R_u + (X + P \cdot h(\mathrm{ID}_{\mathrm{DS}_j}))$, $N = h(Z) \oplus \mathrm{ID}_{\mathrm{SN}_i}$, $Q = h(Z \| \mathrm{ID}_{\mathrm{SN}_i} \| X) \oplus R_u$, $S = \mathrm{SID}_{\mathrm{SN}_i} \oplus h(R_u \| \mathrm{ID}_{\mathrm{SN}_i} \| \mathrm{TS}_i \| Z)$, $J = h(\mathrm{SID}_{\mathrm{SN}_i} \mathrm{SR}_u \mathrm{NQID}_{\mathrm{SN}_i} \mathrm{TS}_i)$. SN_i sends $M, N, Q, S\ J, \mathrm{TS}_i$, where TS_i is the current timestamp.

(ii) DS_j computes Z', $\mathrm{ID}_{\mathrm{SN}_i}'$, R_u', and SID_i'. According to these parameters, DS_j verifies whether J' equals J. DS_j randomly selects $y \in Z_n^*$ and computes the public key $Y = \mathrm{yP}$. Then, DS_j calculates $T_j = h(\mathrm{ID}_{\mathrm{SN}_i}' \| \mathrm{ID}_{\mathrm{DS}_j} \| R_u' | Y | \mathrm{TS}_j)$, $F = \mathrm{SID}_{\mathrm{SN}_i}' \oplus T_j$, $\mathrm{SK} = h(\mathrm{SID}_{\mathrm{SN}_i}'\ \| T_j \| \mathrm{TS}_j)$, $V_e = h(\mathrm{SK} \Big\| T_j \| F \| \mathrm{TS}_j)$. DS_j sends message TS_j, V_e, F, where TS_j is the current timestamp and stores the session key SK.

(iii) SN_i computes T_j', SK', and V_e'. According to these parameters, SN_i verifies whether V_e' equals V_e. If it succeeds, SN_i accepts SK' as the session key.

5. Cryptanalysis of SMDAS Protocol

In this section, we present two security flaws of SMDAS protocol as the adversary $\mathscr{A}$ can acquire private key of SN_i and DS_j through Extract($\mathrm{ID}_{\mathrm{SN}_i}$) and Extract($\mathrm{ID}_{\mathrm{DS}_j}$), respectively, mentioned in [10].

5.1. Lack of Forward Security. In this subsection, we demonstrate if the private key of sensor node i is compromised; then, the session key will be easily recovered by the adversary $\mathscr{A}$:

(i) In the session key distribution phase, $\mathscr{A}$ eavesdrops the message from dew server to sensor node, TS_j, V_e, F.

(ii) $\mathscr{A}$ launches Extract query to the sensor node SN_i and acquires SN_i's private secret keys $\mathrm{SID}_{\mathrm{SN}_i}$.

(iii) After obtaining the parameters above, $\mathscr{A}$ can extract T_j' by $T_j' = F \oplus \mathrm{SID}_{\mathrm{SN}_i}$ and the session key according to the way generating $\mathrm{SK} = h(\mathrm{SID}_{\mathrm{SN}_i} \| T_j' \| \mathrm{TS}_j)$.

Thus, in this way, adversary $\mathscr{A}$ can recover the session key. It can be concluded that the steps described are in accordance with the definition of weak forward security.

5.2. Lack of User Anonymity. We point out an efficient method to prove that SMDAS protocol lacks user anonymity in this subsection by compromising the private key of dew server following the steps below.

(i) $\mathscr{A}$ first eavesdrops the message $M, N, Q, S, J, \mathrm{TS}_i$.

(ii) Then, $\mathscr{A}$ launches Extract($\mathrm{ID}_{\mathrm{DS}_j}$) to get the private key of DS_j, $\mathrm{SID}_{\mathrm{DS}_j}$.

(iii) In this way, $\mathscr{A}$ can compute $Z' = e(M, X)/e(\mathrm{SID}_{\mathrm{DS}_j}, P)$.

(iv) Finally, the adversary can derive the identity of SN_i as $\mathrm{ID}_{\mathrm{SN}_i} = N \oplus h(Z')$.

When the adversary implements the attack described above, $\mathscr{A}$ can easily get the identity of the sensor node. This means SMDAS protocol can hardly protect the anonymity of users.

6. e-SMDAS Protocol

In this section, we propose a new anonymity and secure mutual AKA protocol remedying the flaws of SMDAS protocol, which we call e-SMDAS protocol.

There are three main phases in the proposed protocol, namely, initialization phase, registration phase, and secure session key establishment phase. Particularly, the registration phase can be divided into two parts, the sensor node registration phase and the dew server registration phase. In Table 3, the notations applied in the proposed protocol are presented.

6.1. Initialization Phase. The cloud server, also the registration server, acts as the trusted authority. It first selects a suitable cyclic group G based on an elliptic curve E. The order of the group is the prime p and the generator of the group is P. Then, the server randomly selects $s \in Z_p^*$ as its master key while it computes its public key $X = \mathrm{sP}$ accordingly and defines the three hash functions h_1, h_2, h_3. Finally, the server publishes the public parameters $\{E, G, P, X, p, h_1, h_2, h_3\}$ to initialize the system and keeps s secretly.

6.2. Registration Phase. Before sensor nodes and dew servers are put into usage, they must be registered in the cloud server first to acquire their long-term keys in the further communications. Both the sensor node registration phase and the dew server registration phase are described as follows.

6.2.1. Sensor Node Registration Phase. Before SN_i registers in the cloud server S, SN_i should first choose its identity $\mathrm{ID}_{\mathrm{SN}_i}$ and password $\mathrm{PW}_{\mathrm{SN}_i}$. Then, SN_i can begin the registration phase as it first sends the registration request to the cloud server S.

Table 3: Notations applied in e-SMDAS protocol.

Parameters	Description	Parameters	Description
λ	The security parameter	$e_{\mathrm{SN}_i}, e_{\mathrm{DS}_j}$	The ephemeral private keys of SN_i, DS_j, respectively
S	The cloud/fog server	T_1, T_2	The timestamps
SN_i	The sensor node i	h_1	One-way hash function defined from $\{0,1\}^*$ to Z_p^*
DS_j	The dew server j	h_2	One-way hash function defined from $\{0,1\}^*$ to $\{0,1\}^l$, where l is the length of session key
S	The secret key of S	h_3	One-way hash function defined from $\{0,1\}^*$ to $\{0,1\}^{2\lambda}$
$\mathrm{ID}_{\mathrm{SN}_i}, \mathrm{ID}_{\mathrm{DS}_j}$	The identity of SN_i, DS_j, respectively	$\mathscr{A}$	The adversary

(i) SN_i first chooses its identity $\mathrm{ID}_{\mathrm{SN}_i}$ and password $\mathrm{PW}_{\mathrm{SN}_i}$. It randomly selects l_{SN_i} in Z_p^* and computes $H_1 = h_1(\mathrm{ID}_{\mathrm{SN}_i} \| \mathrm{PW}_{\mathrm{SN}_i} \| l_{\mathrm{SN}_i})$. Finally, SN_i sends message $\mathrm{ID}_{\mathrm{SN}_i}, H_1$ to S.

(ii) After receiving $\mathrm{ID}_{\mathrm{SN}_i}, H_1$ from SN_i, S first checks if this identity has ever been registered. If it has not, then the server computes $L_{\mathrm{SN}_i} = (\mathrm{sH}_1)P$. After finishing computation, S sends message L_{SN_i} back to SN_i.

(iii) After getting L_{SN_i} from S, SN_i stores $\left\{L_{\mathrm{SN}_i}, H_1\right\}$ as its long-term key securely and deletes l_{SN_i} timely.

6.2.2. Dew Server Registration Phase. Just as the sensor node registration phase, the dew server DS_j first registers in the cloud server S. DS_j operates the following steps for registration:

(i) DS_j randomly selects l_{DS_j} in Z_p^* and computes $P_{\mathrm{DS}_j} = l_{\mathrm{DS}_j} P$. Then, it sends its identity $\mathrm{ID}_{\mathrm{DS}_j}$ and P_{DS_j} to S in a secure channel.

(ii) After receiving the message from DS_j, S first checks whether the $\mathrm{ID}_{\mathrm{DS}_j}$ has been registered. If it has not, S generates the long-term key for the dew server. S computes $H_2 = h_1(\mathrm{ID}_{\mathrm{DS}_j} \| P_{\mathrm{DS}_j})$, $L_{\mathrm{DS}_j} = (\mathrm{sH}_2)P$ and sends L_{DS_j} to DS_j.

(iii) On receiving the message from S, DS_j stores $\left\{L_{\mathrm{DS}_j}, l_{\mathrm{DS}_j}\right\}$ securely and publishes P_{DS_j}.

6.3. Secure Session Establishment Phase. After registering in the cloud server, both the sensor node SN_i and the dew server DS_j get their long-term keys. Then, they can establish their session key through the following steps, also illustrated in Figure 2.

(i) SN_i randomly chooses $e_{\mathrm{SN}_i} \in Z_p^*$ and computes the corresponding public key $E_{\mathrm{SN}_i} = e_{\mathrm{SN}_i} P$. Then, SN_i computes $C_1 = h_1(L_{\mathrm{SN}_i} \| \mathrm{ID}_{\mathrm{SN}_i})$, $A = (\mathrm{ID}_{\mathrm{SN}_i} \| L_{\mathrm{SN}_i}) \oplus h_3(e_{\mathrm{SN}_i} P_{\mathrm{DS}_j} \| T_1)$. SN_i sends message $M_1 = A, E_{\mathrm{SN}_i}, T_1$ to DS_j as the request for service, where T_1 is the present timestamp.

(ii) On receiving message from SN_i, DS_j first checks the freshness of the timestamp T_1. Then, it computes $E_{\mathrm{SN}_i} l_{\mathrm{DS}_j}$ and $\mathrm{ID}_{\mathrm{SN}_i} \| L_{\mathrm{SN}_i} = A \oplus h_3(E_{\mathrm{SN}_i} l_{\mathrm{DS}_j} \| T_1)$. If it succeeds, DS_j can obtain $\mathrm{ID}_{\mathrm{SN}_i} \| L_{\mathrm{SN}_i}$, by utilizing which it can compute C_1'. DS_j randomly selects $e_{DS_j} \in Z_p^*$ and computes $E_{\mathrm{DS}_j} = e_{\mathrm{DS}_j} P$, $T_{\mathrm{DS}} = h_3(e_{\mathrm{DS}_j} E_{\mathrm{SN}_i} \| T_2)$ as well as the session key $\mathrm{SK}_{\mathrm{DtS}} = h_2(C_1' | T_{\mathrm{DS}} | T_2)$. Finally, DS_j computes $C_2 = E_{\mathrm{DS}_j} \oplus L_{\mathrm{SN}_i} \oplus \mathrm{ID}_{\mathrm{SN}_i}$, $B = h_1(\mathrm{SK}_{\mathrm{DtS}} \| T_2)$ and sends message $M_2 = B, C_2, T_2$.

(iii) After receiving the message from DS_j, SN_i computes $E_{\mathrm{DS}_j} = C_2 \oplus L_{\mathrm{SN}_i} \oplus \mathrm{ID}_{\mathrm{SN}_i}$, $T_{\mathrm{SN}} = h_3(e_{\mathrm{SN}_i} E_{\mathrm{DS}_j} \| T_2)$ and the session key $\mathrm{SK}_{\mathrm{StD}} = h_2(C_1 | T_{\mathrm{SN}} | T_2)$. Finally, it verifies whether the equality $B = h_1(\mathrm{SK}_{\mathrm{StD}} \| T_2)$ is right.

Hence, both the sensor node SN_i and the dew server DS_j get the same session key:

$$T_{\mathrm{DS}} = h_3\left(e_{\mathrm{DS}_j} E_{\mathrm{SN}_i} \| T_2\right) = h_3\left(e_{\mathrm{DS}_j} e_{\mathrm{SN}_i} P \| T_2\right) = h_3\left(e_{\mathrm{SN}_i} E_{\mathrm{DS}_j} \| T_2\right) = T_{\mathrm{SN}}. \tag{1}$$

In this way, if the dew server is the right potential partner, it can correctly calculate C_1'. SN_i and DS_j can obtain the same session key apparently according to the equality bellow:

$$\mathrm{SK}_{\mathrm{DtS}} = h_2\left(C_1' \middle| T_{\mathrm{DS}} \middle| T_2\right) = h_2\left(C_1 \middle| T_{\mathrm{SN}} \middle| T_2\right) = \mathrm{SK}_{\mathrm{StD}}. \tag{2}$$

7. Security Proof

This section provides the proof of the security of e-SMDAS protocol by three methods. Firstly, we prove the proposed protocol security under the eCK security model. Then, we present a further security attribute analysis using the Scyther tool. Finally, by using BAN logic, we deduce the final security goals.

7.1. Security Theorem. We have proven the correctness of the proposed protocol above; in this subsection, we will prove the security of e-SMDAS protocol.

Theorem 1. *Let $\mathscr{A}$ be a probabilistic polynomial time adversary against the proposed protocol Π with a time bound t, making at most q_s. Send queries q_{h_1}, q_{h_2}, q_{h_3} random oracle queries. Then,*

$$\mathrm{Adv}_{\Pi}(\mathscr{A}) \leq \frac{q_s}{2^{\lambda-2}} + \frac{q_s}{2^{2\lambda-2}} + \frac{2^{\lambda} \cdot q_{h_1}^2 + q_{h_3}^2}{2^{2\lambda}} + \frac{q_{h_2}^2}{2^l} + 2 q_{h_2} q_s^2 \mathrm{Adv}^{\mathrm{ECCDH}}(\mathscr{S}), \tag{3}$$

where $Adv^{ECCDH}(\mathscr{S})$ means the success probability of solving an instance of ECCDH problem by an algorithm $\mathscr{S}$.

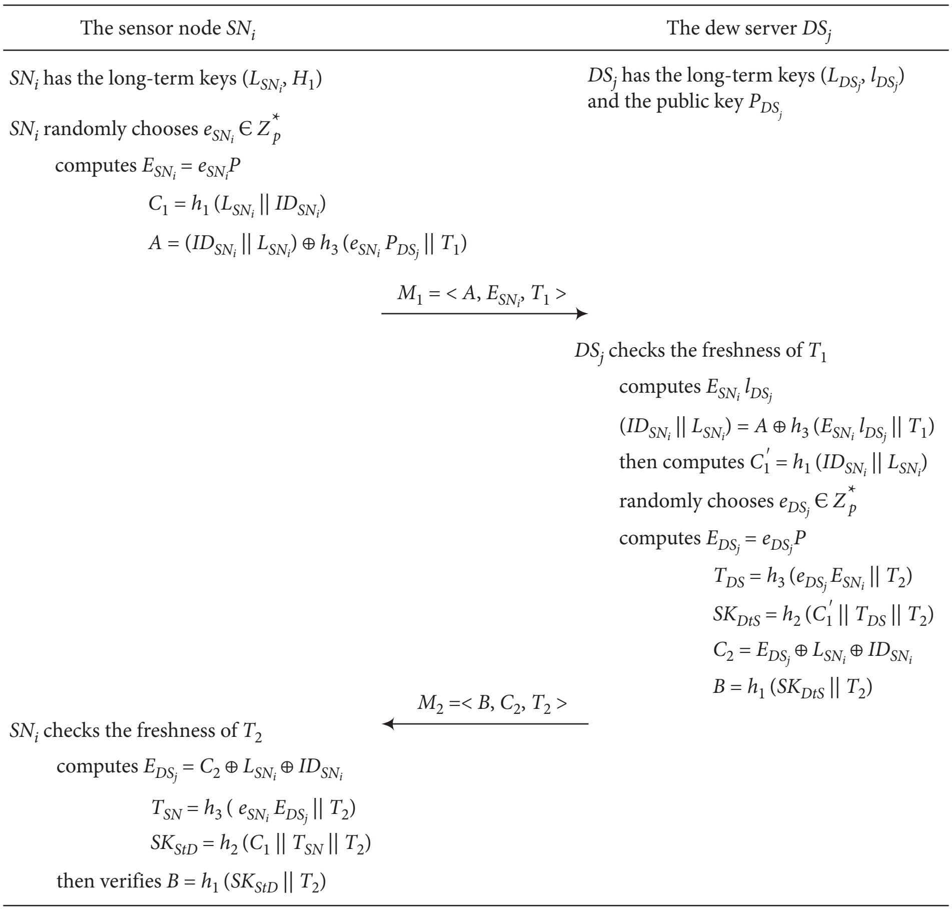

The sensor node SN_i
The dew server DS_j
SN_i has the long-term keys (L_{SN_i}, H_1)
DS_j has the long-term keys (L_{DS_j}, l_{DS_j}) and the public key P_{DS_j}
SN_i randomly chooses $e_{SN_i} \in Z_p^*$
computes $E_{SN_i} = e_{SN_i}P$
$C_1 = h_1 (L_{SN_i} || ID_{SN_i})$
$A = (ID_{SN_i} || L_{SN_i}) \oplus h_3 (e_{SN_i} P_{DS_j} || T_1)$
$M_1 = < A, E_{SN_i}, T_1 >$
DS_j checks the freshness of T_1
computes $E_{SN_i} l_{DS_j}$
$(ID_{SN_i} || L_{SN_i}) = A \oplus h_3 (E_{SN_i} l_{DS_j} || T_1)$
then computes $C_1' = h_1 (ID_{SN_i} || L_{SN_i})$
randomly chooses $e_{DS_j} \in Z_p^*$
computes $E_{DS_j} = e_{DS_j}P$
$T_{DS} = h_3 (e_{DS_j} E_{SN_i} || T_2)$
$SK_{DtS} = h_2 (C_1' || T_{DS} || T_2)$
$C_2 = E_{DS_j} \oplus L_{SN_i} \oplus ID_{SN_i}$
$B = h_1 (SK_{DtS} || T_2)$
$M_2 =< B, C_2, T_2 >$
SN_i checks the freshness of T_2
computes $E_{DS_j} = C_2 \oplus L_{SN_i} \oplus ID_{SN_i}$
$T_{SN} = h_3 (e_{SN_i} E_{DS_j} || T_2)$
$SK_{StD} = h_2 (C_1 || T_{SN} || T_2)$
then verifies $B = h_1 (SK_{StD} || T_2)$

Figure 2: Secure session establishment phase of e-SMDAS protocol.

Proof of Theorem 1: Next, we will prove the security of the proposed protocol through defining a sequence of hybrid experiments where $\mathscr{A}$ correctly guesses the random bit b in the Test query. Specifically, each experiment has a definition of Succ_i to illustrate the advantage.

(i) **Experiment 0**: This experiment simulates the situation of the attacks against the real protocols in the random oracle model. According to the definition, there exists $\text{Adv}(\mathscr{A}) = 2\Pr[\text{Succ}_0] - 1$, which means the origin advantage of adversary.

(ii) **Experiment 1**: In this experiment, $\mathscr{S}$ simulates the random oracles h_1, h_2, and h_3 by keeping hash lists L_{h_1}, L_{h_2}, L_{h_3} as follows:

  (i) If there exists a record of message M as (M, H) in the list L_{h_1}, it returns H. Otherwise, it selects an element H, adds the record (M, H) to the list L_{h_1}, and then returns H.

  (ii) If there exists a record of message M as (M, K) in the list L_{h_2}, it returns K. Otherwise, it selects an element K in the key set, adds the record (M, K) to the list L_{h_2}, and then returns K.

  (iii) If there exists a record of message M as (M, J) in the list L_{h_3}, it returns J. Otherwise, it selects an element J in the key set, adds the record (M, J) to the list L_{h_3}, and then returns J.

The Send, Reveal, Longterm, Ephemeral, and Test queries are also simulated as the real attack. Thus, this experiment is same as the real experiment, which means that the equation $\Pr[\text{Succ}_1] = \Pr[\text{Succ}_0]$ holds.

(i) **Experiment 2**: In this experiment, we simulate all oracles the same as **Experiment 1** except that a collision occurs in the output of the oracle h_1 or the session transcripts. According to the birthday paradox, the probability of collisions in the output of the oracle h_1 is at most $q_{h_1}^2/2^{\lambda+1}$, where q_{h_1} is the maximum times of queries to h_1. The same deduction can be applied to h_2 and h_3. Therefore, the successful probability of **Experiment 2** satisfies $\Pr[\text{Succ}_2] - \Pr[\text{Succ}_1] \leq q_{h_1}^2/2^{\lambda+1} + q_{h_3}^2/2^{2\cdot\lambda+1} + q_{h_2}^2/2^{l+1}$.

(ii) **Experiment 3**: In this experiment, the protocol will not halt except that $\mathscr{A}$ successfully guesses C_1 or T_{DS} (T_{SN}) without querying h_1 or h_3. Therefore, there exists $\Pr[\text{Succ}_3] - \Pr[\text{Succ}_2] \leq 2 \cdot q_s/2^{\lambda} + 2 \cdot q_s/2^{2\cdot\lambda}$.

(iii) **Experiment 4**: In this experiment, we only consider the situation where $\mathscr{A}$ exactly chooses a random session as the test session. Besides, the computation of the test session key is modified to select a random key from the key set. Consequently, the difference between **Experiment 3** and **Experiment 4** is in the event when $\mathscr{A}$ queries the tuple $(C_1|T_{\mathrm{SN}}|T_2)$ or $(C_1|T_{\mathrm{DS}}|T_2)$ to h_2 in the test session. To describe this difference, the following four cases may be considered:

(i) Longterm(SN_i) and Longterm(DS_j) are queried, from which $\mathscr{A}$ can obtain the long-term key L_{SN_i} of SN_i and l_{DS_j}, L_{DS_j} of DS_j. To calculate the session key, either e_{SN_i} or e_{DS_j} is required.

(ii) Longterm(SN_i) and Ephemeral(DS_j) are queried, from which $\mathscr{A}$ can obtain the long-term key L_{SN_i} of SN_i and e_{DS_j} of DS_j. To calculate the session key, e_{SN_i} is required.

(iii) Ephemeral(SN_i) and Longterm(DS_j) are queried, from which $\mathscr{A}$ can obtain the long-term key e_{SN_i} of SN_i and l_{DS_j}, L_{DS_j} of DS_j. To calculate the session key, e_{DS_j} is required.

(iv) Ephemeral(SN_i) and Ephemeral(DS_j) are queried, from which $\mathscr{A}$ can obtain the long-term key e_{SN_i} of SN_i and e_{DS_j} of DS_j. To calculate the session key, L_{SN_i} and l_{DS_j} are required.

If any of these four cases happens, then referring to the method proposed in [26], we can construct an algorithm $\mathscr{S}$ to solve an instance of ECCDH problem, and there exists

$$\Pr[\mathrm{Succ}_4] - \Pr[\mathrm{Succ}_3] \leq q_{h_2} q_s^2 \mathrm{Adv}^{\mathrm{ECCDH}}(\mathscr{S}). \tag{4}$$

Besides, in **Experiment 4**, to guess the bit b in the Test query is random, and other sessions do not matter. Therefore, there exists $\Pr[\mathrm{Succ}_4] = 1/2$. □

7.2. Scyther Security Analysis. Besides proving the security of the proposed model formally, we also use Scyther tool to show the proposed protocol is secure against various attacks. The setting used is presented in Figure 3 to achieve highly strong security, including perfect forward security, resistance to session key reveal attack, and resistance to ephemeral key leakage attack.

The result of analysis is demonstrated in Figure 4. According to Figure 4, we can clearly infer that under the setting predefined, the session key is secure against various attacks.

7.3. BAN Logic Formalized Security Proof. In this subsection, we provide another method to analyze the security of e-SMDAS protocol.

Next, we will prove that the proposed protocol can achieve the mutual authentication and two participants can obtain the same session key. We first present the security goals using BAN logic followed. We simplify the sensor node SN_i as N, and the dew server DS_j as D.

(i) $\mathbf{G}_1 N$ believes $(N \overset{\mathrm{SK}}{\leftrightarrow} D)$.
(ii) $\mathbf{G}_2 D$ believes $(N \overset{\mathrm{SK}}{\leftrightarrow} D)$.
(iii) $\mathbf{G}_3 N$ believes (D believes $(N \overset{\mathrm{SK}}{\leftrightarrow} D)$).
(iv) $\mathbf{G}_4 D$ believes (N believes $(N \overset{\mathrm{SK}}{\leftrightarrow} D)$).

Then, we formalize the original messages into the idealized ones as follows:

(i) $\mathbf{M}_1 N \longrightarrow D$: $\{N, L_N, T_1\}_{e_N \cdot P_D}$, $\xrightarrow{K_2} N, T_1$.
(ii) $\mathbf{M}_2 D \longrightarrow N$: $T_{2K_{\mathrm{ND}}}$, $\xrightarrow{K_2} D_{L_N}, T_2$.

Thirdly, we make the initial assumptions.

(i) $\mathbf{A}_1 D$ believes (fresh (T_1)).
(ii) $\mathbf{A}_2 N$ believes (fresh (T_2)).
(iii) $\mathbf{A}_3 D$ believes $(N \overset{E_N}{\Leftrightarrow} E_N D)$.
(iv) $\mathbf{A}_4 N$ believes $(D \overset{E_D}{\Leftrightarrow} E_D N)$.
(v) $\mathbf{A}_5 D$ believes (N controls $(N \overset{K_{\mathrm{ND}}}{\leftrightarrow} D)$).
(vi) $\mathbf{A}_6 N$ believes (D controls $(N \overset{K_{\mathrm{ND}}}{\leftrightarrow} D)$).

Finally, following the idealized messages, we utilize the predefined notations, rules, and assumptions to deduce the goals of the proposed protocol. The proof process is presented as follows:

(i) From $\mathbf{M}_1$, we can derive the formula $\mathbf{F}_1$ as follows:

(1) $\mathbf{F}_1 D$ sees $(\{N, L_N, T_1\}_{e_N \cdot P_D}, \xrightarrow{K_2} N, T_1)$.

(ii) According to $\mathbf{R}_4$ and $\mathbf{F}_1$, we can deduce the formula $\mathbf{F}_2 \sim \mathbf{F}_4$ as follows:

(1) $\mathbf{F}_3 D$ sees (T_1).
(2) $\mathbf{F}_4 D$ sees $(\xrightarrow{K_2} N)$.

(iii) According to $\mathbf{R}_1$, $\mathbf{A}_3$, and $\mathbf{F}_2$, we can deduce the formula $\mathbf{F}_5$ as follows:

(1) $\mathbf{F}_5 D$ believes (N said (N, K_N, T_1)).

(iv) According to $\mathbf{F}_5$, $\mathbf{A}_1$, and $\mathbf{R}_2$, we can deduce the formula $\mathbf{F}_6 \sim \mathbf{F}_8$ as follows:

(1) $\mathbf{F}_6 D$ believes (N believes (N, K_N, T_1)).
(2) $\mathbf{F}_7 D$ believes (N believes (N, K_N)).
(3) $\mathbf{F}_8 D$ believes (N believes (C_1)).

(v) From $\mathbf{M}_2$, we can derive the formula $\mathbf{F}_9$ below:

(1) $\mathbf{F}_9 N$ sees $(T_{2_{K_{\mathrm{ND}}}}, \xrightarrow{K_1} N, N_{E_D}, T_2)$.

(vi) According to $\mathbf{R}_4$, $\mathbf{A}_2$ and $\mathbf{F}_9$, we can deduce the formula $\mathbf{F}_{10}$, $\mathbf{F}_{11}$, and $\mathbf{F}_{12}$:

(1) $\mathbf{F}_{10} N$ sees $(T_{2K_{\mathrm{ND}}})$.
(2) $\mathbf{F}_{11} N$ sees $(\xrightarrow{K_1} N, N_{E_D})$.
(3) $\mathbf{F}_{12} N$ believes (fresh $0 \xrightarrow{K_1} N, N_{E_D}$).

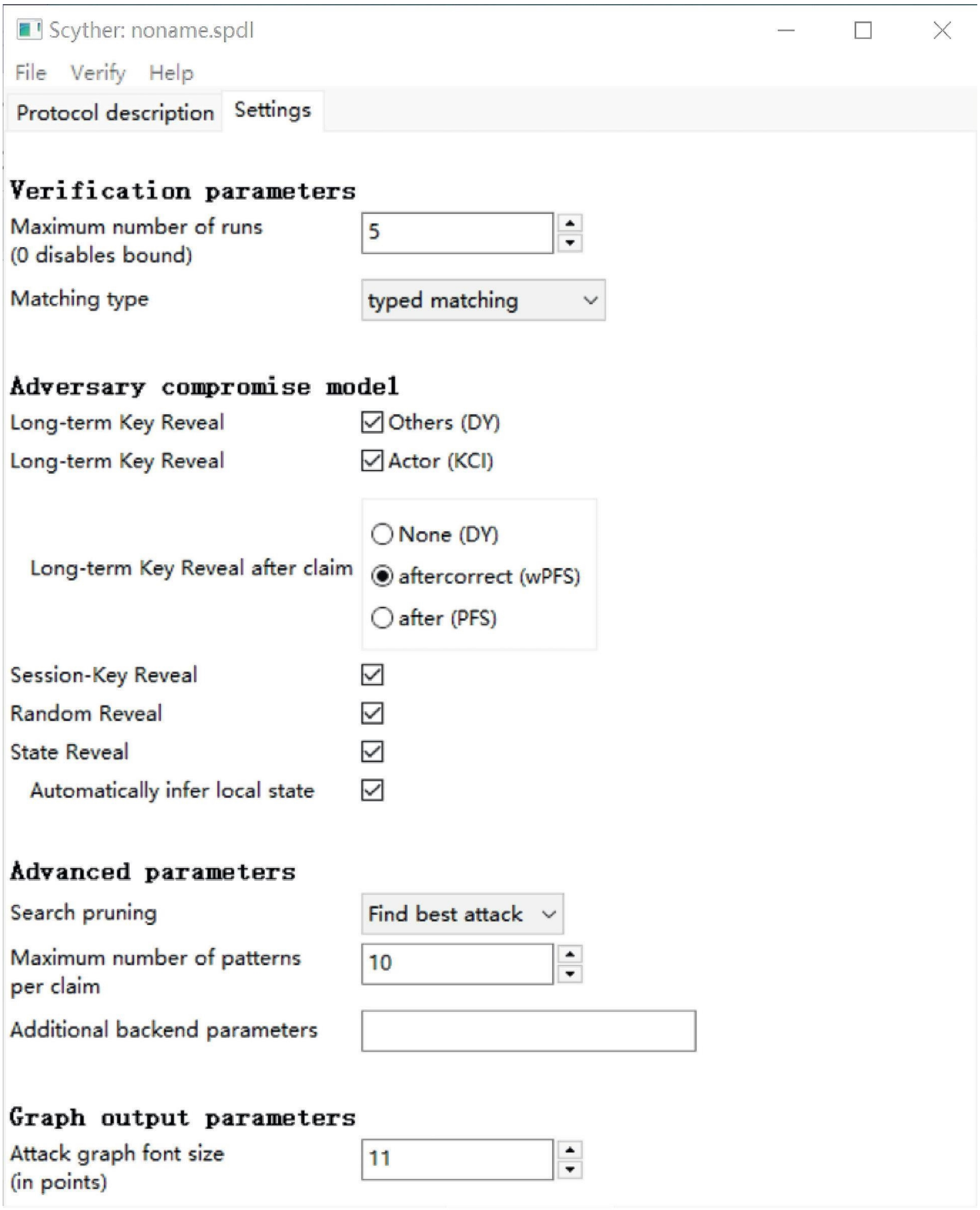
(vii) According to $\mathbf{F}_{11}$, $\mathbf{A}_4$, and $\mathbf{R}_1$, we can deduce the formula $\mathbf{F}_{13}$:

(1) $\mathbf{F}_{14} N$ believes (D said $(\xrightarrow{K_1} N, N)$).

(viii) According to $\mathbf{F}_{13}$, $\mathbf{F}_{12}$, and $\mathbf{R}_2$, we can deduce the formula $\mathbf{F}_{14} \sim \mathbf{F}_{15}$:

(1) $\mathbf{F}_{14} N$ believes (D believes (L_N, N)).
(2) $\mathbf{F}_{15} N$ believes (D believes (C_1)).

(ix) Since $K_{\mathrm{ND}} = h_2(C_1 || h_3(e_N e_D P \| T_2) || T_2)$, we can deduce the formula $\mathbf{F}_{16}$ according to $\mathbf{F}_{15}$, $\mathbf{A}_2$, and $\mathbf{A}_4$, which is also $\mathbf{G}_3$:



Verification parameters

Maximum number of runs (0 disables bound): 5

Matching type: typed matching

Adversary compromise model

Long-term Key Reveal: ☑ Others (DY)

Long-term Key Reveal: ☑ Actor (KCI)

Long-term Key Reveal after claim: ○ None (DY) ● aftercorrect (wPFS) ○ after (PFS)

Session-Key Reveal: ☑

Random Reveal: ☑

State Reveal: ☑

Automatically infer local state: ☑

Advanced parameters

Search pruning: Find best attack

Maximum number of patterns per claim: 10

Additional backend parameters:

Graph output parameters

Attack graph font size (in points): 11

Figure 3: The setting of Scyther.

(1) $\mathbf{F}_{16} N$ believes $(D$ believes $(N \overset{SK}{\leftrightarrow} D))$.

(x) Since $K_{ND} = h_2(C_1 h_3 \| (e_N e_D P \| T_2) \| T_2)$, we can deduce the formula $\mathbf{F}_{17}$ according to $\mathbf{F}_8$, $\mathbf{A}_1$, and $\mathbf{A}_3$, which is also $\mathbf{G}_4$:

(1) $\mathbf{F}_{17} D$ believes $(N$ believes $(N \overset{SK}{\leftrightarrow} D))$.

(xi) According to $\mathbf{F}_{16}$, $\mathbf{A}_5$, and $\mathbf{R}_3$, we deduce the formula $\mathbf{F}_{18}$, which is also $\mathbf{G}_1$:

(1) $\mathbf{F}_{18} N$ believes $(N \overset{SK}{\leftrightarrow} D)$.

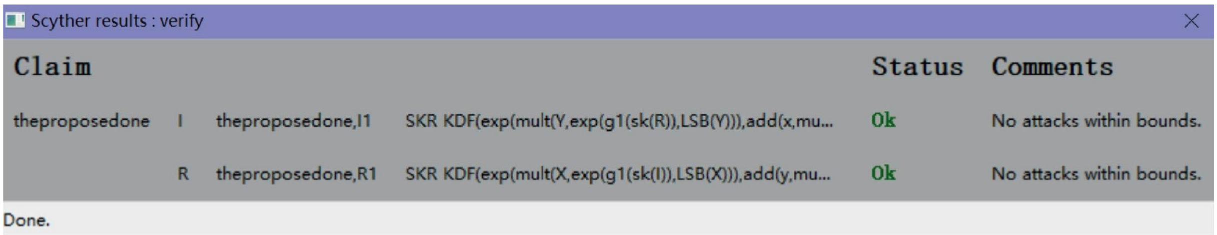
(xii) Similarly, according to $\mathbf{F}_{17}$, $\mathbf{A}_6$, and $\mathbf{R}_3$, we deduce the formula $\mathbf{F}_{19}$, which is also $\mathbf{G}_2$:

(1) $\mathbf{F}_{19} D$ believes $(N \overset{SK}{\leftrightarrow} D)$.

According to $\mathbf{F}_{16}$ to $\mathbf{F}_{19}$, the secure goals $\mathbf{G}_1$ to $\mathbf{G}_4$ of e-SMDAS protocol are achieved. The sensor node SN_i and the dew server DS_j can achieve the mutual authentication and the same session key securely.

8. Performance Analysis

In this section, we present the performance analysis of e-SMDAS protocol, compared with several recent works, namely, SMDAS [10], He et al.'s scheme [27], and Ying et



```
Scyther results : verify

Claim                                                                              Status   Comments

theproposedone   I   theproposedone,I1   SKR KDF(exp(mult(Y,exp(g1(sk(R)),LSB(Y))),add(x,mu...   Ok   No attacks within bounds.

                 R   theproposedone,R1   SKR KDF(exp(mult(X,exp(g1(sk(I)),LSB(X))),add(y,mu...   Ok   No attacks within bounds.

Done.
```

Figure 4: The analysis result by Scyther tool.

Table 4: Comparison of security features with SMDAS protocol.

Security features	Scheme			
	[27]	[28]	SMDAS	e-SMDAS
Mutual authentication	No	Yes	Yes	Yes
Session key agreement	Yes	Yes	Yes	Yes
Replay attack resistance	Yes	No	Yes	Yes
User impersonation attack resistance	No	No	Yes	Yes
User anonymity	No	Yes	No	Yes
Forward security	Yes	Yes	No	Yes

Table 5: Comparison of the communication efficiency.

Scheme	Communication efficiency	
	Computation cost	Communication cost (bits)
SMDAS	$10T_h + T_b + T_{\mathrm{pa}}$	1184
He et al. [27]	$4T_e + 5T_{\mathrm{pa}} + 2T_b + 5T_h$	3296
Ying et al. [28]	$7T_e + 10T_h + 4T_{\mathrm{enc/dec}}$	3840
e-SMDAS	$10T_h + 2T_e$	1024

Note. T_{pa}: point addition in elliptic curve group; T_e: exponentiation operation in cyclic group; T_h: hash function; T_b: bilinear map; T_{eb}: exponentiation operation over bilinear pairing; T_{ma}: modular addition in cyclic group; $T_{\mathrm{enc/dec}}$: encryption/decryption operation.

al.'s scheme [28], from the aspects of security features and computational efficiency.

Table 4 demonstrates the result of security feature comparison with several similar works. According to the work of [29, 30], the comparative result of security features is clear. It is shown in the table that the proposed protocol remedies the flaws of SMDAS protocol. As Table 4 shows, the e-SMDAS protocol can resist replay attack as well as user impersonation attack and satisfy the secure requirements for anonymity and forward security. Generally, our e-SMDAS protocol performs better than the previous one.

Before presenting the analysis, we first denote the notations used in the estimation of the computation efficiency. To be concise, the meanings of T_{pa}, T_e, T_h, T_b, T_{eb}, T_{ma}, and $T_{\mathrm{enc/dec}}$ are time of performing a point addition in elliptic curve group, time of performing an exponentiation operation in cyclic group, time of performing a hash function, time of performing a bilinear map, time of performing an exponentiation over bilinear pairing, time of performing a modular addition in cyclic group, and time of performing an encryption or decryption operation, respectively. Besides, the time of XOR can be negligible. In Table 5, we compare the computation efficiency of the related works with that of ours. For the sensor node in the proposed protocol, the computation cost is $5T_h + T_e$. On the other hand, dew server operates at the cost of $5T_h + T_e$.

To compare the efficiency of communication precisely, we simulate the schemes under the following assumptions. The output length of hash function is 160 bits while that of symmetric encryption tool is 1024 bits. The size of timestamp is 32 bits, while the output of elliptic curve is 160 bit. The comparative result is demonstrated in both Table 5 and Figure 5, in which e-SMDAS appears to be more efficient.

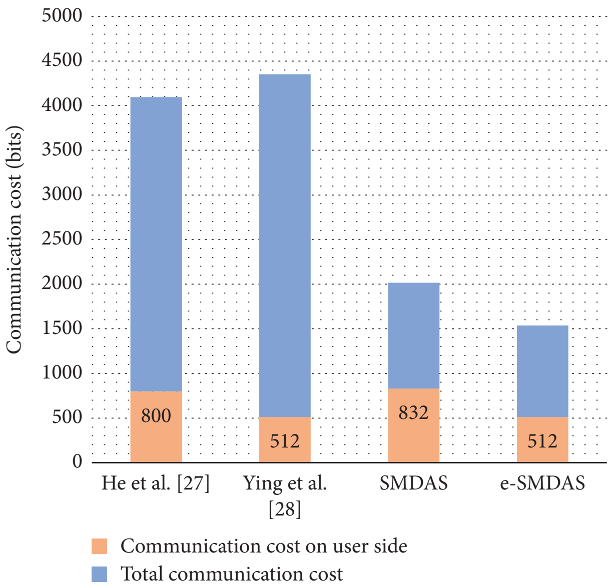


Figure 5: The comparison of the communication cost.

9. Conclusion

The dew-assisted IoT framework is an essential approach developing rapidly in the communication systems, which can provide high efficiency and low latency. In this paper, we first analyze SMDAS protocol showing that this protocol lacks forward security and user anonymity. Then, based on ECCDH problem, we propose an enhancement of the original one, called e-SMDAS protocol. We present the formal security proof of the proposed protocol. Moreover, the test of security by the usage of formalization tool Scyther and BAN logic shows that e-SMDAS can satisfy more security features than the former protocol. Furthermore, the performance analysis is presented at last showing that the enhancement does not affect the running time and computation efficiency.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant no. 61872449).

References

[1] B. Flavio, M. Rodolfo, Z. Jiang, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the conference: The First Edition of the MCC Workshop on Mobile Cloud Computing*, pp. 13–16, dblp, New York, NY USA, 17 August 2012.

[2] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, "A blockchain-based machine learning framework for edge services in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, 2022.

[3] Y. Wang, "Cloud-dew architecture," *International Journal of Cloud Computing*, vol. 4, no. 3, pp. 199–210, 2015.

[4] Y. Wang, "Definition and categorization of dew computing," *Open Journal of Cloud Computing*, vol. 3, no. 1, pp. 1–7, 2016.

[5] K. Hameed, S. Garg, M. B. Amin, and B. Kang, "A formally verified blockchain-based decentralised authentication scheme for the internet of things," *The Journal of Supercomputing*, vol. 77, no. 12, pp. 14461–14501, 2021.

[6] Y. Liu, J. Zhang, and J. Zhan, "Privacy protection for fog computing and the internet of things data based on blockchain," *Cluster Computing*, vol. 24, no. 2, pp. 1331–1345, 2021.

[7] S. Shukla, S. Thakur, S. Hussain, J. G. Breslin, and S. M. Jameel, "Identification and authentication in healthcare Internet-of-Things using integrated fog computing based blockchain model," *Internet of things*, vol. 15, Article ID 100422, 2021.

[8] S. Singh and V. K. Chaurasiya, "Mutual authentication scheme of IoT devices in fog computing environment," *Cluster Computing*, vol. 24, no. 3, pp. 1643–1657, 2021.

[9] S. Amanlou, M. K. Hasan, and K. A. A. Bakar, "Lightweight and secure authentication scheme for IoT network based on publish-subscribe fog computing model," *Computer Networks*, vol. 199, no. 2021, Article ID 108465, 2021.

[10] R. Saurabh, S. O. Mohammad, M. Dheerendra, A. Mishra, and Y. S Rao, "Efficient design of an authenticated key agreement protocol for dew-assisted IoT systems," *The Journal of Supercomputing*, vol. 78, pp. 3696–3714, 2022.

[11] X. Li, J. Niu, S. Kumari, F. Wu, A. K. Sangaiah, and K.-K. R. Choo, "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments," *Journal of Network and Computer Applications*, vol. 103, pp. 194–204, 2018.

[12] R. Amin and G. P. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Networks*, vol. 36, pp. 58–80, 2016.

[13] A. K. Awasthi and K. Srivastava, "A biometric authentication scheme for telecare medicine information systems with nonce," *Journal of Medical Systems*, vol. 37, no. 5, p. 9964, 2013.

[14] Z. Tan, "A user anonymity preserving three-factor authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 38, no. 3, p. 16, 2014.

[15] H. Arshad and M. Nikooghadam, "Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 38, no. 12, p. 136, 2014.

[16] Y. Lu, L. Li, H. Peng, and Y. Yang, "An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem," *Journal of Medical Systems*, vol. 39, no. 3, p. 32, 2015.

[17] L. Han, Q. Xie, and W. Liu, "An improved biometric based authentication scheme with user anonymity using elliptic curve cryptosystem," *International Journal on Network Security*, vol. 19, no. 3, pp. 469–478, 2017.

[18] V. P. Gaikwad, J. V. Tembhurne, C. Meshram, and C.-C. Lee, "Provably secure lightweight client authentication scheme with anonymity for TMIS using chaotic hash function," *The Journal of Supercomputing*, vol. 77, no. 8, pp. 8281–8304, 2021.

[19] M. Masud, G. S. Gaba, S. Alqahtani et al., "A lightweight and robust secure key establishment protocol for internet of medical things in COVID-19 patients care," *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 15694–15703, 2021.

[20] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2021.

[21] Y. Tian, Z. Zhang, J. Xiong, L. Chen, J. Ma, and C. Peng, "Achieving graph clustering privacy preservation based on structure entropy in social IoT," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2761–2777, 2022.

[22] S. A. Chaudhry, M. S. Farash, H. Naqvi, S. Kumari, and M. K. Khan, "An enhanced privacy preserving remote user authentication scheme with provable security," *Security and Communication Networks*, vol. 8, no. 18, pp. 3782–3795, 2015.

[23] N. Ravanbakhsh, M. Mohammadi, and M. Nikooghadam, "Perfect forward secrecy in VoIP networks through design a lightweight and secure authenticated communication scheme," *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 11129–11153, 2019.

[24] M. Nikooghadam and H. Amintoosi, "Perfect forward secrecy via an ECC-based authentication scheme for SIP in VoIP," *The Journal of Supercomputing*, vol. 76, no. 4, pp. 3086–3104, 2020.

[25] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Proceedings of the conference: ProvSec2007 - Provable Security*, pp. 1–16, Springer, Wollongong, Australia, 1 November 2007.

[26] H. Krawczyk, "HMQV: a high-performance secure Diffie-Hellman protocol," *Advances in Cryptology - CRYPTO 2005*, vol. 3621, pp. 546–566, 2005.

[27] D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen, "Efficient privacy-aware authentication scheme for mobile cloud

computing services," *IEEE Systems Journal*, vol. 12, pp. 1621–1631, 2016.

[28] B. Ying and A. Nayak, "Efficient authentication protocol for secure vehicle communications," in *Proceedings of the conference: 2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, pp. 1–5, IEEE, Seoul, Korea (South), 18 May 2014.

[29] C. Chen, B. Xiang, Y. Liu, and K. Wang, "A secure authentication protocol for internet of vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

[30] S. Nagaraju, S. K. V. Jayakumar, and C. S. Priya, "An effective mutual authentication scheme for provisioning reliable cloud computing services," in *Proceedings of the conference: 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 314–321, IEEE, Greater Noida, India, 19 February 2021.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 6800216, 9 pages
https://doi.org/10.1155/2022/6800216

Research Article

Improved Public Auditing System of Cloud Storage Based on BLS Signature

Ruifeng Li, Haibin Yang, Xu An Wang, Zhengge Yi, and Ke Niu

Chinese People's Armed Police Force Engineering University, Xi'an, China

Correspondence should be addressed to Xu An Wang; wangxazjd@163.com and Ke Niu; niuke@163.com

Received 19 January 2022; Accepted 5 May 2022; Published 29 May 2022

Academic Editor: Qi Li

Cloud storage and cloud computing technologies have developed rapidly for a long time, and many users outsource the storage burden of their data to the cloud to obtain more convenient cloud storage services. Allowing users to audit the private data's integrity has become an additional basic function of the cloud server when providing services. In 2021, based on the BLS signature and automatic blocker protocol, Jalil et al. proposed a secure and efficient cloud data auditing protocol. The protocol can realize public audit, batch audit, data update, and protect data privacy. Moreover, the automatic blocker protocol is used to realize the identity authentication of the auditor. The protocol is relatively novel, innovative, and has a larger use space. However, we found that their scheme had security problems. If the cloud server has thoughts of malicious attack, he can forge the proof that he holds users' data with stored labels and pass the audit. Referring to the original protocol and being inspired by them, we propose an improved audit protocol. The improved protocol solves the security problem and is more effective.

1. Introduction

Recently, advanced and innovative technologies represented by cloud computing and cloud storage have become increasingly mature. Cloud storage and cloud computing technologies have the characteristics of convenience, economy, and high scalability. Users can store the generated data in the platform and control their data remotely without purchasing and using local storage devices. Users are increasingly inclined to use cloud storage services to manipulate data more quickly and easily.

Cloud server providers centrally hold massive amounts of users' data, which are easily targeted by malicious attackers, and dishonest cloud server providers will deliberately delete users' data or conceal data security incidents from users for reasons such as reducing their own storage burden or maintaining their reputation. In the application of cloud storage technology, users cannot absolutely manipulate the data, and the integrity of the users' data is threatened. Verifying the integrity of cloud data is a hot topic of current research now.

1.1. Organization. We organize our paper as follows: in Section 1, we introduce the research background and related work. In Section 2, we describe the system model of cloud storage audit protocol. In Section 3, we review Jalil et al.'s public audit protocol. In Section 4, we give our attack on the original protocol and show that it is not efficient. In Section 5, we introduce our improved secure auditing protocol. In Section 6, we analyze the security of the improved protocol and compare the audit efficiency of the improved protocol with the original protocol. Finally, in Section 7, we make the conclusion of our work.

1.2. Related Work. Scholars have proposed many cloud storage data integrity audit protocols with different functions to meet the different needs of users in different application scenarios more effectively. In 2004, based on the RSA signature, Dewarte et al. [1] designed a protocol to audit remote files. However, the exponential calculation on all data blocks in the file will be performed on the user side, which will result in expensive computational overhead. In 2007, Ateniese et al. [2] designed a verification scheme suitable for a cloud storage environment called "provable data possession (PDP)." The protocol uses an RSA-based homomorphic linear authenticator and random sampling technology, and users only download the partial file to be able to verify the integrity. Then, Juels and Kaliski [3] designed another "proof of retrievability(PoR)" scheme suitable for a cloud storage environment, which implements data integrity detection by inserting special data blocks (generally called "sentinels") into the data file.

In the actual application of cloud storage, users may need to perform various modifications and update operations on the data. Therefore, researchers have proposed audit protocols that support dynamic data updates. In 2008, Ateniese et al. [4] first proposed an audit protocol that can achieve dynamic data update with a symmetric encryption method. However, this audit protocol has the shortcoming of limited audit numbers and does not support public data audit. In 2012, Zhu et al. [5] constructed an audit protocol that supports dynamic data update with an index hash table (IHT) based on zero-knowledge proof. In 2015, Erway et al. [6] designed an audit protocol based on the sorted authentication skip list. The protocol supports a complete data dynamic update. In 2016, Jin et al. [7] introduced an index switcher to propose an audit protocol that not only provides fair arbitration but also supports dynamic data updates. In 2017, Shen et al. [8] used a two-way linked list of location arrays to implement the audit of the data. The protocol uses global and block-free sampling verification methods, which can also reduce computing and communication costs. In 2019, Guo et al. [9] designed a verification protocol that supports task outsourcing and supports dynamic data updates. It provides a log audit mechanism to enable users to detect misconduct by dishonest auditors. However, the solution has security loopholes. After multiple audits of data blocks with the same index, theoretically, data labels can be forged by solving linear independent equations. In 2020, the cloud audit scheme suitable for IoT [10] designed by Hou et al. [11] uses a chameleon authentication tree to save the computational overhead during the dynamic data update process and supports batch audit.

If users undertake the periodic audit work, it will generate a large computational overhead and consume a lot of resources [12]. In practical application scenarios, it is important to protect the privacy of user's data [13]. Scholars introduce a third-party auditor (TPA) to help users regularly check the integrity of the data stored on the cloud server. However, when users outsource the audit task, TPA will obtain data content during the implementation of audit tasks [14]. In 2013, Wang et al. [15] designed a public verification scheme, and the scheme supports a privacy protection function based on random masking technology and batch audit function based on the homomorphic linear authenticator. The protocol ensures that TPA cannot obtain the user's real data during the data integrity audit process. In 2014, Worku et al. [16] used random masking technology to propose an efficient public audit protocol with data privacy protection function. Wang et al. [17] designed a shared data audit protocol, which uses the ring signature technology and can protect the users' identity privacy. In 2015, Xiong et al. [18] used an ID-based encryption algorithm to design a privacy protection protocol, and the protocol uses distributed hash table network to protect sensitive data. In 2016, Li et al. [19] used online/offline signatures to design a lightweight public audit protocol with data privacy protection function.

Traditional cloud audit protocols are mostly based on the design of PKI cryptosystem, which brings complicated certificate management issues. In 2013, the first public identity-based audit scheme was designed by Zhao et al. [20]. The protocol minimizes the information carried in the verification process and the information obtained or stored by TPA, which simplifies key management and reduces communication and calculation overhead. In 2014, Wang et al. [21] proposed an ID-based data audit scheme, which formally defines the ID-based remote file verification model. The protocol gave the first security proof of the identity-based audit protocol based on CDH problem's difficulty. In 2016, Wang et al. [22] designed an agent-oriented ID-based remote data audit protocol. According to user's authorization, the protocol can realize three modes of private audit, entrusted audit, and public audit. In the same year, Yu et al. [23] used zero-knowledge proof to propose an ID-based cloud audit protocol that supports the privacy protection of users' data. The protocol regulates the identity-based audit protocol and its security model and can realize zero-knowledge privacy protection for TPA. In 2019, as the solution to the complex key management problem in cloud data integrity verification, Li et al. [24] used fuzzy identity to design an audit protocol. Xue et al. [25] designed an ID-based audit protocol using blockchain to construct random challenge messages. In their protocol, TPA cannot forge audit results to deceive users [26]. Peng et al. [27] designed a new ID-based data ownership verification protocol using compressed authentication arrays, which can simultaneously and efficiently support batch verification for multiple users in terms of computing and communication. Rabaninejad et al. [28] used the online/offline signature to design an ID-based PDP, and the protocol is implemented to support privacy protection, batch audit, and full dynamic data update [26].

However, the key escrow problem exists in ID-based cloud audit protocols, so many cloud audit protocols based on certificateless signature have been proposed. In the certificateless signature system, the user and the key generation center (KGC) cooperate to produce the private key for the user, which can avoid the strong dependence of the system security on the KGC security [29]. In 2013, Wang et al. [30] designed a certificateless cloud audit protocol, but He et al. [31] later pointed out the security problem. In 2015, Zhang et al. [32] designed the certificateless cloud data verification protocol that can resist malicious auditors. In 2017, Kang et al. [33] applied the certificateless cloud audit protocol to wireless body area networks. The proposed protocol can resist malicious auditors and protect data content. The certificateless cloud audit protocol proposed by He et al. [34] can protect users' privacy, but it has also been pointed out that there are security problems. He et al. [35]

applied the certificateless data audit protocol to the data management system of the smart grid, reducing the computational overhead. In 2018, Yang et al. [36] designed a certificateless cloud audit scheme for group user file sharing, which supports the protection of data content and users' identity privacy. In 2019, Wu et al. [37] defined the security model of the certificateless cloud audit protocol with privacy protection. The proposed protocol supports the protection of multiuser group identity privacy. In 2020, Huang et al. [38] designed a certificateless data verification protocol supporting the batch audit function, which realized efficient key update based on the Chinese remainder theorem.

1.3. Our Contribution. Recently, Jalil et al. [39] proposed an effective cloud data public audit protocol based on BLS signature to realize public audit and protect file content privacy. The protocol implements batch audit and dynamic update. Their scheme also uses automatic blocker protocol (ABP) to prevent unauthorized TPA from participating in the audit work, which is highly innovative, and ABP is essentially an access control facility [40], which can detect threats from auditors [41]. However, we found that their protocol has security issues. Even if the cloud server does not hold the stored data, he can mathematically prove that he holds the user's data. Then, we propose an improved and secure protocol with high security. The analysis shows the safety and effectiveness of our improved program in actual environments.

2. System Model

To facilitate understanding, we define and explain the various symbols and variables that appear in the original scheme and the improved scheme in Table 1.

Table 1: Notations.

Notations	Descriptions
$(b_1, ..., b_n)$	Unencrypted n data blocks
$(e_1, ..., e_n)$	Encrypted n data blocks
G	Multiplicative cyclic group
E	Bilinear mapping
H	Secure hash function $H(\cdot)$: $\{0, 1\}^* \longrightarrow Z_q^*$
Z_q^*	Prime field
g	Generator of G
λ	System initialization parameter
F	User's data file
k_s	User's secret key
m_i	Name of data block e_i
k_p	Public key of user
Q	Challenged subset of $(1, n)$
S_i	Authentication label for e_i
S	Collection of S_i
c	Number of challenged data blocks
a_i, r, p_i	Random values
V, μ, μ', R	Intermediate parameter
S_U	Collection of authentication labels of multiusers
$\|Z_q^*\|$	The size of an element of Z_q^*
$\|S\|$	The size of a label
$\|E_G\|$	The computational cost of a power on G
$\|M_G\|$	The computational cost of a multiplication on G
$\|A_G\|$	The computational cost of an add on G
$\|E\|$	The computational cost of a bilinear mapping
$\|H\|$	The computational cost of a hash

The existing cloud audit systems generally include three interactive entities: cloud server provider (CSP) provides users with data storage services to obtain remuneration. CSPs are incredible. They may delete cloud data for profit or steal users' data privacy. Users: users are the owners of the data, and they upload files to the cloud to save their own storage cost. Third-party auditor (TPA) is not an entirely believable auditor entrusted by users, and on the one hand, TPA performs the audit task faithfully, and on the other hand, TPA attempts to decipher the content of the user's data with curiosity.

The interaction process of all entities: the user preprocesses the data to be stored and uploads it to CSP. When the data integrity needs to be verified, TPA generates a challenge with relevant parameters and sends them to CSP. Based on the challenge parameters, CSP uses cloud data to generate the proof that he holds the user data in full and sends the proof to TPA. TPA uses the proof to audit the data's integrity and sends the result to the user.

3. Review of Jalil et al.'s Protocol

There are three entities involved in Jalil et al.'s scheme. Jalil et al. used the BLS signature to achieve public audit and protect data content privacy. The program also supported batch audit and dynamic update. In addition, the proposed system enhanced the level of security authentication through an ABP to protect the system from unauthorized TPA. In particular, their scheme contains the following algorithm.

3.1. DataProtection Protocol. To protect data privacy, data file blocks need to be encrypted first. The user divides the data file F into n data blocks $(b_1, \cdots, b_n)$ and then uses the AES encryption algorithm to encrypt the data blocks and obtain the encrypted data blocks $(e_1, \cdots, e_n)$.

3.2. Setup Protocol. The user takes the security parameter $\lambda \in Z_q^*$ as input, for each data block e_i, outputs the corresponding private key $k_{s_i} \in Z_q^*$, and calculates the corresponding public key $k_{p_i} = g^{k_{s_i}} \in G$.

3.3. SignatureGen Protocol. For each data block e_i, the user generates a random value $a_i \in Z_q^*$ and calculates the corresponding label S_i:

$$S_i = \left(H(m_i) \cdot g^{a_i}\right)^{k_{s_i}}, \tag{1}$$

where m_i is the name of relevant blocks e_i and H is SHA256 hash function, which defines intermediate parameters $V_i = g^{a_i} \in G$.

Then, the user uploads V_i and m_i to the auditor and uploads e_i and S_i with pk_i to cloud for $i \in [1, n]$ and deletes the local data.

3.4. ChallGen Protocol. When the user needs to verify the integrity of cloud data, he sends an audit request to the TPA. TPA first randomly selects c elements to form a subset Q of $[1, n]$. For all $i \in Q$, TPA selects a random $p_i \in Z_q^*$ and sends all i and p_i to CSP.

3.5. Response Protocol. When CSP receives an audit challenge from TPA, he first asks the user whether the user has issued an audit request, thereby confirming the authenticity of the challenge from the TPA. After receiving user's affirmative reply, CSP confirms that the challenge is true and performs the next step. This process is implemented through the ABP. CSP uses the following equation to calculate the aggregate tag and sends the evidence S to the auditor:

$$S = \prod_{i=1}^{c} S_i. \tag{2}$$

3.6. CheckProof Protocol. When the TPA receives the corresponding evidence generated by the CSP for the challenge, he calculates the following equation to verify the integrity of the data:

$$E(S, g) = \prod_{i=1}^{c} E\left(H(m_i) \cdot V_i, k_{p_i}\right). \tag{3}$$

If equation (3) is true, he shows that the CSP has faithfully performed the service and ensured the integrity of the cloud data.

3.7. BatchAuditing Protocol. Each user divides the original file into n data blocks, then uses different encryption keys to encrypt the respective data blocks, generates private and public keys for different data blocks, and uses equation (1) to generate data tags. All users send $(e_i, S_i, pk_i, i\ d)$ for $i \in [1, n]$ to the cloud and upload metadata $(m_i, V_i, i\ d)$ to TPA, where $i\ d$ represents the user's identifier. When the data integrity needs to be verified, TPA randomly selects c data block indexes to be challenged and sends them to CSP. After CSP receives the challenge and confirms the authenticity of the challenge, based on the label set S_j of each user, the aggregate label S_U is calculated for all challenged data blocks:

$$S_U = \prod_{j=1}^{u} S_i, \quad \text{where}\,[1 \le j \le u]. \tag{4}$$

CSP generates evidence $(S_U, k_{p_{ij}})_{(1 \le i \le c, 1 \le j \le u)}$ and sends it to TPA. After receiving the evidence, the TPA verifies whether the following equation holds:

$$E(S_U, g) = \prod_{j=1}^{u} \left\{ E\left(\prod_{i=1}^{c} H(m_i)_j \cdot V_{ij}, k_{p_{ij}} \right) \right\}. \tag{5}$$

If equation (5) is true, it means that the integrity of the data has not been damaged.

4. Our Attack

In the audit protocol of Jalil et al.'s scheme, the correctness of the audit cannot be achieved. Even if the user's data held by the CSP are incomplete, CSP can pass the audit. In the SignatureGen protocol, the user calculates the signatures $(\{S_i\}_{[1 \le i \le n]})$ as equation (1). In equation (1), the calculation process of $(\{S_i\}_{[1 \le i \le n]})$ is determined by the private key value sk_i and the name of the data block m_i. However, $(\{S_i\}_{[1 \le i \le n]})$ are not signatures of the content e_i. In response protocol, CSP only uses equation (2) to calculate the aggregation signature, but he does not calculate the aggregation of the data content. The integrity proof generated by the CSP has nothing to do with the content of the data block. The CSP can use the stored signatures $(\{S_i\}_{[1 \le i \le n]})$ to generate the integrity evidence and pass the audit, so he can store the name m_i locally instead of the content e_i. In addition, in the original scheme, the number of public keys and private keys required is extremely large, which is proportional to n. Both in terms of certificate management and storage overhead of three entities, it is more complicated and cumbersome. In the CheckProof protocol, c bilinear mappings are used. The cost of calculation is also relatively high. In this section, we will show that CSP can generate an integrity proof that passes the audit from TPA without the store data block e_i.

The relevant data stored by CSP include the following:

$$\begin{gathered} e_1, e_2, \ldots, e_n, \\ S_1, S_2, \ldots, S_n, \\ m_1, m_2, \ldots, m_n, \\ k_{p_1}, k_{p_2}, \ldots, k_{p_n}. \end{gathered} \tag{6}$$

User needs to store the following:

$$\begin{gathered} k_{s_1}, k_{s_2}, \ldots, k_{s_n}, \\ k_{p_1}, k_{p_2}, \ldots, k_{p_n}. \end{gathered} \tag{7}$$

The data stored by TPA include the following:

$$\begin{gathered} m_1, m_2, \ldots, m_n, \\ V_1, V_2, \ldots, V_n, \\ k_{p_1}, k_{p_2}, \ldots, k_{p_n}. \end{gathered} \tag{8}$$

We can see that the storage costs of the three entities are proportional to n, and the storage costs are relatively large, which violates the original intention of cloud storage. In addition, CSP and TPA need to store n public keys, users need to store the same number of private and public keys as the number of e_i requiring a lot of certificates, and certificate management is more complicated.

In the response protocol of Julil et al.'s protocol, CSP only generates the aggregation of signatures. CSP stores S_i, so regardless of whether CSP stores data, aggregate tags S can be generated according to equation (2). As long as the stored signatures are correct, the correct data audit proof can be generated and verified by the CSP.

In the CheckProof stage, after the auditor accepts the proof, he needs to verify whether equation (3) is true or not and calculates c bilinear mappings. The bilinear mapping is computationally expensive and reduces the audit efficiency.

5. Improvements to the Secure Auditing Protocol

Based on the above analysis, the original protocol is improved here to enhance security and efficiency. The difference comparison between the original scheme and the improved scheme is shown in Figure 1.

5.1. DataProtection Protocol. The user encrypts n data blocks $(b_1, \cdots, b_n)$ divided from the data file F using the AES encryption algorithm and obtains the encrypted data blocks $(e_1, \cdots, e_n)$, which can protect data privacy.

5.2. Setup Protocol. CSP inputs security parameters λ and outputs public parameters $\{G, g, E, H\}$. Among them, G is a multiplicative cyclic group, g is the generator of G, E is the bilinear mapping, and H is the hash function. The user randomly generates $k_s \in Z_q^*$ and calculates $k_p = g^{k_s} \in G$.

5.3. SignatureGen Protocol. For each data block e_i, the user calculates the corresponding label S_i:

$$S_i = \left(H(i) \cdot g^{e_i}\right)^{k_s}, \tag{9}$$

The tag $(\{S_i\}_{[1 \le i \le n]})$ is calculated by the secret key k_s, data block e_i, and data block index i. Then, the user deletes the local data and tags after uploading them to the cloud.

5.4. ChallGen Protocol. To verify whether the data are complete, the user sends a message to TPA requesting an audit first. TPA randomly selects c elements from $(1, n)$ to form a subset Q, and then, he randomly selects $p_i \in Z_q^*$ for all $i \in$ Q. Finally, all i and p_i are sent to CSP.

5.5. Response Protocol. When CSP receives an audit challenge from TPA, he first ensures the authenticity of the challenge by querying the user. When the user's authenticity is confirmed, the CSP will accept the challenge. This process is implemented through the ABP. CSP randomly generates $r \in Z_q^*$ and uses the following equations to calculate the proof:

$$R = k_p^r, \tag{10}$$

$$S = \prod_{i=1}^{c} S_i^{p_i}, \tag{11}$$

$$\mu\prime = \sum_{i=1}^{c} p_i e_i, \tag{12}$$

$$\mu = \mu' + r, \tag{13}$$

and then sends the proof $\{R, S, \mu\}$ to the auditor.

5.6. CheckProof Protocol. When the CSP sends the evidence to the TPA, TPA verifies the authenticity of equation (14):

$$e(S \cdot R, g) \overset{?}{=} e\left(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\mu}, k_p\right). \tag{14}$$

If equation (14) is true, the data are completed and not corrupted. The process of proving the truth of equation (14) is as follows:

$$\begin{aligned} e(S \cdot R, g) &= e\left(\prod_{i=1}^{c} S_i^{p_i} \cdot \left(g^{k_s}\right)^r, g\right) = e\left(\prod_{i=1}^{c} ((H(i) \cdot g^{e_i}))^{p_i} \cdot g^r, g^{k_s}\right) = e\left(\prod_{i=1}^{c} H(i)^{p_i} \cdot \prod_{i=1}^{c} g^{e_i p_i} \cdot g^r, k_p\right) \\ &= e\left(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\sum_{i=1}^{c} e_i p_i + r}, k_p\right) = e\left(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\mu}, k_p\right). \end{aligned} \tag{15}$$

5.7. BatchAuditing Protocol. u users use different encryption keys to encrypt the data blocks belonging to themselves among the n data blocks divided from the original file, generate private keys $k_{s_j (1 \le j \le u)}$ and public keys $k_{p_j (1 \le j \le u)}$, and then use equation (9) to generate data tags. All users delete the local data after the task of transferring (e_i, S_i) to the cloud server is completed. To prove the completeness of the data, the TPA randomly selects k data block indexes to be challenged, sending the indexes and corresponding random values $p_{i(1 \le i \le c)}$ to the CSP. After the CSP receives and confirms the authenticity of the content, TPA randomly generates $r_j \in Z_q^*$ for each user and calculates:

$$R = \prod_{j=1}^{u} k_{p_j}^{r_j} = \prod_{j=1}^{u} g^{k_{s_j} \cdot r_j}, \tag{16}$$

$$\mu_j' = \sum_{i=1}^{c} p_i \cdot e_{ij}, \tag{17}$$

$$\mu_j = \mu_j' + r_j. \tag{18}$$

Based on the set $S_{j(1 \le j \le u)}$ of each user, the aggregate tag S_U is calculated for all challenged data blocks:

$$S_U = \prod_{j=1}^{u} \prod_{i=1}^{c} S_{ij}^{p_i}. \tag{19}$$

CSP generates evidence $P = (S_U, k_{p_{ij}})_{(1 \le i \le c, 1 \le j \le u)}$ and sends it to TPA as a basis for the verification. Upon receipt, TPA indicates whether the cloud data are completed by verifying the following equation:

$$E(S_U \cdot R, g) \overset{?}{=} \prod_{j=1}^{u} E\left(\left(\prod_{i=1}^{c} H(i)^{q_i} \cdot g^{\mu_j}, k_{p_j}\right)\right). \tag{20}$$

If equation (20) holds, it proves that data integrity has not been compromised. The proof of the correctness of (15) is as follows:

$$\begin{aligned} & E(S_U \cdot R, g) \\ & = E\left(\prod_{j=1}^{u} \prod_{i=1}^{c} S_{ij}^{p_i} \cdot \prod_{j=1}^{u} g^{k_{s_j} \cdot r_j}, g\right) \\ & = \prod_{j=1}^{u} E\left(\prod_{i=1}^{c} \left(H(i) \cdot g^{e_{ij}}\right)^{k_{s_j} p_i} \cdot g^{k_{s_j} \cdot r_j}, g\right) \\ & = \prod_{j=1}^{u} E\left(\prod_{i=1}^{c} \left(H(i) \cdot g^{e_{ij}}\right)^{p_i} \cdot g^{r_j}, g^{k_{s_j}}\right) \\ & = \prod_{j=1}^{u} E\left(\prod_{i=1}^{c} H(i)^{p_i} g^{\sum_{i=1}^{c} e_{ij} p_i + r_j}, k_{p_j}\right) \\ & = \prod_{j=1}^{u} E\left(\prod_{i=1}^{c} H(i)^{p_i} g^{\mu}, k_{p_j}\right). \end{aligned} \tag{21}$$

6. Analysis of the Improved Protocol

The security of the improved protocol is first analyzed and explained here, including preventing forgery attack from CSP and attack from TPA to steal data content privacy. Then, the storage and computation overhead of the improved protocol are analyzed in comparison with the original protocol, to prove that the improved protocol is safe and efficient.

6.1. Security Analysis.

(1) Anti-Forgery Attack: if in the cloud, the CSP generates a forged audit certificate $\widetilde{\mu}$ and the stored user data are corrupted or tampered with, then it means that the group can compute the discrete logarithm problem with probability $1 - 1/q$ (q is a large prime number).

A forged data possession proof $\widetilde{\mu} = \sum_{i=1}^{c} p_i \widetilde{e}_i + r$ will be generated by the CSP in the case of incorrect data, and we define the following:

$$\begin{aligned} \Delta\mu &= \widetilde{\mu} - \mu \\ &= \widetilde{\mu}' - \mu' \\ &= \sum_{i=1}^{c} p_i \widetilde{e}_i - \sum_{i=1}^{c} p_i e_i \\ &= \sum_{i=1}^{c} p_i \Delta e_i. \end{aligned} \tag{22}$$

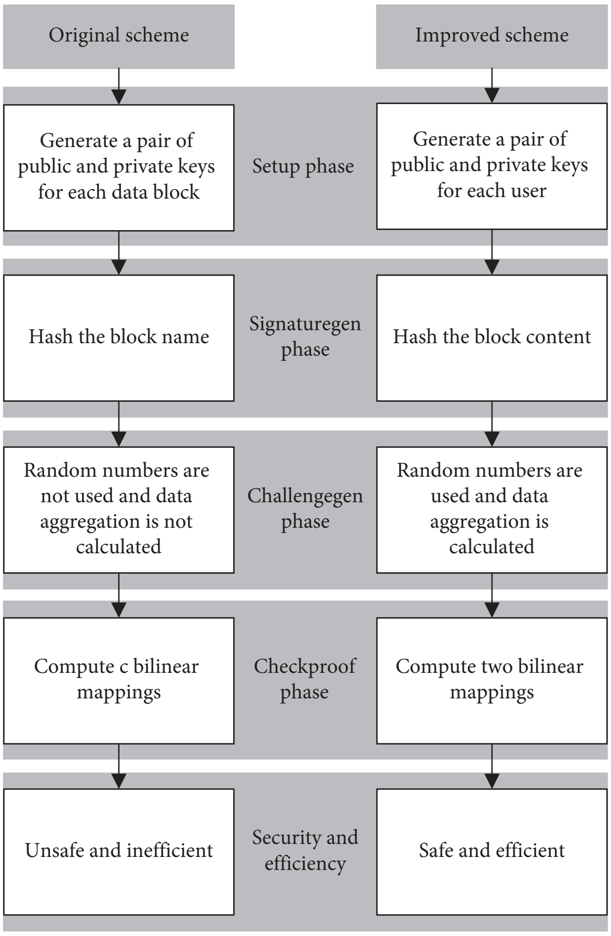

Original scheme
Improved scheme
Generate a pair of public and private keys for each data block
Setup phase
Generate a pair of public and private keys for each user
Hash the block name
Signaturegen phase
Hash the block content
Random numbers are not used and data aggregation is not calculated
Challengegen phase
Random numbers are used and data aggregation is calculated
Compute c bilinear mappings
Checkproof phase
Compute two bilinear mappings
Unsafe and inefficient
Security and efficiency
Safe and efficient

Figure 1: Difference between two schemes.

Because $\widetilde{\mu}$ is the forged evidence, there must be a difference between $\widetilde{\mu}$ and μ, and there is at least one $\Delta e_i \ne 0$. Assuming that CSP's forged proof of data possession $\widetilde{\mu}$ can pass TPA's audit, therefore

$$E(S \cdot R, g) = E\left(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\widetilde{\mu}}, k_p\right). \tag{23}$$

The correct proof can pass the TPA audit; therefore,

$$E(S \cdot R, g) = E\left(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\mu}, k_p\right). \tag{24}$$

From equations (23) and (24), we get $E(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\mu}, k_p) = E(\prod_{i=1}^{c} H(i)^{p_i} \cdot g^{\widetilde{\mu}}, k_p)$, so $g^{\mu'} = g^{\widetilde{\mu}'}$ and $g^{\Delta\mu} = 1$. Since G is a cyclic group, so $\forall a, b \in G$, $\exists x \in Z_q^*$ makes $b = a^x$.

Given a and b, then g can be written as $g = a^{y_1} \cdot b^{y_2} \in G$, and $y_1, y_2 \in Z_q^*$, and therefore,

$$1 = g^{\Delta\mu} = \left(a^{y_1} b^{y_2}\right)^{\Delta\mu} = a^{y_1 \Delta\mu} b^{y_2 \Delta\mu}. \tag{25}$$

simplified further to get $b = a^{-y_1 \Delta\mu / y_2 \Delta\mu}$.

To make equation (25) not true, only if the denominator $y_2 = 0$, then equation (25) is meaningless,

Table 2: Storage cost comparison.

	CSP	TPA	USER
Original protocol	$n\|Z_q^*\| + n\|S\|$	$3n\|Z_q^*\|$	$2n\|Z_q^*\|$
Improved protocol	$n\|Z_q^*\| + n\|S\|$	$\|Z_q^*\|$	$2\|Z_q^*\|$

Table 3: Calculation cost comparison.

	CSP	TPA	USER
Original protocol	$c\|M_G\|$	$2c\|M_G\| + c(E) + c(H)$	$4n\|E_G\| + n\|H\| + n\|M_G\|$
Improved protocol	$c\|M_G\| + c\|E_G\|$	$c\|M_G\| + 2\|E\| + c\|H\| + c\|E_G\|$	$n\|E_G\| + n\|H\| + n\|M_G\|$

and $y_1, y_2 \in Z_q^*$, so $P[y_2 = 0] = 1/q$, and the probability that equation (25) is true is $1 - 1/q$.

It is concluded that if the CSP can successfully forge the data block, then he can calculate the discrete logarithm problem and the probability is $1 - 1/q$ but obviously the discrete logarithm problem is a difficult problem, so the CSP cannot forge the fake data block that has passed the audit.

(2) Privacy Protection: first, an authentication protocol (ABP) is used to prevent unauthorized adversaries from entering the system.

Then, in the DataProtection protocol, the user's original data $(b_1, \cdots, b_n)$ are encrypted by AES to obtain $(e_1, \cdots, e_n)$. The data uploaded to the cloud are encrypted data. The CSP does not hold the encryption and decryption keys of the AES encryption algorithm, so it is impossible to know the real data content of the user, avoiding the leakage of data privacy.

Finally, for TPA, the improved protocol uses random masking technology to realize data protection. Assuming that TPA is curious about the challenged data blocks' content $(e_1, \cdots, e_c)$ and audits c data blocks $t\,(t \geq 1)$ times, q_{ji} is represented as random parameters during the j – th time audit on the i – th data block, and then, the set of random numbers is $Q = \{p_{ij}\}_{1 \leq i \leq c, 1 \leq j \leq t}$. An evidence set consisting of t pieces is proofs $= \{(\mu_j, S_j, R_j)\}_{1 \leq j \leq t}$. TPA can obtain the following equations:

$$\begin{cases} p_{11}e_1 + p_{12}e_2 + \cdots + p_{1c}e_c + r_1 = \mu_1 \\ p_{21}e_1 + p_{22}e_2 + \cdots + p_{2c}e_c + r_2 = \mu_2 \\ \vdots \\ p_{t1}e_1 + p_{t2}e_2 + \cdots + p_{tc}e_c + r_t = \mu_t. \end{cases} \tag{26}$$

In the above equations, TPA knows $\{p_{ij}\}_{1 \leq i \leq c, 1 \leq j \leq t}$ and $\{\mu_j\}_{1 \leq j \leq t}$, but he does not know $\{e_i\}_{1 \leq i \leq c}$ and $\{r_j\}_{1 \leq j \leq t}$. There are $c + t$ unknown numbers in equation (26), no matter how many times TPA audits the same data blocks; that is, no matter what the value t is, it will always be less than $c + t$, and TPA cannot solve equation (26) and cannot know the content of the data blocks $(e_1, \cdots, e_c)$ and $(b_1, \cdots, b_c)$.

6.2. Efficiency Analysis. In the original protocol, the user needs to generate the corresponding public keys k_{p_i} and private keys k_{s_i} for $e_{i\,(1 \leq i \leq n)}$. After uploading the data blocks and tags $(\{S_i\}_{[1 \leq i \leq n]})$ to CSP, the user still needs to store his own public keys and private keys, and the storage cost is $2n|Z_q^*|$. In addition to data blocks, CSP also needs to store tags, and the storage cost is $2n|Z_q^*|$. $(\{v_i, m_i, pk_i\}_{[1 \leq i \leq n]})$ is stored at TPA, so the storage overhead is $3n|Z_q^*|$.

In the improved protocol, the user only holds a pair of k_p and k_s, and the storage overhead on the user side is $2|Z_q^*|$. CSP needs to store $(\{S_i, e_i\}_{[1 \leq i \leq n]})$, and the storage overhead is $n|Z_q^*| + n|S|$. When TPA verifies the evidence, he needs the user's public key in addition to the challenge information, and the storage cost is $|Z_q^*|$. The storage cost comparison between the original protocol and the improved protocol is shown in Table 2. The storage overhead of the improved scheme is lower than that of the original scheme.

Because the multiplication and addition operations on Z_q^* have minimal computational overhead compared with other operations, we omit them. In the original protocol, the user needs to calculate $k_{p_i} = g^{k_{s_i}}$, $S_i = (H(m_i) \cdot g^{a_i})^{k_{s_i}}$, and $V_i = g^{a_i} \in G$, and the calculation cost is $4n|E_G| + n|H| + n|M_G|$. CSP needs to calculate $S = \prod_{i=1}^{c} S_i$, and the calculation cost is cM_G. TPA needs to calculate equation (3), and the calculation cost is $2c|M_G| + c|E| + c|H|$.

In the improved protocol, the user needs to calculate $S_i = (H(i) \cdot g^{e_i})^{k_s}$, and the calculation cost is $n|E_G| + n|H| + n|M_G|$. CSP needs to calculate $S = \prod_{i=1}^{c} S_i^{p_i}$ and $\mu\prime = \sum_{i=1}^{c} p_i e_i$, and the calculation cost is $c|M_G| + c|E_G|$. TPA needs to calculate equation (14), and the calculation cost is $c|E_G| + 2|E| + c(H) + c(M_G)$. The calculation cost comparison between the original protocol and the improved protocol is shown in Table 3. Among the entities of the improved scheme, only CSP's calculation overhead is slightly higher than the original scheme. The calculation overhead of TPA and user in the improved scheme is significantly reduced compared with the original scheme.

7. Conclusion

According to the analysis in this study, it is clear that the protocol of Jalil et al. is insecure. We point out the security loophole in the original protocol and attacked it, and then, we propose an audit scheme with higher security and efficiency based on the directions that can be improved.

Data Availability

The data supporting this systematic review were taken from previously reported studies and datasets, which have been cited. The processed data are available from the corresponding author upon request.

Conflicts of Interest

There are no potential conflicts of interest.

Authors' Contributions

Ruifeng Li is responsible for the writing of the article and the construction of the improved scheme, Xu An Wang 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 and the collecting of the information related to this article, Zhengge Yi is responsible for the verification of the security of this article, and Ke Niu revised the finished manuscript.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2017YFB0802000), National Natural Science Foundation of China (No. 62172436 and No. 62102452), 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] S. Jajodia and L. Strous, *Integrity and Internal Control in Information Systems VI*, IICIS, Lausanne, CA, USA, 2003.

[2] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 598–609, Alexandria, VA, USA, Octobner 2007.

[3] A. Juels and B. Kaliski, "Pors: proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pp. 584–597, Alexandria, VA, USA, Octobner 2007.

[4] G. Ateniese, R. Pietro, L. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, pp. 1–10, Istanbul, Turkey, September 2008.

[5] Y. Zhu, H. Hu, G. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2231–2244, 2012.

[6] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," *ACM Transactions on Information and System Security*, vol. 17, no. 4, pp. 1–29, 2015.

[7] H. Jin, H. Jiang, and K. Zhou, "Dynamic and public auditing with fair arbitration for cloud data," *IEEE Transactions on Cloud Computing*, vol. 6, no. 3, pp. 680–693, 2016.

[8] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, "An efficient public auditing protocol with novel dynamic structure for cloud data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2402–2415, 2017.

[9] W. Guo, H. Zhang, S. Qin et al., "Outsourced dynamic provable data possession with batch update for secure cloud storage," *Future Generation Computer Systems*, vol. 95, pp. 309–322, 2019.

[10] K. Yu, Z. Guo, Y. Shen, W. Wang, J. Lin, and T. Sato, "Secure artificial intelligence of things for implicit group recommendations," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2698–2707, 2021.

[11] G. Hou, J. Ma, C. Liang, and J. Li, "Efficient audit protocol supporting virtual nodes in cloud storage," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 5, pp. 1–14, 2020.

[12] K. Yu, L. Tan, L. Lin, X. Cheng, Z. Yi, and T. Sato, "Deep-learning-empowered breast cancer auxiliary diagnosis for 5GB remote E-health," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 54–61, 2021.

[13] Y. Tian, Z. Zhang, J. Xiong, L. Chen, J. Ma, and C. Peng, "Achieving graph clustering privacy preservation based on structure entropy in social IoT," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 1–17, 2021.

[14] T. Deng, X. Li, J. Xiong, and Y. Wu, "POISIDD: privacy-preserving outsourced image sharing scheme with illegal distributor detection in cloud computing," *Multimedia Tools and Applications*, vol. 81, no. 3, pp. 3693–3714, 2021.

[15] C. Wang, S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[16] S. Worku, C. Xu, J. Zhao, and X. He, "Secure and efficient privacy-preserving public auditing scheme for cloud storage," *Computers & Electrical Engineering*, vol. 40, no. 5, pp. 1703–1713, 2014.

[17] B. Wang, B. Li, and H. Li, "Oruta: privacy-preserving public auditing for shared data in the cloud," *IEEE transactions on cloud computing*, vol. 2, no. 1, pp. 43–56, 2014.

[18] J. Xiong, F. Li, J. Ma, X. Liu, Z. Yao, and P. Chen, "A full lifecycle privacy protection scheme for sensitive data in cloud computing," *Peer-to-Peer Networking and Applications*, vol. 8, no. 6, pp. 1025–1037, 2015.

[19] 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.

[20] J. Zhao, C. Xu, F. Li, and W. Zhang, "Identity-based public verification with privacy-preserving for data storage security in cloud computing," *IEICE - Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E96.A, no. 12, pp. 2709–2716, 2013.

[21] H. Wang, Q. Wu, B. Qin, and J. Doming, "Identity-based remote data possession checking in public clouds," *IET Information Security*, vol. 8, no. 2, pp. 114–121, 2014.

[22] H. Wang, D. He, and S. Tang, "Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1165–1176, 2016.

[23] Y. Yu, M. Au, G. Ateniese et al., "Identity-based remote data integrity checking with perfect data privacy preserving for

cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 767–778, 2017.

[24] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, and K. Choo, "Fuzzy identity-based data integrity auditing for reliable cloud storage systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 1, pp. 72–83, 2019.

[25] J. Xue, C. Xu, J. Zhao, and J. Ma, "Identity-based public auditing for cloud storage systems against malicious auditors via blockchain," *Science China (Information Sciences)*, vol. 62, no. 3, pp. 45–60, 2019.

[26] L. Tan, K. Yu, C. Yang, K. Choo, and A. Bashir, "A blockchain-based Shamir's threshold cryptography for data protection in industrial internet of things of smart city," in *Proceedings of the 1st Workshop on Artificial Intelligence and Blockchain Technologies for Smart Cities with 6G*, pp. 13–18, New Orleans, Louisiana, United States, October 2021.

[27] S. Peng, F. Zhou, J. Li, Q. Wang, and Z. Xu, "Efficient, dynamic and identity-based remote data integrity checking for multiple replicas," *Journal of Network and Computer Applications*, vol. 134, no. 5, pp. 72–88, 2019.

[28] R. Rabaninejad, M. Asaar, M. Attari, and M. Aref, "An identity-based online/offline secure cloud storage auditing scheme," *Cluster Computing*, vol. 23, no. 5, pp. 1455–1468, 2019.

[29] S. Al-Riyami and K. Paterson, "Certificateless public key cryptography," in *Proceedings of the 9th International Conference on the Theory and Application of Cryptology*, pp. 452–473, Tainan, Taiwan, China, 2003.

[30] B. Wang, B. Li, H. Li, and F. Li, "Certificateless public auditing for data integrity in the cloud," in *Proceedings of the Communications and Network Security (CNS)*, pp. 136–144, Washington, D. C., USA, October 2013.

[31] D. He, S. Zeadally, and L. Wu, "Certificateless public auditing scheme for cloud-assisted wireless body area networks," *IEEE Systems Journal*, vol. 12, no. 1, pp. 64–73, 2018.

[32] Y. Zhang, C. Xu, S. Yu, H. Li, and X. Zhang, "SCLPV: secure certificateless public verification for cloud-based cyber-physical-social systems against malicious auditors," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 4, pp. 159–170, 2015.

[33] B. Kang, J. Wang, and D. Shao, "Certificateless public auditing with privacy preserving for cloud-assisted wireless body area networks," *Mobile Information Systems*, vol. 2017, no. 3, pp. 1–5, 2017.

[34] D. He, N. Kumar, H. Wang, L. Wang, and K. Choo, "Privacy-preserving certificateless provable data possession scheme for big data storage on cloud," *Applied Mathematics and Computation*, vol. 314, no. 1, pp. 31–43, 2017.

[35] D. He, N. Kumar, S. Zeadally, and H. Wang, "Certificateless provable data possession scheme for cloud-based smart grid data management systems," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1232–1241, 2017.

[36] H. Yang, S. Jiang, W. Shen, and Z. Lei, "Certificateless provable group shared data possession with comprehensive privacy preservation for cloud storage," *Future Internet*, vol. 10, no. 6, pp. 1–17, 2018.

[37] G. Wu, Y. Mu, W. Susilo, F. Guo, and F. Zhang, "Privacy-Preserving certificateless cloud auditing with multiple users," *Wireless Personal Communications*, vol. 106, no. 3, pp. 1161–1182, 2019.

[38] L. Huang, J. Zhou, G. Zhang, and M. Zhang, "Certificateless public verification for data storage and sharing in the cloud," *Chinese Journal of Electronics*, vol. 29, no. 4, pp. 639–647, 2020.

[39] B. Jalil, T. Hasan, G. Mahmood, and H. Noman, "A secure and efficient public auditing system of cloud storage based on BLS signature and automatic blocker protocol," *Journal of King Saud University - Computer and Information Sciences*, vol. 2021, no. 9, pp. 1–14, 2021.

[40] Q. Li, B. Xia, H. Huang, Y. Zhang, and T. Zhang, "TRAC: traceable and revocable access control scheme for mHealth in 5G-enabled IioT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3437–3448, 2021.

[41] K. Yu, L. Tan, S. Mumtaz et al., "Securing critical infrastructures: deep-learning-based threat detection in IioT," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 76–82, 2021.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 6841428, 11 pages
https://doi.org/10.1155/2022/6841428

Research Article

Practical Undeniable Multiparty Drawing-Straw Protocol in Asynchronous Networks for Resource-Constrained Information Systems

Ching-Fang Hsu,[1] **Lein Harn**,[2] **Zhe Xia**,[3] **and Hang Xu**[1]

[1] *Computer School, Central China Normal University, Wuhan 430079, China*
[2] *Department of Computer Science Electrical Engineering, University of Missouri, Kansas, MO 64110, USA*
[3] *Department of Computer Science, Wuhan University of Technology, Wuhan 430071, China*

Correspondence should be addressed to Zhe Xia; xiazhe@whut.edu.cn

Received 1 April 2022; Revised 30 April 2022; Accepted 6 May 2022; Published 26 May 2022

Academic Editor: Jinbo Xiong

The next generation of mobile networks and communications (5G networks) has a very strong ability to compute, store, and so on. Group-oriented applications demonstrate their potential ability in resource-constrained information systems (RISs) towards 5G. The security issues in RIS towards 5G have attracted great attention. For example, how to conduct fair and orderly multiparty communication in an intelligent transportation system (ITS). One of the main challenges for secure group-oriented applications in RIS towards 5G is how to manage RIS communications fairly in multiparty applications. In other words, when the users cannot transmit their messages simultaneously, the order of their communication can cause security concerns in multiparty applications. A feasible solution to the problem is for the group of users to follow a specific order to transmit their messages. Otherwise, some users may take advantage over other users if there has no agreeable order to be followed. In this paper, we propose a novel cryptographic primitive, called multiparty drawing-straw (MDS) protocol, which can be used by a group of users to determine the order of the group to participate in the multiparty applications. Our scheme is based on Pedersen's verifiable secret sharing (VSS), which is a well-known scheme. Our proposed protocol is fair since the output is uniformly distributed, and this is an attractive feature for secure multiparty applications in RIS towards 5G.

1. Introduction

Since the emergence of the next generation of mobile networks and communications (5G), technologies such as resource-constrained information systems (RIS) towards 5G have attracted more attention, as various smart devices have been constantly connected to the Internet over the past decades. The number of devices connected to the Internet is increasing since its appearance. Now, this number far exceeds that of people in the world, we are no longer talking about the Internet but about the internet of things (IoT). IoT gives rise to revolutionary applications for emerging technologies of RIS towards 5G. group-oriented applications also show the great potential of society.

Group-oriented applications demonstrate the importance of RIS towards 5G, such as joint data collection for traffic analysis, weather prediction, multiuser interactive computation, and so on. These applications motivate the demand for secure group-oriented applications over open and insecure networks. In particular, the devices in RIS towards 5G are heterogeneous, and the RIS communication environments are asynchronous, where multiple users cannot transmit their messages simultaneously. One of the main challenges for secure group-oriented applications in RIS towards 5G is how to secure the communications among these heterogeneous devices in such an asynchronous environment. Note that it is widely known that asynchronous transmission can cause security problems in cryptographic functions.

Devices in RIS towards 5G generate, process, and exchange vast amounts of security and safety-critical data as well as privacy-sensitive information; hence, they are appealing targets of various attacks [1–8]. To ensure the correct and safe operation of RIS towards 5G systems, it is crucial to ensure the integrity of the underlying devices, in particular of their code and data, against malicious modifications [9]. Recent researches have revealed many security vulnerabilities in the embedded devices [2, 4, 6, 7, 10, 11]. This highlights new challenges in the design and implementation of secure embedded systems that typically must provide multiple functions, security features, and real-time guarantees at a minimal cost [12]. How to design a lightweight protocol to determine the order of the group members to communicate in RIS towards 5G applications is needed in a network that involves multiple devices/users. For example, how to manage multiparty communication in an intelligent transportation system (ITS; see Figure 1). In some specific applications, like multiparty bidding or multiparty gaming, for the sake of fairness among users, users need to transmit their messages in a particular order. Otherwise, users can gain unfair advantages over other users if there has no particular order to be followed. Although users can rely on a mutually trusted center to decide this order, most Internet users would prefer to make their own decisions. The objective of this paper is to design such a lightweight protocol to determine the order of the group members to participate in applications.

In many multiparty applications, the users need to follow a specific order to transmit their messages. Otherwise, messages can collide with each other if multiple transmissions occurred simultaneously. Moreover, in some applications, a user who on purposely transmits his message last may gain unfair advantages. Coin flipping is a simple way of deciding the order between two users. It is widely used in sports and other games to decide the random factors such as which side of the field a team will play from or which side will attack or defend initially. Coin-flipping protocol is a cryptographic primitive that has been introduced by Blum [13] and is one of the basic building blocks of secure two-party computation. Coin flipping is the process of throwing a coin into the air to choose between two possible and equally likely outputs. In cryptography, a commitment scheme can be used to achieve a coin-flipping protocol. Aharonov et al. [14] proposed a quantum protocol with no dishonest player that can bias the coin with a probability higher than 0.9143. Ambainis [15] proposed an improved protocol with cheating probability at most 3/4. Since then, several different protocols have been proposed [16, 17] that achieve the same bound of 3/4. In the following, we describe Blum's two-party coin-flipping protocol [13] between Alice and Bob:

(i) Alice chooses a random bit $a \in \{0, 1\}$ and sends a commitment $c = commit\ (a)$ to Bob

(ii) Bob chooses a random bit $b \in \{0, 1\}$ and sends it to Alice

(iii) Alice sends the bit a to Bob together with $decommit\ (c)$

(iv) If Bob does not abort during the protocol, Alice outputs $a \oplus b$; otherwise, she outputs a random bit

(v) If Alice does not abort during the protocol and c is a commitment to a, then Bob outputs $a \oplus b$; otherwise, he outputs a random bit

The trend of network applications inspires us to consider scenarios involving multiple players. A novel cryptographic primitive, called *multiparty drawing-straw protocol* (MDS), is introduced in this paper. This technique can be used by a group of users to determine the order of the group to participate in applications. The users need to follow the decided order to take turns to make a movement or release a message in these applications. For example, in multiparty computation, multiple parties want to jointly compute a function over their inputs and keep these inputs private. MDS can be used to determine the order of releasing their inputs. In a real-world solution to provide MDS, the group leader prepares a set of straws of different lengths. Each user of the group randomly draws a straw from the unseen set of straws prepared by the group leader. At the end of the offering, the order of the group is determined by the lengths of straws chosen by group users. The fairness of this solution depends on the trustworthiness of the group leader. If the group leader colludes with any group member, the output of this process can be biased. The requirement of a mutually trusted party is unrealistic in some applications. The MDS without the assistance of a mutually trusted party is desirable.

If there are only two players in MDS, the coin-flipping protocol [13, 18] can be used to determine the order of players. The "*winner*" of a coin-flipping protocol can be the starter. Thus, the coin-flipping protocol is a special type of MDS. Actually, we can use the coin-flipping protocol in a straightforward manner to provide a solution for a general MDS. In this solution, all players are arranged on the leaves of a binary tree. Using a coin-flipping protocol between two users can determine a "*winner.*" Repeatedly executing the coin-flipping protocol multiple times following the tree structure (i.e., with complexity $O(n)$ can determine the "*1st winner*" among n users). Then, using the same approach on remaining $n-1$ users can determine the "*2nd winner*" and so on. However, this approach is not effective because of the time-consuming process. Multiparty coin-flipping protocols [19–21] have been developed recently. However, these protocols are restricted for multiple parties to jointly choose one of the two possible outputs, which are different from our MDS. In 2008, Lit et al. [22] have proposed a secure multiparty ranking problem (SMR) [22], which is extended from Yao's Millionaires' problem [23]. This problem has been studied in [24]. We assume that there are n users and each user has a secret input. In SMR, it intends to get the order of inputs in the ascending ranking sequence while not leaking the value of any input. In particular, each user knows his order of input but does not know the orders of the other users' inputs. The SMR is different from MDS since (a) in SMR each user knows only the order of his input, but in MDS each user knows all inputs, and (b) in SMR the inputs

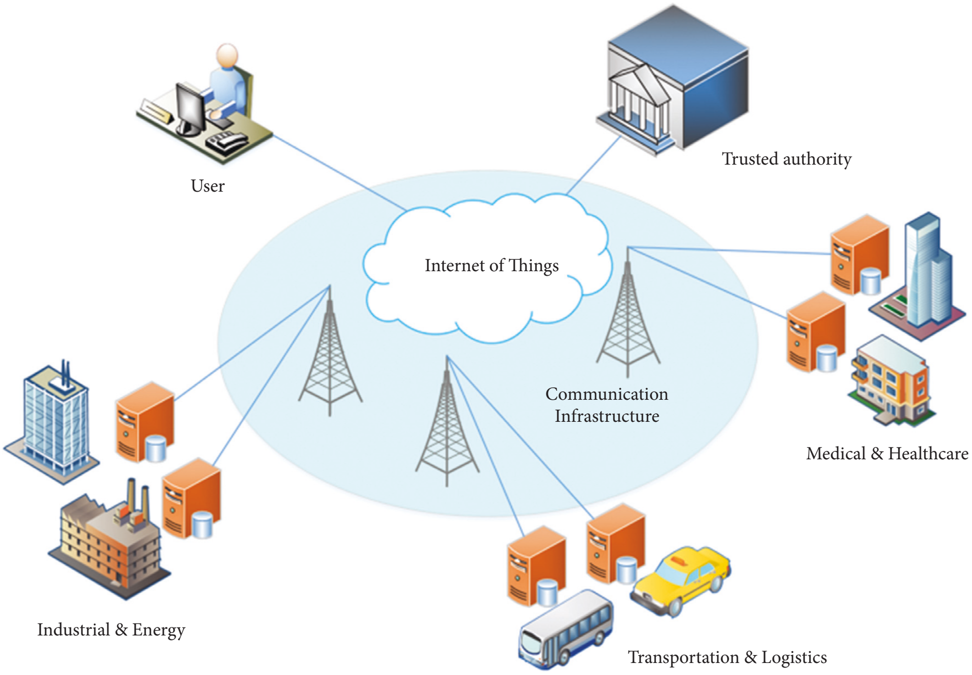


Figure 1: A typical ITS model.

are users' data, but in MDS the inputs are random secrets selected by users.

In this paper, we propose a *multiparty undeniable drawing-straw protocol* (MUDS). Formally speaking, this primitive realizes the following function: $\underbrace{(s_1, s_2, \ldots, s_n)}_{n} \mapsto (o_1, o_2, \ldots, o_n)^n$, where $\underbrace{(s_1, s_2, \ldots, s_n)}_{n}$ are an array including n secret inputs and $(o_1, o_2, \ldots, o_n)$ is a permutation uniformly distributed over $\{1, 2, 3, \ldots, n\}$. User U_i receives his order o_i along with other orders. We present a protocol for this primitive. Our protocol consists of two phases, the commitment phase and the open-commitment phase. In the commitment phase, every group user chooses a secret and releases a commitment of the secret. In the second phase, every group user opens the commitment by revealing the secret. The output of the group order is determined by all released secrets of group users. In our protocol, after making any commitment in phase 1, the user can no longer deny his commitment in phase 2 since we employ a *secret sharing scheme* (SS) and *verifiable secret sharing scheme* (VSS) in our design. If the majority of users are honest, we can prove that our protocol is secure by setting the threshold, t, as $t = [n/2]$, where n is the number of group users.

Many communication networks in practice are asynchronous in which multiparty users cannot transmit their messages simultaneously. The asynchronous transmission can cause security problems in cryptographic functions. For example, the work in [19] has used Blum's two-party coin-flipping protocol [13] to demonstrate the problem. The main concern in designing coin-flipping protocols is to prevent the bias of the output. The bias of a coin-flipping protocol measures the maximum influence of malicious parties on the output of the honest parties. The bias is 0 if the output is always uniformly distributed, and the bias is 1/2 if the adversary can force the output to be always (say) 1. In Blum's protocol [13], since Alice recovers the output $a \oplus b$ before sending her decommit (c) to Bob, [19] demonstrated that Alice has an advantage over Bob, and the bias of the protocol is 1/4.

We use another example, *rational secret sharing* [25], to demonstrate the security problem caused by asynchronous networks. In 2004, Halpern et al. [25] considered a scenario in which users in the secret reconstruction are neither completely honest nor arbitrarily malicious; instead, the users are assumed to be rational. In rational secret sharing, it assumes that all users are rational and want to maximize their utility. A rational user acts honestly when he cannot gain any advantage over other users (i.e., they will all obtain the secret) but acts dishonestly when he can gain an advantage over others (i.e., he is the only one to obtain the secret). The classical SSs including Shamir's SS [26] fail in a rational setting. In Shamir's secret reconstruction, if the user who broadcasts his share last, the user will see that others have broadcasted their shares already. Thus, he will remain silent because other parties cannot reconstruct the secret, but this user can reconstruct the secret by using his own share and shares broadcasted by other parties. There are vast research papers on rational secret sharing (RSS) [27–29]. In fact, the objective of the RSS scheme is to ensure rational users that the secret can be reconstructed successfully. This is the same as that of the fair secret reconstruction scheme, which was originally proposed by Tompa et al. in [30] in 1988. In most RSS schemes, information exchanged among users is restricted to be in a synchronous network. There are only handful papers on RSS schemes assuming asynchronous networks. These include Fuchsbauer et al.'s result [27], which requires cryptographic primitives, and the results of Ong et al. [28] and Moses et al. [31], which require to assume that a certain number of shareholders must be honest.

The motivation of our paper is to develop a multiparty coin-flipping protocol that involves more than two parties without the assistance of a mutually trusted party. RIS towards 5G devices can employ this protocol to determine their order in Internet applications. Since the order is determined by all RIS towards 5G devices, this protocol needs to prevent dishonest devices from cheating in the process. We consider adopting both Shamir's SS and Pedersen's noninteractive VSS to achieve this objective. The primary reason to adopt both schemes is due to their simplicity to be implemented in an asynchronous network. Shamir's SS is unconditionally secure and polynomial-based, which is the most widely used secret sharing scheme since it is simple and computationally efficient. While for Pedersen's VSS, the privacy of Pedersen's VSS is unconditionally secure, and the correctness of the shares is based on a computational assumption. In Pedersen's VSS, verification is based on the commitments computed by the owner of the secret, and there is no interaction among verifiers during verification. The verification of shares can be performed by each verifier individually.

We propose a novel design to overcome the security problem caused by asynchronous networks. Our protocol is *undeniable* since in the process of making a commitment for the secret, where the secret needs to be shared by all other users. Moreover, shares can be verified by other users. Thus, after making a commitment, the user can no longer deny his commitment. If some user denies making the commitment, the secret can still be able to reconstruct by other honest users. The undeniable feature prevents the users from disrupting the protocol by releasing either a fake secret or no information after making their commitments. Our protocol is based on Shamir's SS [26] and Pedersen's verifiable secret sharing scheme (VSS) [32]. Thus, our design is particularly suitable for group-oriented applications in RIS towards 5G.

Here, we summarize the contributions of our paper.

(i) A novel cryptographic primitive, called *multiparty drawing-straw protocol* (MDS), is introduced

(ii) An MDS protocol with undeniability (MUDS) is proposed that can overcome the security problem caused by asynchronous networks

(iii) MUDS is useful in many multiparty applications, providing a fair way to determine an order of users

The rest of the paper is organized as follows. In Section 2, we present some preliminaries. The model of our proposed protocol is introduced in Section 3, including protocol description, type of entities, and attacks of the proposed protocol. The protocol is outlined in Section 4. We give a concrete protocol in Section 5. The conclusion is given in Section 6.

2. Preliminaries

Our proposed MUDS is based on Pedersen's VSS [33]. We review this scheme in this section.

Chor et al. [34] proposed the notion of VSS in which shareholders can verify that their shares are valid without revealing the secrecy of their shares and the secret. We give a definition of VSS below.

Definition 1 (*t-out-of-n* verifiable secret sharing scheme (VSS)). A *t-out-of-n* verifiable secret sharing scheme $\pi = (G, R, V)$ consists of a sharing algorithm G, a reconstruction algorithm R, and a verification algorithm V. The sharing algorithm G guarantees that it is impossible for any adversary to reconstruct the secret from fewer than t shares. The reconstruction algorithm R guarantees that the secret can be recovered from any t or more than t shares. The verification algorithm V guarantees that shareholders can verify their shares are generated consistently without compromising the secrecy of both their shares and the secret.

VSSs of Feldman [32] and Pedersen [33] are based on cryptographic commitment schemes. The security of Feldman's VSS is on the hardness of solving discrete logarithm, while the privacy of Pedersen's VSS is unconditionally secure, and the correctness of the shares is based on a computational assumption. Benaloh [35] proposed an interactive VSS, which is unconditionally secure. Stadler [36] proposed the first publicly verifiable secret sharing (PVSS) scheme that allows each shareholder to verify the validity of all shares. Most noninteractive VSSs [30, 32] can only verify the validity of his/her own share, but not of other shareholders' shares. The security of Schoenmaker's PVSS [37] is based on the discrete logarithm problem. Peng and Wang's PVSS [38] uses a linear code, and Ruiz and Villar's PVSS [39] uses Pailler's cryptosystem [40]. There are noninteractive PVSSs based on bilinear pairing [41, 42].

Pedersen's VSS is information-theoretic secure. There are public parameters, $g, h \in Z_p$, and assumes that no one knows $\log_g h$. Pedersen's VSS uses the following commitment scheme.

Pedersen's Commitment Scheme. To commit the secret S, the dealer computes and publishes a commitment $E(s, k) = g^s h^k = E_0 \mathrm{mod} p$, where k is a random integer with $k \in Z_p$. Such a commitment can later be opened by releasing s and k.

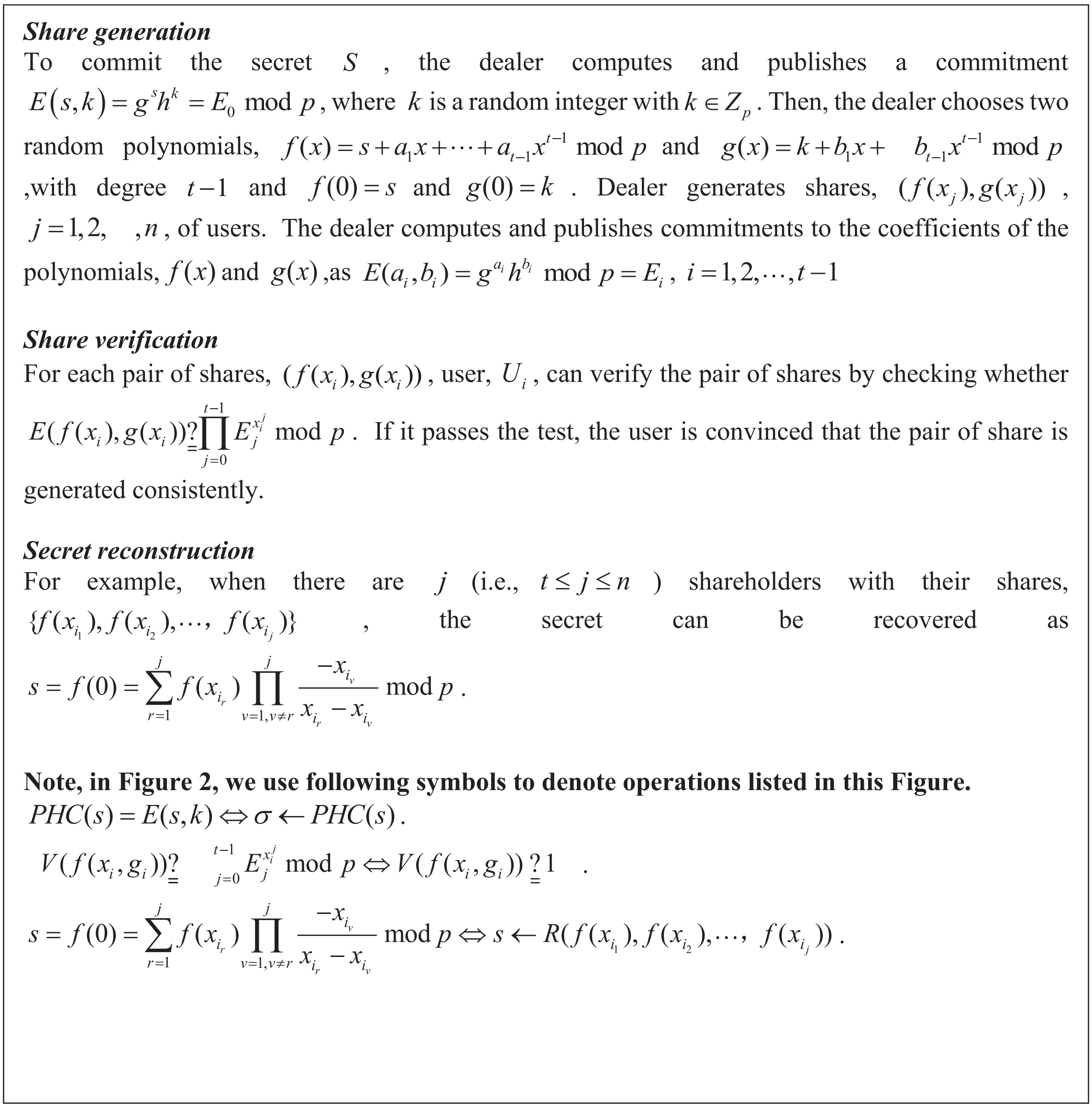
Definition 2 (perfectly hiding commitment (PHC) scheme). A commitment scheme is perfectly hiding if it does not reveal any information about the committed value in the commitment phase.

In [33], it has proven that the commitment E_0 reveals no information on the secret S and that the committer cannot open a commitment to s as $s' \neq s$ unless he can solve $\log_g h$. Pedersen's commitment scheme is a PHC. Pedersen's VSS consists of three algorithms as shown in Figure 2.

3. Model

3.1. Description of MUDS. MDS deals with the following setting: n users, $U = \{U_1, U_2, \ldots, U_n\}$, interactively work together to uniformly choose an order for some computation. In the following, we give a formal definition of MDS.

Definition 3 (multiparty drawing-straw (MDS) protocol). MDS computes the following functionality:



Share generation

To commit the secret S, the dealer computes and publishes a commitment $E(s,k) = g^s h^k = E_0 \bmod p$, where k is a random integer with $k \in Z_p$. Then, the dealer chooses two random polynomials, $f(x) = s + a_1 x + \cdots + a_{t-1} x^{t-1} \bmod p$ and $g(x) = k + b_1 x + \quad b_{t-1} x^{t-1} \bmod p$,with degree $t-1$ and $f(0) = s$ and $g(0) = k$. Dealer generates shares, $(f(x_j), g(x_j))$, $j = 1, 2, \quad , n$, of users. The dealer computes and publishes commitments to the coefficients of the polynomials, $f(x)$ and $g(x)$,as $E(a_i, b_i) = g^{a_i} h^{b_i} \bmod p = E_i$, $i = 1, 2, \cdots, t-1$

Share verification

For each pair of shares, $(f(x_i), g(x_i))$, user, U_i, can verify the pair of shares by checking whether $E(f(x_i), g(x_i)) \overset{?}{=} \prod_{j=0}^{t-1} E_j^{x_i^j} \bmod p$. If it passes the test, the user is convinced that the pair of share is generated consistently.

Secret reconstruction

For example, when there are j (i.e., $t \le j \le n$) shareholders with their shares, $\{f(x_{i_1}), f(x_{i_2}), \cdots, f(x_{i_j})\}$, the secret can be recovered as

$$s = f(0) = \sum_{r=1}^{j} f(x_{i_r}) \prod_{v=1, v \neq r}^{j} \frac{-x_{i_v}}{x_{i_r} - x_{i_v}} \bmod p .$$

Note, in Figure 2, we use following symbols to denote operations listed in this Figure.

$PHC(s) = E(s,k) \Leftrightarrow \sigma \leftarrow PHC(s)$.

$V(f(x_i, g_i)) \overset{?}{=} \quad {}_{j=0}^{t-1} E_j^{x_i^j} \bmod p \Leftrightarrow V(f(x_i, g_i)) \overset{?}{=} 1$.

$$s = f(0) = \sum_{r=1}^{j} f(x_{i_r}) \prod_{v=1, v \neq r}^{j} \frac{-x_{i_v}}{x_{i_r} - x_{i_v}} \bmod p \Leftrightarrow s \leftarrow R(f(x_{i_1}), f(x_{i_2}), \cdots, f(x_{i_j})) .$$

Figure 2: Pedersen's VSS.

$$\underbrace{(s_1, s_2, \cdots, s_n)}_{n} \mapsto (o_1, o_2, \cdots, o_n)^n, \qquad (1)$$

where l is a security parameter, $\underbrace{(s_1, s_2, \cdots, s_n)}_{n}$ are an array including n secret inputs, and $(o_1, o_2, \ldots, o_n)$ is a permutation uniformly distributed over $\{1, 2, 3, \ldots, n\}$. At the end, each user U_i obtains his order o_i along with an order of others.

The output permutation $(o_1, o_2, \ldots, o_n)$ obtained in MDS should be determined by all users without the assistance of a mutually trusted third party. We adopt Pedersen's VSS in our design. In the first phase, each user needs to select a random secret and then compute and release a commitment of the secret to all other users. In the second phase, each user releases his secret to all other users. To avoid the problem caused by any user who may deny releasing his real secret in the second phase, we introduce the following definition of MUDS.

Definition 4 (multiparty undeniable drawing-straw (MUDS) protocol). In a MUDS, no user can deny his secret after revealing his commitment of the secret to others. The security of this functionality is called undeniability.

3.2. Entities and Possible Attacks. The security objective of our protocol is to enable all IoT devices to work together to determine the order among them in a fairway. Since the protocol allows each device to contribute an input and the final order is determined by all inputs, our protocol needs to prevent dishonest devices from cheating in the process. We consider dishonest devices (also called *attackers*) may take advantage of most asynchronous networks by releasing their inputs last. Thus, we divide our protocol into two phases. In the first phase, each input is divided into shares by the device owner. Shares are distributed to other devices secretly. A commitment of this input is also published by the owner.

Other devices can verify that their shares are generated consistently by the owner. In the second phase, each input can be released in any asynchronous way. If any dishonest device owner refuses to release their input or releases a fake input, other honest devices can work together to recover the input. We adopt Shamir's SS and Pedersen's VSS to achieve this objective.

In a VSS, the owner of the secret is the *prover*, and all other users are the *verifiers*. The verifiers want to verify that their shares are generated consistently without compromising the secrecy of the secret. In Pedersen's VSS, verification is based on the commitments computed by the owner of the secret, and there is no interaction among verifiers during verification. The verification of shares can be performed by each verifier individually. Inconsistent shares may be generated due to the following two reasons: (a) in shares generation/distribution, nature noise, such as transmission noise or computational error, may cause the inconsistency and (b) inconsistent shares may be generated by a user who tries to cheat other honest users. In summary, we adopt Pedersen's VSS in our design since Pedersen's VSS is (a) unconditionally secure and (b) noninteractive.

Attackers may try to obtain secrets from commitments. Pedersen's commitment can prevent this attack. Moreover, we need to prevent colluded attacks on users. Since communication networks are asynchronous, colluded attackers can always release their fake secrets after knowing the secrets of other users. Any fake secret can be detected by VSS. Moreover, in our protocol, we employ a threshold SS to ensure that any committed secret can always be reconstructed by the majority of honest users if the threshold is $t = [n/2]$, where n is the number of users.

3.3. Properties. Our protocol has the following properties:

Randomness. The output is uniformly distributed. No user can influence the output.

Secrecy. From the commitment of each secret, the secret cannot be recovered. Furthermore, the secret is protected by a threshold SS.

Efficiency. In our proposed protocol, all users work together to determine the output. We use Shamir's SS [26] and Pedersen's VSS [33] based on polynomials. At the beginning of each phase, each user needs to act as a dealer to compute and release values to others. There has no interaction among users to verify shares. Since both Shamir's SS and Pedersen VSS [33] are simple and efficient, our protocol is very efficient.

Undeniability. After publishing any commitment of the secret, the user can no longer deny the secret since the secret can also be recovered by honest users.

4. Proposed Protocol

4.1. Outline. Let us assume that there are n users, $U = \{U_1, U_2, \ldots, U_n\}$, participated in a MUDS. These users need to interactively work together to generate an output, which is the order of the users. In our proposed protocol, there are two phases, the commitment and open-commitment phases. In the commitment phase, each user selects a secret and acts as the dealer to use a threshold SS to generate shares for other users. Each user makes the commitment of the secret publicly known. For each received share, other users can verify that the share is generated consistently by the owner of the secret. If the verification is failed, a request for regeneration of a share can be sent to the owner of the secret till a share is verifiable.

In the open-commitment phase, each user releases his secret of the commitment to other users. Each released secret can be verified by other users using his commitment made in the commitment phase. If any released secret is an invalid secret, other honest users can work together to recover the secret by using their shares of the secret obtained in the commitment phase. This property, called *undeniability*, in our proposed protocol prevents any user to deny his commitment of a secret.

After obtaining all secrets of users, each user can determine the order of group based on the secrets selected by users.

4.2. Protocol. We illustrate the detail of the protocol in Figure 3.

4.3. Security Proof. We say that two probability ensembles are *statistically indistinguishable* if their statistical difference is negligible.

Lemma 1. *In our framework, if all users are honest, the probability ensemble defined by the order* $(o_1, o_2, \ldots, o_n)$ *and the probability ensemble defined by a permutation uniformly chosen over* $\{1, 2, 3, \ldots, n\}$ *are statistically indistinguishable.*

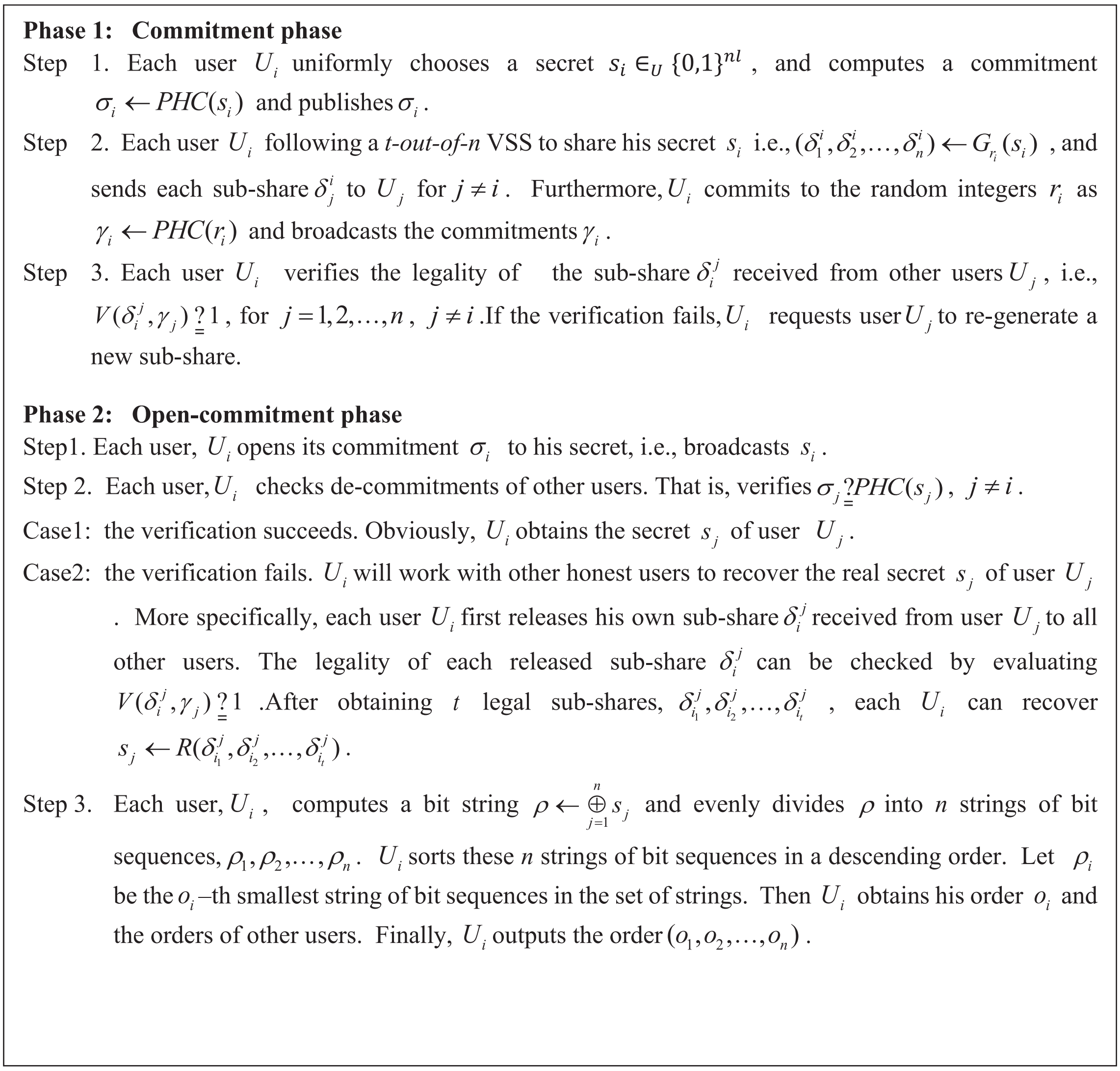
Proof. We observe that the difference between the two probability ensembles is that some orders probably have the same value in the former. That is, the event $\exists i \neq j\,(o_i = o_j)$ may appear in the former ensemble. Fortunately, we can show that this event occurs with a negligible probability.

From step 3, we know that $\Pr[\exists i \neq j\,(o_i = o_j)] = \Pr[\exists i \neq j\,(\rho_i = \rho_j)]$. If all users are honest, then $\rho_1, \rho_2, \ldots, \rho_n$ are uniformly distributed. We have $\Pr[\exists i \neq j\,(\rho_i = \rho_j)] \leq \sum_{i \neq j} \Pr[(\rho_i = \rho_j)] \leq (n/2) \times (1/2^l)$.

Thus, we have $\Pr[\exists i \neq j\,(o_i = o_j)] \leq (n/2) \times (1/2^l)$. Note that $(n/2)$ is a positive constant. Thus, the event $\exists i \neq j\,(o_i = o_j)$ occurs with a negligible probability.

We say that two probability ensembles X, Y are *computationally indistinguishable*, denoted as $X \approx Y$, if there is no probabilistic polynomial-time algorithm distinguishing them. □

Lemma 2. *In our framework, let* $\overline{\rho}$ *denote the probability ensemble defined by* $\rho_1, \rho_2, \ldots, \rho_n$*, and* $\overline{r}$ *denote the probability ensemble defined by a bit-string uniformly chosen over* $\{0, 1\}^{nl}$*. If the commitment scheme is a PHC, the secret sharing*



Phase 1: Commitment phase

Step 1. Each user U_i uniformly chooses a secret $s_i \in_U \{0,1\}^{nl}$, and computes a commitment $\sigma_i \leftarrow PHC(s_i)$ and publishes σ_i.

Step 2. Each user U_i following a *t-out-of-n* VSS to share his secret s_i i.e., $(\delta_1^i,\delta_2^i,\ldots,\delta_n^i) \leftarrow G_{r_i}(s_i)$, and sends each sub-share δ_j^i to U_j for $j \neq i$. Furthermore, U_i commits to the random integers r_i as $\gamma_i \leftarrow PHC(r_i)$ and broadcasts the commitments γ_i.

Step 3. Each user U_i verifies the legality of the sub-share δ_i^j received from other users U_j, i.e., $V(\delta_i^j,\gamma_j) \overset{?}{=} 1$, for $j = 1,2,\ldots,n$, $j \neq i$ .If the verification fails, U_i requests user U_j to re-generate a new sub-share.

Phase 2: Open-commitment phase

Step1. Each user, U_i opens its commitment σ_i to his secret, i.e., broadcasts s_i.

Step 2. Each user, U_i checks de-commitments of other users. That is, verifies $\sigma_j \overset{?}{=} PHC(s_j)$, $j \neq i$.

Case1: the verification succeeds. Obviously, U_i obtains the secret s_j of user U_j.

Case2: the verification fails. U_i will work with other honest users to recover the real secret s_j of user U_j . More specifically, each user U_i first releases his own sub-share δ_i^j received from user U_j to all other users. The legality of each released sub-share δ_i^j can be checked by evaluating $V(\delta_i^j,\gamma_j) \overset{?}{=} 1$ .After obtaining t legal sub-shares, $\delta_{i_1}^j,\delta_{i_2}^j,\ldots,\delta_{i_t}^j$, each U_i can recover $s_j \leftarrow R(\delta_{i_1}^j,\delta_{i_2}^j,\ldots,\delta_{i_t}^j)$.

Step 3. Each user, U_i, computes a bit string $\rho \leftarrow \oplus_{j=1}^{n} s_j$ and evenly divides ρ into n strings of bit sequences, $\rho_1,\rho_2,\ldots,\rho_n$. U_i sorts these n strings of bit sequences in a descending order. Let ρ_i be the o_i–th smallest string of bit sequences in the set of strings. Then U_i obtains his order o_i and the orders of other users. Finally, U_i outputs the order $(o_1,o_2,\ldots,o_n)$.

Figure 3: Proposed protocol.

scheme is a t-out-of-n VSS, and there are at most $t-1$ malicious users, then $\overline{r} \approx \overline{\rho}$.

Proof. Let us focus on $\overline{\rho}$ and assume that there is at most one malicious user U_i. There are the following types of attacks that can possibly bias $\overline{\rho}$.

(1) The secret S_i chosen by malicious U_i is not uniformly distributed.

Note that $\rho \leftarrow \oplus_{j=1}^{n} s_j$. Thus, U_i cannot bias $\overline{\rho}$, and then $\overline{r} \approx \overline{\rho}$ holds.

(2) Malicious user U_i may refuse to release their decommitment or releases an illegal decommitment.

The *t-out-of-n* VSS guarantees that other honest users can recover the secret. This attack cannot bias $\overline{\rho}$, and then $\overline{r} \approx \overline{\rho}$ holds.

(3) In step 1, phase 2, U_i may open his commitment σ_i to be a maliciously chosen value $\widetilde{S}_i$, which is different from the real value he has committed in step 1 of phase 1.

If U_i succeeds in this cheating, $\overline{\rho}$ is not uniformly distributed. Fortunately, the computationally binding the commitment guarantees that the success probability of this cheating is negligible. Thus, $\overline{r} \approx \overline{\rho}$.

In a similar way, we can prove that this lemma holds even if there are at most $t-1$ malicious users. □

Theorem 1 (the distribution of the output). *In our framework, if the commitment scheme is perfectly hiding, the secret sharing scheme is a t-out-of-n VSS, and there are at most $t-1$ malicious users, then the probability ensemble defined by the order $(o_1, o_2, \ldots, o_n)$ and the probability ensemble defined by a permutation uniformly chosen over $\{1, 2, 3, \ldots, n\}$ are computationally indistinguishable.*

Proof. Let $\overline{o}$ denote the probability ensemble defined by the order $(o_1, o_2, \ldots, o_n)$ and σ' denote the probability ensemble defined by a permutation uniformly chosen over $\{1, 2, 3, \ldots, n\}$. Then we want to show $\overline{o} \approx \overline{o'}$.

Let us focus on step 3 in phase 2. We know that $\overline{o}$ is a function of $\overline{\rho}$, denoted by $\overline{o} = f(\overline{\rho})$. Following Lemma 2, we have $\overline{r} \approx \overline{\rho}$. Thus, $f(\overline{\rho}) \approx f(\overline{r})$. Furthermore, we have $\overline{o} \approx f(\overline{r})$. Note that $f(\overline{r})$ describes the case when all users are honest. From Lemma 1, we have $\overline{o'} \approx f(\overline{r})$. Finally, we obtain $\overline{o'} \approx \overline{o}$. □

Remark 1. The distribution of the order obtained by our protocol is not uniformly distributed. Note that $\overline{o} = f(\overline{\rho})$ and the string ρ is not uniformly distributed. From the proof of Lemma 2, we know that the malicious user may break the binding in step 1, phase 2, although the probability is negligible. One may prefer to employ a perfectly binding commitment scheme instead. In this case, the malicious users may break the computational hiding of the scheme in step 1, phase 1, although the probability is negligible too, and bias the distribution of ρ.

Remark 2. It is easy to verify that Theorem 1 still holds even if the commitment scheme is computationally binding and computationally hiding.

Theorem 2 (undeniability). *In our framework, if the commitment scheme is perfectly hiding, the secret sharing scheme is a t-out-of-n VSS, and there are at most $t - 1$ malicious users; then no user can deny his secret after revealing his commitment of the secret to others.*

Proof. Note that the threshold of *t-out-of-n* VSS is set to be $t = [n/2]$, to generate subshares for users. Thus, after making any commitment of secret, the user can no longer deny the commitment. If the user tries to deny the commitment by either releasing a fake subsecret or releasing no information, the subsecret can still be recovered by honest users. □

Theorem 3 (secrecy). *In our framework, if the commitment scheme is perfectly hiding, the secret sharing scheme is a t-out-of-n VSS, and there are at most $t - 1$ malicious users; then each subsecret of the user cannot be recovered from its commitment and is protected by a threshold SS.*

Proof. This property should be satisfied using a *t-out-of-n* VSS with a PHC commitment scheme as defined by Definition 2. □

4.4. Efficiency. In phase 1, each user needs to employ a one-time share sharing algorithm G to generate subshares for other users, a one-time commitment algorithm PHC to commit his secret and $(n - 1)$-time share verification algorithms V to verify subshares received from other users. In phase 2, if we assume that all users act honestly by releasing their secrets, each user needs to employ $(n - 1)$-time commitment verification algorithm to verify each released secret of other users. Thus, we can conclude that the complexity of using this approach is $O(n)$ for a group with n members.

To the best of our knowledge, our proposed scheme is the first MDS that can be used for a group of players to fairly determine the order of the group in applications. In particular, the coin-flipping protocol is a special type of MDS, that is, there are only two players. Although multiparty coin-flipping protocols [19–21] have been developed recently, these protocols are restricted for multiple parties to jointly choose one of the two possible outputs, which are different from our MDS.

When we compare our protocol with the coin-flipping protocol, our protocol is more efficient. As we have mentioned earlier in the introduction that employing a coin-flipping protocol repeatedly can achieve the same objective as ours. However, the coin-flipping protocol works two devices at a time to determine their order. That is, we can use the coin-flipping protocol in a straightforward manner to provide a solution for a general MDS. In this approach, all players are arranged on the leaves of a binary tree. Using a coin-flipping protocol between two users can determine a "winner." Repeatedly executing the coin-flipping protocol multiple times following the tree structure (i.e., complexity $O(n)$ can determine the "1st winner" among n users). Then, using the same approach on the remaining $n - 1$ users can determine the "2nd winner" and so on. However, this approach is not effective because the complexity of using this approach is $O(n^2)$ for a group with n members. It is a time-consuming process. Thus, for large size of devices, it takes too much computational delay to determine their final order. In our proposed protocol, all devices work together at once to determine their order with the complexity $O(n)$. We are currently building a test platform. With the completion of this platform, we will take practical measurements and make comparisons with other approaches as described in [43] for our future work.

5. Concrete Instantiation

5.1. Protocol. We illustrate the detail of the proposed protocol by a concrete instantiation in Figure 4, where we use Pedersen's verifiable secret sharing (VSS) to allow each user to commit to a secret in the first phase. As follows, the user uses Shamir's secret sharing to divide the secret into multiple shares and share it among other users. The validity of these shares can be verified using VSS. In the second phase, each user releases his secret to the other users. If the user tries to deny his commitment by either releasing a fake secret or refusing to release the secret, misbehavior can be detected. At this moment, the other honest users can work together to recover this secret. This feature, called undeniability, can prevent the users from denying their secrets after making the commitments. The output of MDS $(o_1, o_2, \ldots, o_n)$ is a function of all secrets $(s_1, s_2, \ldots, s_n)$ of users.

5.2. Efficiency and Features. In this section, we evaluate the efficiency and summarize the features of this concrete protocol.

Efficiency. In our proposed protocol, all users work together to determine the output. Our protocol does not depend on any mutually trusted party. At the beginning of

Phase 1: Commitment phase

There are some public parameters, where p is a prime and $g, h \in Z_p$, and assumes that no one knows $\log_g h$.

Step1. Each user, U_i, randomly selects a sub-secret, s_i, and computes and publishes a commitment $\sigma(s_i, k_i) = g^{s_i} h^{k_i} = \sigma_0^i \bmod p$ where k_i is a random integer with $k_i \in Z_p$.

Step2. Each user, U_i, chooses two random polynomials, $f_i(x) = s_i + a_{i,1}x + \cdots + a_{i,t-1}x^{t-1} \bmod p$ and $g_i(x) = k_i + b_{i,1}x + \cdots + b_{i,t-1}x^{t-1} \bmod p$,with degree $t-1$ and $f_i(0) = s_i$ and $g_i(0) = k_i$. U_i generates a pair of sub-shares, $(f_i(x_j), g_i(x_j))$, for each other user, $j = 1, 2, \cdots, n$, $j \neq i$. Each pair of sub-shares, $(f_i(x_j), g_i(x_j))$ are sent to user, U_j, secretly.

Step 3. U_i computes and publishes commitments to the coefficients of the polynomials, $f_i(x)$ and $g_i(x)$ as $\sigma(a_{i,j}, b_{i,j}) = g^{a_{i,j}} h^{b_{i,j}} \bmod p = \sigma_j^i$, $j = 1, 2, \ldots, t-1$.

Step 4. Each pair of sub-shares, $(f_j(x_i), g_j(x_i))$ received from other user, U_j, can be verified by user, U_i, by checking whether $\sigma(f_j(x_i), g_j(x_i)) \overset{?}{=} \prod_{l=0}^{t-1} \sigma_l^{j \cdot x_i^l} \bmod p$. If it passes the verification, the pair of sub-shares are verified; otherwise, user, U_i, requests user, U_j, to re-generates a new pair of sub-shares.

Phase 2: Open-commitment phase

Step1. Each user, U_i, releases his pair of sub-secrets, (s_i, k_i), to all other users.

Step 2. Each pair of sub-secrets, (s_i, k_i), of user, U_i, can be verified by checking whether $\sigma_0^j \overset{?}{=} g^{s_i} h^{k_i} \bmod p$. If the sub-secrets cannot be verified successfully, all other users should try to recover the sub-secret, s_i, using their sub-shares, $f_i(x_j)$ $j = 1, 2, \ldots, n$, $j \neq i$. Note that each sub-share $f_i(x_j)$ received from user, U_i, can also be verified following Step 4, in Phase 1. For example, if sub-shares, $f_i(x_j)$, $j = 1, 2, \cdots, l$, $l \geq t$ have been verified, the sub-secret, S_i, can be computed using the Lagrange interpolation formula as $s_i = \sum_{j=1}^{l} f_i(x_j) \prod_{k=1, k\neq j}^{l} \frac{-x_k}{x_j - x_k} \bmod p$.

Step 3. After obtaining all sub-secrets, s_i, $i = 1, 2, \ldots, n$ each user computes $\rho \leftarrow \bigoplus_{j=1}^{n} s_j$ and evenly divides ρ into n strings of bit sequences, $\rho_1, \rho_2, \ldots, \rho_n$. U_i sorts these n strings in a descending order. Let ρ_i denote the o_i-th smallest string of bit sequences in the set of strings. Then U_i obtains his order o_i and the orders of other users. Finally, U_i outputs the order $(o_1, o_2, \cdots, o_n)$.

Figure 4: Concrete instantiation.

each phase, each user needs to compute and release values to others. But there has no interaction among users to verify shares and to determine the output. In the first phase, each user needs to execute $2t$ modular exponentiations to compute his commitments and $2(t+1)$ modular exponentiations to verify each pair of subshares. Overall, each user needs $2(n-1)(t+1)$ modular exponentiations to verify all subshares from other users. In the second phase, each user needs two modular exponentiations to verify each subsecret of other users. Overall, each user needs $2(n-1)$ modular exponentiations to verify all subsecrets from other users.

In summary, we list the features of our proposed protocol as follows:

(1) The output of the protocol depends on inputs generated by all users and is uniformly distributed.

(2) Subshares generated by the owner of the subsecret can be verified by other users using the commitments of the subsecret.

(3) Once subshares are successfully verified in the first phase, the user can no longer deny his subsecret of the commitment. If the user tries to deny his subsecret, the subsecret can be recovered by honest users

and is used in the evaluation of the output in the second phase.

(4) The protocol can resist colluded users working together to attack the security of our protocol.

6. Conclusion

We propose a novel cryptographic primitive, called multiparty undeniable drawing straws (MUDS), that allows n users to work together to determine the order of users. MUDS is very useful in multiparty applications since it provides a fair way of determining an order of users. MUDS can also prevent cheaters from taking advantage of honest users by releasing their values last. Our proposal is more efficient and secure than the state-of-the-art cryptographic solutions, so it is absolutely attractive for multiparty applications in RIS towards 5G.

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.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grants nos. 61772224, 62172181, and 62072133), the National Natural Science Foundation of China (Grants nos. U21A20465, 61922045, and U1836115), and the key projects of Guangxi Natural Science Foundation (no. 2018GXNSFDA281040).

References

[1] G. Hernandez, O. Arias, D. Buentello, and Y. Jin, *Smart Nest Thermostat—A Smart Spy in Your home*University of Central Florida, Orlando, FL, USA, 2014.

[2] A. G. Illera and J. V. Vidal, *Lights off! the Darkness of the Smart Meters,* BlackHat, Europe, 2014.

[3] M. Kabay, *Attacks on Power Systems: Hackers*, Malware, Santa Clara, CA, USA, 2010.

[4] K. Koscher, A. Czeskis, F. Roesner et al., "Experimental security analysis of a modern automobile," in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 2010.

[5] B. Miller and D. Rowe, "A survey SCADA of and critical infrastructure incidents," in *Proceedings of the Research in Information Technology (RIIT)*, Calgary, Alberta, Canada, October 2012.

[6] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," in *Whitepaper,* Black Hat, USA, 2014.

[7] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.

[8] J. Vijayan, "Stuxnet renews power grid security concerns," *Computerworld*, vol. 26, 2010.

[9] J. Xiong, R. Bi, Y. Tian, X. Liu, and D. Wu, "Toward lightweight, privacy-preserving cooperative object classification for connected autonomous vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2787–2801, 2022.

[10] D. M. Nicol, "Hacking the lights out," *Scientificfic American*, vol. 305, no. 1, pp. 70–75, 2011.

[11] A. Soullie, "Industrial control systems: pentesting PLCs 101," in *Whitepaper,* Black Hat, Europe, 2014.

[12] A. R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and Privacy Challenges in Industrial Internet of Things," in *Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, San Francisco, CA, USA, June 2015.

[13] M. Blum, "Coin flipping by telephone a protocol for solving impossible problems," *ACM SIGACT News*, vol. 15, no. 1, pp. 23–27, 1983.

[14] D. Aharonov, A. Ta-Shma, U. V. Vazirani, and A. C. Yao, "Quantum bit escrow," in *Proceedings of the STOC'00: Thirty-Second Annual ACM Symposium on Theory of Computing*, pp. 705–714, New York, NY, USA, May 2000.

[15] A. Ambainis, "A new protocol and lower bounds for quantum coin flipping," *Thirtieth Annual ACM Symposium on Theory of Computing*, vol. 68, no. 2, pp. 398–416, 2004.

[16] R. W. Spekkens and T. Rudolph, "Degrees of concealment and bindingness in quantum bit commitment protocols," *Physical Review A*, vol. 65, Article ID 012310, 2001.

[17] A. Nayak and P. Shor, "Bit-commitment-based quantum coin flipping," *Physical Review A*, vol. 67, no. 1, Article ID 012304, 2003.

[18] T. Moran, M. Naor, and G. Segev, "An optimally fair coin toss," *Theory of Cryptography Lecture Notes in Computer Science*, vol. 5444, pp. 1–18, 2009.

[19] A. Beimel, E. Omri, and I. Orlov, "Protocols for multiparty coin toss with dishonest majority," *Advances in Cryptology*, vol. 6223, pp. 538–557, 2010.

[20] I. Haitner and E. Tsfadia, "An almost-optimally fair three-party coin-flipping protocol," *SIAM Journal on Computing*, vol. 46, no. 2, pp. 408–416, 2014.

[21] A. Ambainis, H. Buhrman, Y. Dodis, and H. Röhrig, "Multiparty quantum coin flipping," in *Proceedings of the 19th IEEE Annual Conference on Computational Complexity*, pp. 250–259, Amherst, MA, USA, June2004.

[22] W. Liu, S.-S. Luo, and P. Chen, "A study of secure multi-party ranking problem," in *Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence Networking and Parallel/Distributed Computing*, pp. 727–732, Qingdao, China, August 2007.

[23] A. C. Yao, "How to generate and exchange secrets," in *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pp. 218–229, Toronto, Canada, October 1986.

[24] C. Cheng, Y.-L. Luo, C.-X. Chen, and X.-K. Zhao, "Research on Secure Multi-Party Ranking Problem and Secure Selection Problem," in *Proceedings of the International Conference on Web Information Systems and Mining (WISM)*, Sanya, China, October 2010.

[25] J. Halpern and V. Teague, "Rational secret sharing and multiparty computation: extended abstract," in *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing - STOC'04*, pp. 623–632, New York, NY, USA, June 2004.

[26] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[27] G. Fuchsbauer, J. Katz, and D. Naccache, "Efficient Rational Secret Sharing in Standard Communication Networksficient

rational secret sharing in standard communication networks," in *Proceedings of the 7th Theory Of Cryptography Conference-TCC'10, LNCS 5978*, pp. 419–436, Berlin, Germany, February 2010.

[28] S. J. Ong, D. C. Parkes, A. Rosen, and S. P. Vadhan, "Fairness with an honest minority and a rational majority," in *Proceedings of the 6th Theory of Cryptography Conference-TCC'09, LNCS 5444*, pp. 419–436, San Francisco, CA, USA, March 2009.

[29] C. Tartary, H. Wang, and Y. Zhang, "An efficient and information theoretically secure rational secret sharing scheme based on symmetric bivariate polynomials," *International Journal of Foundations of Computer Science*, vol. 22, no. 6, pp. 1395–1416, 2011.

[30] M. Tompa and H. Woll, "How to share a secret with cheaters," *Journal of Cryptology*, vol. 1, no. 3, pp. 133–138, 1988.

[31] W. K. Moses Jr., and C. Pandu Rangan, "Rational secret sharing over an asynchronous broadcast channel with information theoretic security," *International Journal of Network Security & its Applications*, vol. 3, no. 6, pp. 1–18, 2011.

[32] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proceedings of the 6th IEEE Symposium on Foundations of Computer Science*, pp. 427–437, Los Angeles, CA, USA, October 1987.

[33] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology—CRYPTO'91,* Springer-Verlag, Berlin, Germany, 1992.

[34] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults," in *Proceedings of the 26th Annual Symposium on Foundations of Computer Science (SFCS 1985)*, pp. 383–395, Portland, OR, USA, October 1985.

[35] J. C. Benaloh, "Secret sharing homomorphisms: keeping shares of a secret," *Advances in Cryptology*, vol. 263, pp. 251–260, 1987.

[36] M. Stadler, "Publicly verifiable secret sharing," *Advances in Cryptology*, vol. 3, pp. 190–199, 1996.

[37] B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic voting," in *Proceedings of the Advances in Cryptology-CRYPTO'99, LNCS 1666*, pp. 148–164, Santa Barbara, CA, USA, August 1999.

[38] A. Peng and L. Wang, "One publicly verifiable secret sharing scheme based on linear code," in *Proceedings of the 2nd Conference on Environmental Science and Information Application Technology*, pp. 260–262, Wuhan, China, July 2010.

[39] A. Ruiz and J. L. Villar, "Publicly verifiable secret sharing from Paillier's cryptosystem," in *Proceedings of the WEWoRC'05, LNI P-74*, pp. 98–108, Leuven, Belgium, January2005.

[40] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the Advances in Cryptology-EUROCRYPT'99, LNCS 1592*, pp. 223–238, Prague, Czech Republic, May 1999.

[41] Y. Tian, C. Peng, and J. Ma, "Publicly verifiable secret sharing schemes using Bilinear pairings," *International Journal on Network Security*, vol. 14, no. 3, pp. 142–148, 2012.

[42] T.-Y. Wu and Y.-M. Tseng, "A pairing-based publicly verifiable secret sharing scheme," *Journal of Systems Science and Complexity*, vol. 24, no. 1, pp. 186–194, 2011.

[43] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 8696558, 17 pages
https://doi.org/10.1155/2022/8696558

Research Article

Improved Vessel Trajectory Prediction Model Based on Stacked-BiGRUs

Yang Xu,[1,2] Jilin Zhang,[2,3] Yongjian Ren,[1,2] Yan Zeng,[1,2] Junfeng Yuan,[1,2] Zhen Liu,[1,2] Lei Wang,[1,2] and Dongyang Ou[1,2]

[1] *School of Computer Science and Technology, Hangzhou Dianzi University, HangZhou 310018, China*
[2] *Key Laboratory of Complex Systems Modeling and Simulation, Hangzhou Dianzi University, HangZhou 310018, China*
[3] *School of Cyber Security, Hangzhou Dianzi University, HangZhou 310018, China*

Correspondence should be addressed to Jilin Zhang; jilin.zhang@hdu.edu.cn

Received 28 February 2022; Revised 3 April 2022; Accepted 6 April 2022; Published 13 May 2022

Academic Editor: Jinbo Xiong

An intelligent maritime navigation system is expected to play an important role in the realm of Internet of Vessels (IoV). As a key technology in navigation systems, vessel trajectory prediction technology is critical to the IoV. Automatic identification system (AIS), an automated tracking system, is used extensively for vessel trajectory prediction. However, certain characteristics in the AIS data, such as the large number of anchored trajectories in the area, anomalous sharp turns of some trajectories, and the behavioral differences of vessels in different segments, limit the prediction accuracy. In this study, we propose a novel vessel trajectory prediction model for accurate prediction with the following characteristics: (1) an anchor trajectory elimination algorithm to eliminate anchor trajectories; (2) a statistical trajectory restoration algorithm to repair sharp turning; (3) a two-stage clustering algorithm (D-KMEANS) to distinguish vessel behavior; and (4) a deep bidirectional gate recurrent unit (Stacked-BiGRUs) model to predict vessel trajectory and compare the accuracy of the model before and after improvement. The results show that the mean square error and the mean absolute error of the improved model are reduced by 27% and 46%, respectively. This research shows good potential for maritime navigation early warning and safety.

1. Introduction

As an extension of Internet technology, the Internet of things (IoT) takes advantage of communication sensing technology to realize the information exchange between things [1, 2]. Automatic driving technology benefits from the rapid development of the IoT [3], and its functions such as intelligent collision avoidance and collaborative control [4] are becoming increasingly mature. Internet of Vessels (IoV) provides vessel sensing and traffic information service in the whole drainage area. IoV exchanges large volumes of data among vessels and base stations, such as course, speed, and location. Therefore, IoV has the ability to provide intelligent navigation, a safer collision avoidance decision, and an efficient port area management by realizing the refinement of vessel trajectory prediction. However, in offshore ports with high vessel density and complex traffic, it very challenging to predict the trajectory of vessels. Unlike vehicle or pedestrian trajectory prediction, moving objects in a maritime environment are not restricted by geometric structure and their movement patterns are more complex than those of land vehicles. According to the "International Regulations for Preventing Collisions at Sea" (COLREGS), the navigation rules of vessels rely substantially on experience, which is difficult to quantitatively analyze. Furthermore, the historical Automatic Identification System (AIS) data contain the potential movement patterns of vessels, such as usual behavior in some areas or periodic entry and exit of a channel. However, the current methods rarely eliminate anchor trajectory, repair abnormal AIS data, and classify the behavior of vessels before predicting. The following features in the raw AIS data reduce the prediction accuracy:

(1) There are irregular anchor trajectories in the raw data. The vessel in the anchored state will float with the wind and waves, producing irregular trajectories in a small range. The vessel at anchor can be regarded as a static obstacle, which will mislead the model training and reduce the prediction accuracy. Therefore, eliminating anchor trajectory is of great significance to improve the accuracy of trajectory prediction.

(2) There are acute bends caused by abnormal points in the raw data. The cause of the abnormal point is that the vessel urgently avoids obstacles or the marine equipment sends wrong data. The purpose of model training is to learn the usual behaviors of the vessel, but the abnormal point will reduce the convergence speed of the model. Therefore, it is necessary to design an algorithm to repair abnormal points.

(3) There are different behaviors in vessel navigation, such as setting sail, crossing the waterway, and working. Mixing these low similarity trajectories will reduce the accuracy of prediction. Therefore, classifying vessel behaviors plays an important role in improving prediction accuracy.

Recurrent neural networks (RNNs) can explore the inherent laws from AIS data and have superior generalization ability. In this study, we proposed an improved vessel trajectory prediction model based on Stacked-BiGRUs. The main contributions are as follows:

(1) We proposed an anchor trajectory elimination algorithm to eliminate anchor trajectories. The anchor trajectory is identified by the speed characteristics of vessel berthing and setting sail.

(2) We designed a statistical trajectory restoration algorithm to repair outliers. The outliers are repaired based on the probability distribution of the latitude and longitude changes in the trajectory.

(3) We proposed a two-stage trajectory clustering method (D-KMEANS) to classify the vessel behaviors. The trajectories are classified by the DBSCAN and KMeans to extract behavior sets.

(4) We built a Stacked-BiGRUs model. Compared with other recurrent neural networks, the bidirectional structure gained additional feature extraction, which effectively improved the prediction accuracy.

2. Related Work

2.1. Trajectory Restoration. Under ocean environment conditions, vessel trajectory data are prone to inaccuracies due to equipment abnormalities, wind, and waves. Therefore, data repair technology is required to eliminate these inaccuracies. The technology is mainly divided into constraint-based trajectory restoration and machine learning method-based trajectory restoration.

For the restoration method based on constraints, Song et al. [5] first proposed a data cleaning method based on speed constraints, considering the limitation of the speed of data change. They achieved good results in a series of time series data restoration experiments. Tu et al. [6] used an improved RDP algorithm to address acute bends and self-intersections in trajectory data, which improved the accuracy of trajectory prediction. Li et al. [7] used an improved A^* shortest path algorithm to fully consider road network topology and historical matching points and proposed a new trajectory restoration algorithm. Gao et al. [8] used a dynamic programming method to set multiple intervals for sequence data, and searched for candidate repair points in an iterative manner, avoiding excessive repair of sequence data.

For the repair method based on machine learning, Kanarachos et al. [9] combined wavelet, neural network, and Hilbert transform to propose a new time series anomaly detection algorithm. Cheng et al. [10] proposed a trajectory restoration algorithm based on bidirectional LSTM, which had a good effect on trajectory restoration in curved waterways. Xue et al. [11] proposed a fractional gradient RBF neural network that drives momentum. The training error of this algorithm was lower than that of gradient descent, stochastic gradient descent, and momentum gradient descent.

2.2. Trajectory Clustering. As an important spatiotemporal object data type, vessel trajectories record the behavior characteristics of vessels. The trajectory behavior category can be divided using the clustering method. The method of vessel trajectory clustering, which is categorized based on distance, density, graph, and statistics, is summarized as follows:

For distance-based clustering methods, Mao et al. [12] proposed an incremental clustering algorithm, OCLUST, for the online processing of trajectory stream data, which achieved superior performance in clustering streaming trajectories. Xiong et al. [13] proposed a privacy and availability data clustering scheme (PADC) based on KMeans to enhance the selection of the initial center point.

For density-based clustering methods, Liu et al. [14] applied the extended density-based spatial clustering of applications with noise (DBSCAN) algorithm to correlate International Maritime Organization (IMO) rules with vessel trajectories for cluster analysis. Sun et al. [15] proposed a clustering method based on the minimum boundary matrix and the similarity of the buffer zone, and applied the DBSCAN algorithm twice to improve the accuracy of trajectory clustering. Han et al. [16] proposed an enhanced spatial clustering method based on density, which ensured the accuracy of the behavior recognition result by including additional geospatial information based on vessel speed and direction.

For graph clustering methods, Tian et al. [17] present a graph clustering privacy-preserving method that improves the security of private information. Budimirovic et al. [18] present a novel graph clustering method (IBC1/IBC2) to cluster human behaviors, and the method has reference significance in vessel behavior clustering.

For statistics-based clustering methods, Wen et al. [19] applied the improved algorithm PrefixSpan for sequential pattern mining to vessel trajectory pattern mining, defined vessel principal vectors, cross sections, and boundaries, and identified vessel trajectories with similar motion patterns through pruning strategies. Peel et al. [20] described the activities of vessels as the four states of anchoring, sailing, entering/exiting ports, and trawling. Hidden Markov models were used to identify the laws of vessel activities and cluster the different states of vessels. Riveiro et al. [21] used kernel density estimation (KDE) to cluster vessel trajectories by selecting a suitable kernel function and window width and using observations to characterize the overall vessel motion pattern.

2.3. Trajectory Prediction. There have been several studies on vessel trajectory prediction methods. These methods are mainly based on dynamic model analysis, statistics, and machine learning.

2.3.1. Trajectory Prediction Based on Dynamic Model Analysis. The Kalman filter is a classic method in the field of linear system analysis. Several scholars have proposed various trajectory prediction methods based on the Kalman filter. Jaskolsk et al. [22] used the Discrete Kalman filter (KF) algorithm to improve the possibilities of vessel motion trajectory and monitoring in the TSS (Traffic Separation Scheme) and fairways area. Qiao et al. [23] proposed a dynamic trajectory prediction method based on Kalman filtering that used the estimated value at the previous moment and the observation value at the current moment to update the estimation of state variables, and subsequently predict the position of the vessel at the next moment.

2.3.2. Trajectory Prediction Based on Statistical Models. A Bayesian network is a probabilistic graph model that comprises a directed acyclic graph composed of nodes representing variables and directed edges connecting these nodes. By combining empirical knowledge and prior information, posterior information was obtained. Mazzarea et al. [24] proposed the Bayesian vessel position prediction algorithm KB-PF based on particle filters.

The Markov model is a statistical model that can predict the trend of data changes at equal time intervals in the future based on historical data. Tong et al. [25] used Markov chain- and gray prediction-related methods to propose a hidden Markov model based on the adaptive update parameters of environmental data captured by dynamic objects. It showed high accuracy in curve prediction. Qiao et al. [26] developed a trajectory prediction algorithm, PutMode, based on Continuous Time Bayesian Networks (CTBNs), and the experimental results showed that PutMode could predict the possible motion curves of objects in a more accurate and efficient manner.

The Gaussian process is a stochastic process mainly used to solve regression problems. Rong et al. [27] proposed a probabilistic trajectory prediction model, which decomposed vessel motion into horizontal and vertical predictions. In the horizontal direction, a Gaussian process was used to model the uncertainty of horizontal motion, and the vertical direction was estimated through acceleration. Anderson et al. [28] regarded the trajectory as a one-dimensional Gaussian process. They calculated the posterior distribution of the predicted value by obtaining the joint prior density and covariance matrix of the observed value and the predicted value. Qiao et al. [29] proposed the Gaussian mixture model-based trajectory prediction method (GMTP), which used a Gaussian mixture model to model complex motion modes and calculated the probability distribution of different motion modes. Subsequently, the Gaussian process regression was used to predict the plausible motion trajectory of a moving object. Dalsnes et al. [30] proposed the Gaussian mixture model (GMM), which provided a measure of the uncertainty of the prediction results and addressed multiple modalities.

2.3.3. Trajectory Prediction Based on Machine Learning. Extreme learning machines (ELMs) are a single hidden layer feedforward neural network. Mao et al. [31] proposed an ELM-based trajectory prediction algorithm to predict the trajectory of vessels. The algorithm did not require the weights and biases of an iterative neural network; thus, its training speed was faster.

An autoencoder (AE) is an unsupervised neural network model that includes encoding and decoding. Inspired by the generative model, Murray et al. [32] proposed a bilinear autoencoder method to iteratively predict the future state and then generate the entire vessel trajectory. The model could estimate the distribution of future trajectories of vessels and quantify the uncertainty in predicting vessel positions.

A long short-term memory (LSTM) solves the long-term dependence of RNNs. Gao et al. [33] proposed a method that combines the advantages of LSTM and TPNet. The proposed method was not only easy to implement and suitable for real-time analysis, but also presented a high prediction accuracy. Nguyen et al. [34] proposed a scalable sequence-to-sequence learning model combined with LSTM. Chen et al. [35] combined the advantages of LSTM, support vector machine (SVM), and extreme value optimization algorithms and avoided the weak generalization ability and robustness of a single deep learning method. Suo et al. [36] compared the accuracy and training efficiency of gated recurrent unit (GRU) and LSTM in vessel trajectory prediction. Xiao et al. [37] proposed a two-step LSTM, the unidirectional and bidirectional LSTM (UB-LSTM), combined with behavior recognition for vehicle trajectory prediction. Zhang et al. [38] proposed a multiscale convolutional neural network (MSCNN)-based high-frequency (HF) radar vessel trajectory prediction method to predict the trajectory hidden in the clutter. Jaseena et al. [39] combined the wavelet transform and the bidirectional LSTM and proposed the EWT-LSTM model to forecast wind. Xue et al. [40] proposed social-scene-LSTM for pedestrian trajectory prediction, which was a novel hierarchical LSTM-based network. It considered the social neighborhood and scene composition and employed three different LSTMs to capture people, society, and scene scale information. The accuracy of pedestrian trajectory prediction was significantly improved.

3. Trajectory Prediction Model

This section introduces the trajectory prediction model. As shown in Figure 1, the model was divided into the four parts, namely anchor trajectory elimination, outlier repair, classification of vessel behavior, and trajectory prediction. We proposed an anchor trajectory elimination algorithm and a statistical trajectory restoration algorithm to improve trajectory quality. In the classification of vessel behavior, we designed a two-stage trajectory clustering algorithm (D-KMEANS) to extract the main navigation modes of vessels. Finally, in trajectory prediction, we trained the Stacked-BiGRUs model and use sliding window to predict vessel trajectory.

3.1. Anchor Trajectory Elimination Algorithm. Some anchored trajectories existed during the sailing cycle of vessels. As shown in Figure 2, these trajectories generally appeared as overlapping points at the same position or irregular clumps formed by reciprocating motion in a small area. We proposed an anchor trajectory elimination algorithm to eliminate the anchor trajectory.

Dividing the trajectories of different vessels according to the MMSI number, if the total number of vessels is Ms, get the trajectory set $\text{Traj} = \{\text{Traj}_1, \text{Traj}_2, \ldots, \text{Traj}_M\}$; Traj_M is the set of trajectory points of the m th vessel, eliminating the anchor trajectory Traj_m for each vessel.

The very high-frequency (VHF) transceiver automatically broadcasted the vessel's kinematic information (vessel position, speed, heading, etc.) and static information (vessel name, vessel unique identifier, message serial number, vessel type, vessel size, current time, etc.) [41] in the form of AIS messages. We defined the trajectory of the vessel in article m as $\text{Traj}_m = \{p_i(MMSI, t, \text{lon}, \text{lat}, \text{Sog}) | i = 1, 2, \ldots, N\}$, and p_i represented the locus point at time i on Traj_m, which included the marine mobile service identification (MMSI), timestamp (t), longitude (lon), latitude (lat), and speed over the ground (Sog).

Based on the above symbols, the anchor trajectory elimination process of Traj_m is shown in Algorithm 1. The specific process of the algorithm is as follows:

(1) Every point of Traj_m was traversed. When the Sog at a certain point p_i was less than Sog_0, the next T_s points were continuously judged. When the Sog of all points was less than Sog_0, the point p_i was marked as an anchoring point and the sailing point was located. Otherwise, the detection of the anchoring point was continued until the end of the trajectory traversal.

(2) If the anchor point of the vessel is detected, continue to detect whether there is a trajectory point P_k after the trajectory point P_{i+T_s}, of which the Sog is higher than the sailing speed threshold Sog_1. If yes, continue to check whether the consecutive Sog of T_s points following the trajectory point P_k is higher than the sailing speed threshold Sog_1. If yes, determine the trajectory point P_k as the point where the anchor is weighed, and delete the trajectory between the anchor point and the point where the anchor is weighed; otherwise, repeat step (2) to detect the anchor point until the loop is over.

(3) Return to step (1) and continue to detect until the end of the trajectory traversal.

3.2. Statistical Trajectory Restoration Algorithm. The trajectories that have undergone anchor trajectory elimination still include some abnormal points, resulting in abnormal movement patterns. To repair these outliers, a statistical trajectory restoration algorithm is used.

Each vessel trajectory is split into longitude and latitude sequences, which are marked as $S_\beta = \{\beta_1, \beta_2, \ldots, \beta_n\}$. The longitude and latitude sequences then both receive anomaly repairs. In the S_β, β = lon (longitude) or *lat* (latitude). S_{lon} and S_{lat}, respectively, represent the sequence of all longitudes and latitudes of a trajectory. The acceleration of a trajectory point β_i is a_i, which is calculated by the formula:

$$\begin{aligned} a_i &= \frac{V\beta_{i,i+1} - V\beta_{i-1,i}}{t_{i+1} - t_{i-1}}, \\ V\beta_{i,i+1} &= \frac{\beta_{i+1} - \beta_i}{t_{i+1} - t_i}, \\ V\beta_{i-1,i} &= \frac{\beta_i - \beta_{i-1}}{t_i - t_{i-1}}, \\ i &= 2, 3, \ldots, n-1. \end{aligned} \tag{1}$$

The specific steps of probabilistic trajectory anomaly repair are as follows:

(1) The acceleration sequence $S_a = \{a_1, a_2, \ldots, a_n\}$ of S_β is calculated from formula 1. S_a is used to establish the table of acceleration probability distribution P_a. The schematic diagram of P_a is shown in Figure 3. Consider the number of intervals is $n-2$ and the interval size is $\delta = \max(S_a) - \min(S_a)/n - 2$; the probability value of each interval equals the ratio between the number of trajectory points whose accelerations fall within the interval and the total number of trajectory points.

(2) Initialization $W_i = 0$, $i = 2, 3, \ldots, n-1$. The sequence is windowed (size: 3, step: 1). The subsequence under each window is $S_{\beta_{i-1,i,i+1}} = \{\beta_{i-1}, \beta_i, \beta_{i+1}\}$, $i = 2, 3, \ldots, n-1$.

(3) Determine whether the current i is equal to $n-1$. If yes, record the post-repair sequence as S'_β. If no, update $i = i + 1$ and then proceed to step (4).

(4) Repair trajectory points under the i-th window.

(a) Make $W_{i+1} = W_i$;

(b) Build a repair value array for the trajectory points $\{\beta_b, \beta_b - \varepsilon_{\max,\beta} + u, \ldots, \beta_b + \varepsilon_{\max,\beta}\}$, where $b = i-1, i, i+1$. The elements in the array are sorted in ascending order. $\varepsilon_{\max,\beta}$ and u are the

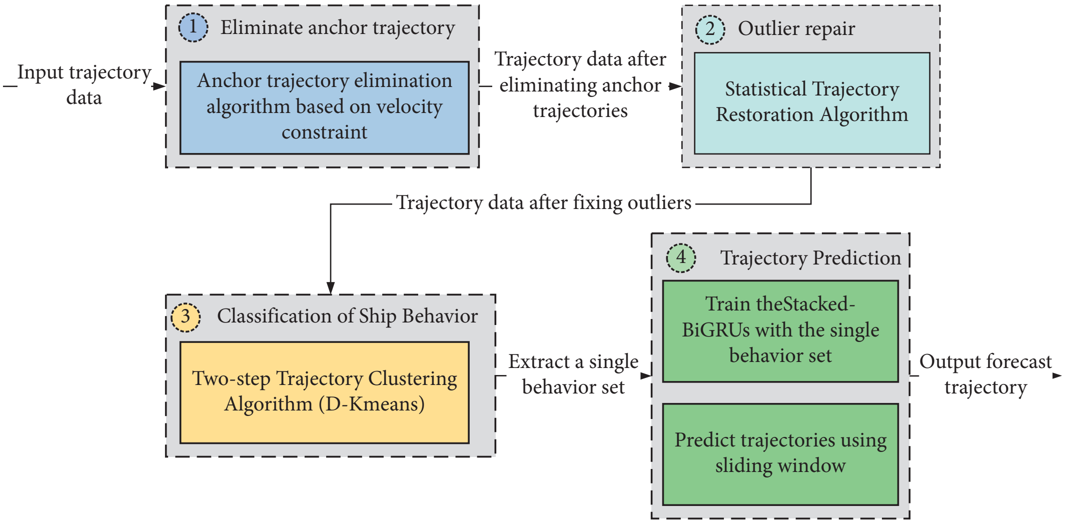

Input trajectory data
1 Eliminate anchor trajectory
Anchor trajectory elimination algorithm based on velocity constraint
Trajectory data after eliminating anchor trajectories
2 Outlier repair
Statistical Trajectory Restoration Algorithm
Trajectory data after fixing outliers
3 Classification of Ship Behavior
Two-step Trajectory Clustering Algorithm (D-Kmeans)
Extract a single behavior set
4 Trajectory Prediction
Train theStacked-BiGRUs with the single behavior set
Predict trajectories using sliding window
Output forecast trajectory

Figure 1: Improved vessel trajectory prediction model.

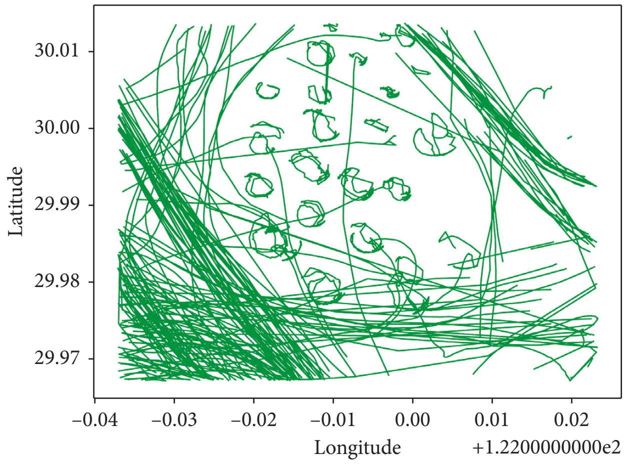


Figure 2: A marine chart of anchor trajectories.

```
Input: trajectory to be processed $\mathrm{Traj}_m$, number of trajectory point $n$, anchoring speed threshold $\mathrm{Sog}_0$, sailing speed threshold $\mathrm{Sog}_1$, Detection length $T_s$
Output: processed trajectory $\mathrm{Traj}_m$
Traj_m = $[P_1, P_2, \ldots, P_n]$
For $P_i$ in $\mathrm{Traj}_m$ do//Start identifying anchoring point
    If $P_i \cdot \mathrm{Sog} < \mathrm{Sog}_0$ and all $P_j$ in $[P_i.t, P_i.t + T_s]$ meet $P_j \cdot \mathrm{Sog} < \mathrm{Sog}_0$
    Mark $P_i$ as an anchor point
    For $P_k$ in $[P_i.t + T_s, P_n.t]$ do//Start identifying sailing point
        If $P_k \cdot \mathrm{Sog} > \mathrm{Sog}_1$ and all $P_l$ in $[P_k.t + T_s + 1, P_k.t + 2T_s + 1]$ meet $P_l \cdot \mathrm{Sog} > \mathrm{Sog}_1$
        Mark $P_k$ as an sailing point
        Delete the trajectory between $P_i$ and $P_k$ //Eliminate anchor trajectory
        Break
    End If
End For
    End If
End For
Return $\mathrm{Traj}_m$
```

Algorithm 1: Anchor Trajectory Elimination.

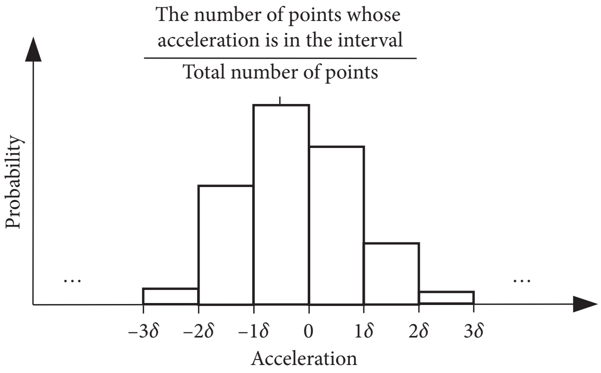


Figure 3: Schematic of the acceleration probability distribution of trajectory points.

```
Input: Sequence to be repaired $S_\beta = \{\beta_1, \beta_2, \ldots, \beta_n\}$, maximum repair range $\varepsilon_{\max,\beta}$, step length of each repair $u$
Output: repaired sequence $S'_\beta = \{\beta'_1, \beta'_2, \ldots, \beta'_n\}$
Compute $S_a = \{a_2, a_3, \ldots, a_{n-1}\}$ according to equation (1)
Create a Probability distribution table of acceleration $P_a$ according to step 1
Initialize $W_i$, $i = 2, 3, \ldots, n-1$
For $i = 2$ to $n-1$ do//Start repairing
  $S_{\beta_{i-1,i,i+1}} = \{\beta_{i-1}, \beta_i, \beta_{i+1}\}$ //Windowing the subsequence (size: 3, step: 1)
  For $\beta'_{i+1}$ in $\{\beta_{i+1} - \varepsilon_{\max,\beta}, \beta_{i+1} - \varepsilon_{\max,\beta} + u, \ldots, \beta_{i+1} + \varepsilon_{\max,\beta}\}$
    For $\beta'_i$ in $\{\beta_i - \varepsilon_{\max,\beta}, \beta_i - \varepsilon_{\max,\beta} + u, \ldots, \beta_i + \varepsilon_{\max,\beta}\}$
    For $\beta'_{i-1}$ in $\{\beta_{i-1} - \varepsilon_{\max,\beta}, \beta_{i-1} - \varepsilon_{\max,\beta} + u, \ldots, \beta_{i-1} + \varepsilon_{\max,\beta}\}$
      Compute $a_i$ according to equation (2)
      Read $P(a'_i)$ from $P_a$
      If $W_{i+2} < W_{i+1} + P(a'_i)$ //If the probability goes up
        $S_{\beta_{i-1,i,i+1}}{}' = \{\beta'_{i-1}, \beta'_i, \beta'_{i+1}\}$//Update the subsequence
        $W_{i+2} = W_{i+1} + P(a'_i)$//Update the acceleration sequence
        Update $S_a$, $a_i = a'_i$ //Update the probability
        Update $P_a$ //Update the table of acceleration probability distribution
      End If
    End For
  End For
End For
End For
Return $S'_\beta = \{\beta'_1, \beta'_2, \ldots, \beta'_n\}$
```

Algorithm 2: Statistical Trajectory Restoration.

maximum repair range and step length of each repair, respectively. Traverse the candidate repair value array and then attempt to repair the trajectory point using the candidate repair values. Calculate the post-repair acceleration according to Formula (2) and then obtain the probability from the acceleration probability distribution table P_a. If $W_{i+1} < W_i + P(a'_i)$, replace with $\{\beta'_{i-1}, \beta'_i, \beta'_{i+1}\}$ and update the probability value $W_{i+1} = W_i + P(a'_i)$.

(c) Determine whether b is equal to $i+1$. If yes, skip to step (3). If no, update $b = b + 1$ and then return to step b).

$$\begin{aligned} a'_i &= \frac{V\beta'_{i,i+1} - V\beta'_{i-1,i}}{t_{i+1} - t_{i-1}}, \\ V\beta'_{i,i+1} &= \frac{\beta'_{i+1} - \beta'_i}{t_{i+1} - t_i}, \\ V\beta'_{i-1,i} &= \frac{\beta'_i - \beta'_{i-1}}{t_i - t_{i-1}}, \\ i &= 2, 3, \ldots, n-1. \end{aligned} \tag{2}$$

The specific flow of the algorithm is shown in Algorithm 2:

3.3. Ship Behavior Classification Algorithm. Vessels have different or even conflicting navigation behaviors in the voyage cycle. For example, when the vessel starts sailing, the trajectory is in one direction, and the shape of the trajectory is short and dense. When the vessel goes to the target location at high speed, the trajectory is characterized by long distance, less turning, and smoothness. When the vessel reaches the target location, the trajectory is characterized by periodic repeated folding. Mixing different and conflicting trajectory features is not conducive to improving the accuracy of prediction. After obtaining the trajectory repaired in the previous section, this section mainly introduces the vessel behavior classification algorithm based on D-KMeans (DBSCAN-KMeans), which is used to distinguish different behavior patterns. The behavior sets are used for model training.

Ship locations are considered as spatial data; similar vessel behaviors can be given as clusters with enough spatial proximity. From the characteristics of vessel behaviors, we found that the DBSCAN meets the requirement of extracting the behavior trajectories. In the DBSCAN, it is necessary to specify two parameters, MinPts and ϵ, which are the smallest number of vessels in a cluster and the sailing radius to a behavioral cluster. When vessels are sailing, a distance between vessels is typically calculated by the Mercator method. The distance unit of the Mercator method is sea mile. When the DBSCAN is applied to oceanographic data such as AIS data, the Mercator method is more accurate than the Euclid method to calculate the distance between two data points. Moreover, the time complexity of the Mercator method is similar to the Euclid method. Therefore, it is more reasonable to adopt the Mercator method in vessel trajectory clustering.

After clustering by DBSCAN, a large number of clusters $C = (C_1, C_2, \ldots, C_m)$ are generated. To merge these clusters into three vessel behaviors, KMeans is required. Because the points belonging to the same behavior have similar speed, KMeans is expected to cluster the average speed set $V = \{\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(m)}\}$; $\mu^{(i)}$ is the average speed of points in C_i. In KMeans, the data are divided into k clusters, setting the k value of KMeans to 3; by calculating the average speed of each cluster in the result of the previous step, and merging the first-step clusters with similar average speed, three types of vessel behaviors were obtained.

The algorithm is shown in Algorithm 3. The D-KMeans flow is described below:

(1) The DBSCAN algorithm is used to cluster the vessel trajectory points that received outliers repair in section 3.3 to obtain the first-step clustering result.

(2) The average speed set of each cluster $V = (\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(m)})$ is calculated from the first-step clustering result $C = \{C_1, C_2, \ldots, C_m\}$ in step (1).

(3) With $k = 3$, the KMeans algorithm is used for the second-step clustering of average speed set $V = \{\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(m)}\}$. This is to obtain the three behaviors of the vessel, including setting sail, crossing waterway, and working.

3.4. Stacked-BiGRUs Model. After obtaining the vessel behavior set in the previous section, we used the behavior set to train the Stacked-BiGRUs model. As shown in Figure 4, the Stacked-BiGRUs model includes an input layer, three BiGRU units, and a dense layer.

The trajectory data are vectorized, and the trajectory points of several consecutive time steps are used as an input trajectory $l = [p_1, p_2, \ldots, p_n]$.

To ensure dimensionless interference, the trajectory data were standardized before being used as the input trajectory of the model. The z-score standardization method was used to process the longitude and latitude in the trajectory data separately. As shown in Equation (3), l is the input trajectory, u is the mean of the series, σ is the standard deviation of the series, and $l = [p_1', p_2', \ldots, p_n']$ is the normalized input trajectory.

$$l' = \frac{l - u}{\sigma}. \tag{3}$$

In the trajectory prediction task, the bidirectional recurrent neural network processes the entire trajectory in the forward and reverse orders, and each output node comprises complete context information at the current time. The bidirectional GRU (BiGRU) structure is shown in Figure 5. The first GRU network processes the forward vessel trajectory, whereas the second GRU network processes the reverse vessel trajectory. The outputs of the forward and reverse networks are spliced into the final output $h_t = (h_{t1}, h_{t2})$ after each time step. Compared with an ordinary GRU, the BiGRU has additional feature extraction.

In the forward calculation process, the trajectory $[p_1', p_2', \ldots, p_n']$ is input into the forward GRU unit, and the hidden layer output of the forward unit is saved. In the backward calculation process, input the trajectory $[p_n', p_{n-1}', \ldots, p_1']$ into the backward GRU unit, and save the output of the backward hidden layer. At each moment, concatenate the corresponding output results; the output of the BiGRU layer is $[h_1, h_2, \ldots, h_n]$.

The dense layer maps the output to the target dimension, and the result *pre* of the Stacked-BiGRUs model is the next location of the vessel. The output *pre* should be mapped to the original dimension of the sample pre*ı*.

$$\text{pre}' = \sigma \text{pre} + u. \tag{4}$$

Multi-step prediction of vessel trajectory can be realized using the sliding window method. Figure 6 is a schematic diagram of the sliding window method, in which the window size is 5 and the number of response steps is 1. For the trajectory on the left, the sliding window inputs the historical trajectory from t-4 to t, and outputs the predicted point pre' at t+1. For the trajectory on the right, the historical trajectory point from t-3 to t and pre' was taken as input, and the predicted trajectory point at t+2 was the predicted point. In this way, the predicted trajectory of any time step can be output.

To evaluate the model, the mean square error (MSE) and the mean absolute error (MAE) were used to evaluate the effect of trajectory prediction. MSE is the squared expectation of the difference between the predicted value and the true value; MAE is the average of the absolute error.

```
Input: Samples to be clustered $L$, sailing radius $\epsilon$, the smallest number of vessels in a cluster Min$Pts$, number of behaviors $k$
Output: clustering results $Act$
Mark all points in $L = \{p_1, p_2, \ldots, p_n\}$ as unvisited//Start DBSCAN clustering
Calculate the matrix $M$, with each cell representing the Mercator distance between each two points
Do
  Randomly select an unvisited point $p_i$
  Mark $p_i$ as visited
  Initialize $C = \varnothing$
    If there are at least Min$Pts$ points in $\epsilon$ field of $p_i$, then//The mercator distance between two points can be found in the M
    Initialize $C_{\text{temp}} = \varnothing$, add $p$ to $C_{\text{temp}}$
    Let $N$ be the points set in the $\epsilon$ field of $p_i$
    For each $p_i'$ in $N$
      If $p_i'$ is unvisited, then
      Mark $p_i'$ as visited
      If there are at least Min$Pts$ points in the $\epsilon$ field of $p_i'$, then
      Add points to $N$
    End If
    If $p_i'$ is not a member of any cluster, then
        Add $p_i'$ to $C_{\text{temp}}$
      End If
    End If
  End For
  Add $C_{\text{temp}}$ to $C$
  Else mark $p_i$ as noise point
  End If
Until all the points are marked, $C = \{C_1, C_2, \ldots, C_m\}$ //DBSCAN clustering is complete
Compute the average speed set $V = \{\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(m)}\}$ of each $C_i$ in $C = \{C_1, C_2, \ldots, C_m\}$
Select $k$ points as the initial center point: $\{\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(k)}\}$ //Start KMeans clustering
Do
  Initialize $Act_i = \varnothing\,(1 \le i \le k)$
  For $x^{(j)}$ in $V = \{\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(m)}\}$ do
    Compute the speed difference between $x^{(j)}$ and $\mu^{(i)}$
    The cluster label of $x^{(j)}$ was determined according to the nearest cluster center
    Add $x^{(j)}$ to the nearest cluster: $\text{Act}_i = \text{Act}_i \cup \{x^{(j)}\}$
    For $i = 1, 2, \ldots, k$ do
      Compute the new cluster center: $(\mu^{(i)})^{'} = 1/|Act_i|\sum_{x\in Act_i} x$
    If $(\mu^{(i)})' = \mu^{(i)}$, then
        Update $(\mu^{(i)})'$ as the cluster center
      End If
    End For
  End For
Until the update of all clusters is complete//KMeans clustering is complete
Return $\text{Act} = \{\text{Act}_1, \text{Act}_2, \ldots, \text{Act}_k\}$ //Behavior classification is complete
```

Algorithm 3: D-KMeans.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2,$$
$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|y_i - \hat{y}_i\right|. \tag{5}$$

4. Results and Discussion

4.1. Experimental Environment and Dataset. The platform hardware configuration was a 2.9 GHz six-core Intel i5-9400CPU with 16 GB memory and Intel UHD Graphics 630. The following frameworks were used in the development process: *Python* 3.7-based deep learning framework TensorFlow 2.0 and Keras, Scikit-learn for data processing, and GeoPandas and MovingPandas for trajectory analysis and visualization.

The dataset was selected from the data of vessels in the East China Sea, containing more than 100 GB of AIS point information collected from different types of vessels. The data were stored in the Analytical Massively Parallel Processing (MPP) database in real time, and the spatial connection and spatial index (PostGIS) were established simultaneously to realize the rapid extraction of trajectory data at a specific time and area. Between January 28 and February 1, 2021, 624,307 AIS data points from 522 vessels were selected as experimental data.

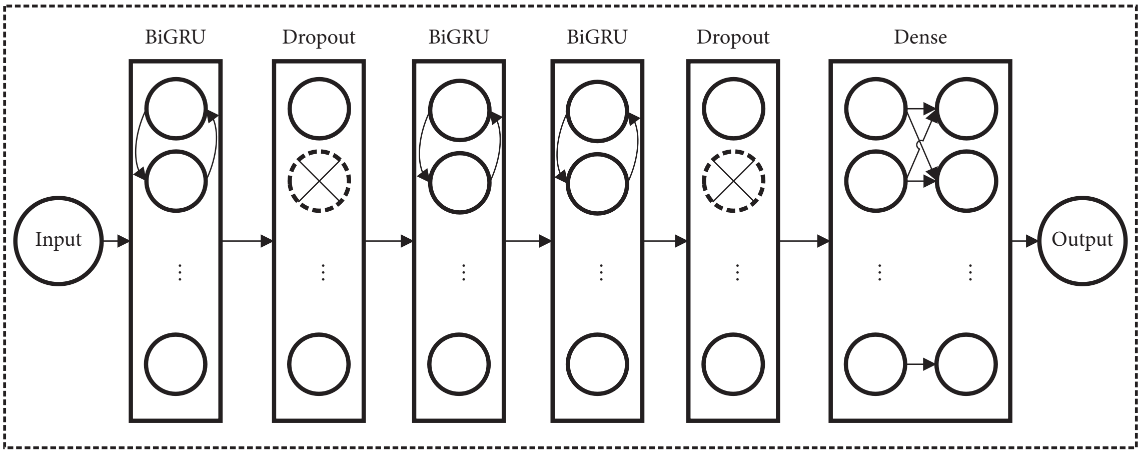


Figure 4: Structure of a Stacked-BiGRUs model.

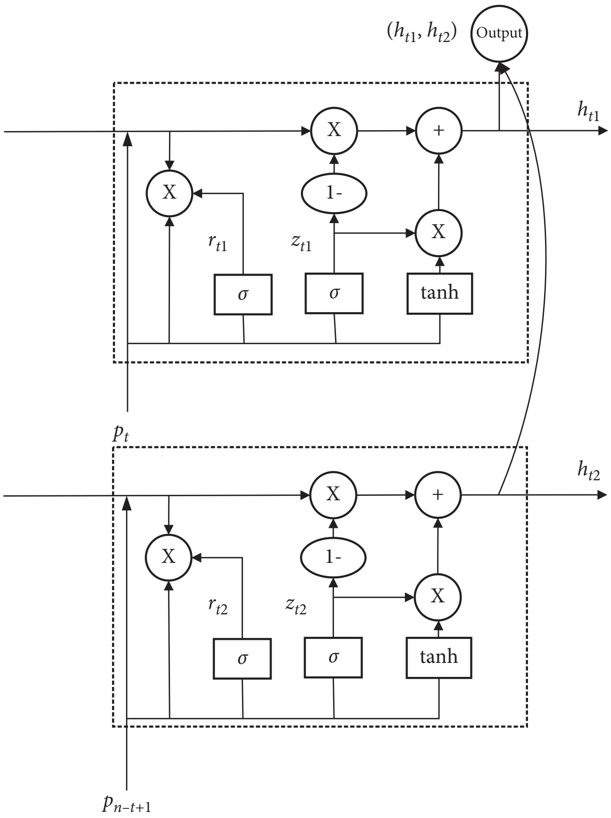


Figure 5: Structure of a bidirectional GRU.

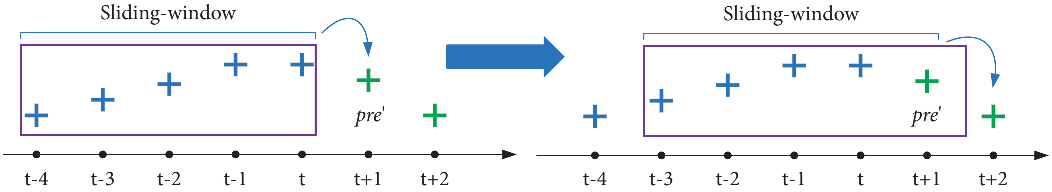


Figure 6: Sliding window method in trajectory prediction.

4.2. Anchor Trajectory Elimination. The anchor trajectory elimination algorithm is based on speed constraints; hence, it was necessary to perform statistical analysis on the speed of the Zhoushan offshore vessel. The primary research object of this study was a small vessel with a length of less than 60 m. The hull was characterized by small linear dimensions, low mass, small acceleration, and stopping inertia. Therefore, it was easily affected by external forces during movement. When the length of this type of vessel is twice the length of the berth, the speed of the vessel can be controlled below 0.3 knots.

From Figure 7, the speed of the vessels in the dataset is approximately two knots, and the remaining speeds are distributed between 0 and 0.5 knots and between 4 and 12 knots. The position of the vessel below 0.3 knots represents the anchoring state of the vessel. In the experiment, the anchor speed threshold V_T in the algorithm was set to 0.3 knots and the time step T_s was set to 5.

The experiment uses the dataset marked with anchor trajectories to test the algorithm performance. The dataset contains a total of 39,662 AIS trajectory points of 58 vessels, of which 16,411 trajectory points are vessel anchor points, and 17 vessels are completely berthed vessels. Table 1 shows the comparison of the number of stopped vessels and the number of anchored AIS points before and after processing the dataset by the algorithm. The results show that the total number of vessels processed by the algorithm has decreased by 17, all completely berthed vessels have been identified, and their anchoring trajectories have been eliminated; all points are reduced by 41%, and the total number of anchor points is reduced by 97.9%, indicating that the anchor trajectory elimination algorithm can effectively eliminate most of the anchor trajectories. The remaining anchor points that have not been cleared are mainly composed of abnormal points.

The visualization comparison of the chart before and after the algorithm processing is shown in Figure 8. The line segments of different colors represent the AIS trajectories of different vessels. The dense ring-shaped trajectories in the figure represent the trajectory data of the floating and anchored vessels. Affected by wind and ocean currents, it reciprocates in a small area. The picture on the right shows the processed chart trajectory. Compared to the left picture, the anchor trajectories are completely eliminated, which proves that the algorithm has a better processing effect.

4.3. Trajectory Restoration. After eliminating anchor trajectories, a section of the vessel trajectory that includes 1527 AIS points was selected to carry out experiments; the anchor trajectory was eliminated and the trajectory was split into a longitude and latitude sequence. The latitude sequence was selected as an example to show the repair result. Gaussian noise was added to the true sequence to obtain a dirty sequence, as shown in Figure 9.

The max repair range $\varepsilon_{\max,lat}$ changed from one to four, as shown in Figure 10. As the repair cost increased, the repair effect increased accordingly. The repaired curve gradually fitted the real curve before Gaussian noise was added.

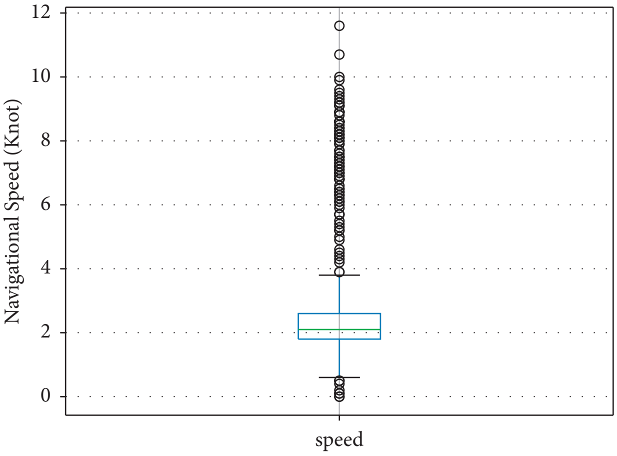


Figure 7: Box diagram of vessel speed.

Figure 11 is a graph of the RMSE of curve repair versus repair cost. Experimental results show that when the repair cost was four, the repair effect was strong, and the RMSE reached 0.0131, which is 58.9% lower than when the repair cost was one.

4.4. Classification of Ship Behavior. After repairing the outliers, the two-stage vessel trajectory flow clustering algorithm D-KMEANS was used to extract the trajectory of vessels crossing the waterway, as shown in Figure 12. We chose a vessel that has been processed in the previous section; the trajectory of the vessel 271217 contained 2459 AIS points. The vessel had experienced multiple departure and return cycles, and the behaviors of the vessel in different voyages had obvious temporal and spatial characteristics. We considered the historical trajectory data as the sample L and used the D-KMEANS algorithm for clustering.

(1) We calculated the distance between each point of the trajectory and stored it as a Mercator distance matrix. The density of DBSCAN reached a radius of 3.6, the minimum sample value was 2, and the 2459 points in L were clustered. The spatial clustering results were clustered into 266 categories, as shown in Figure 12. Of these, 23 categories contained only one piece of data, which were outliers.

(2) We excluded abnormal categories, leaving 243 categories to form a new sample L'. We then calculated the average speed of each type of AIS point and recorded the average speed set of all as V. The first step of clustering was complete.

Second, we performed a second-step clustering of the average speed set using KMEANS, as follows:

(1) We clustered the average velocity set k into three categories. The average speed cluster between 0.5 and 1.5 knots was classified as low speed, the average speed cluster between 1.5 and 3 knots was classified as medium speed, and the average speed cluster between 5.5 and 11 knots was classified as high speed.

Table 1: Comparison of data before and after anchor trajectory eliminating.

State	Number of vessels	Number of points	Number of berthed vessels	Number of anchor points
Before	58	39662	17	16411
After	39	23578	0	327

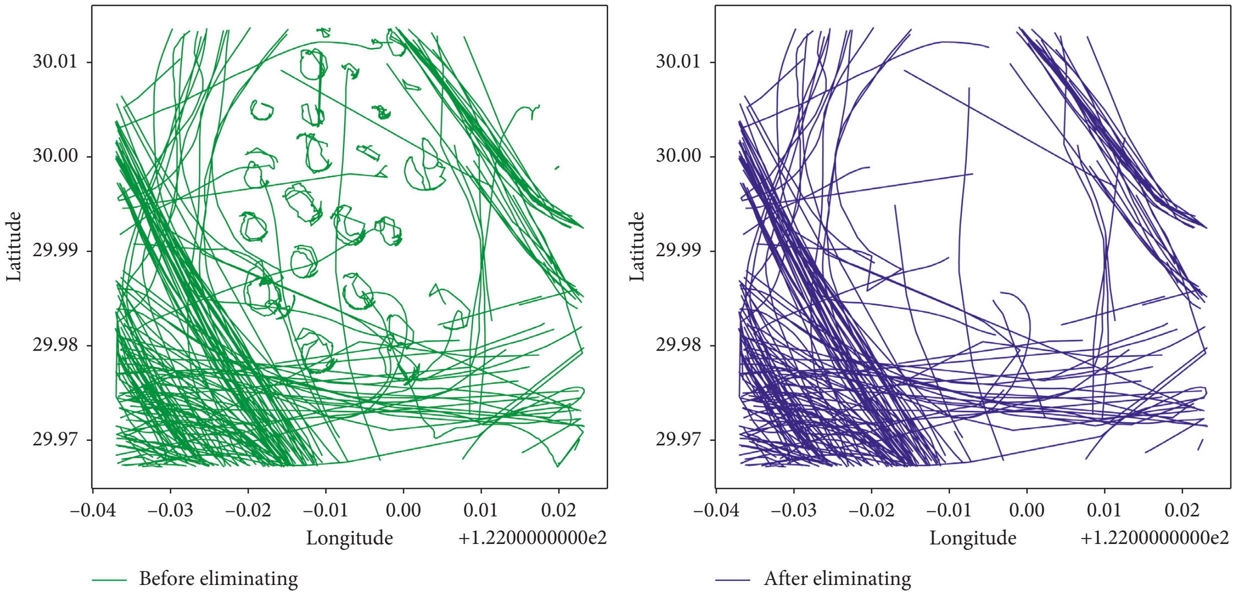


Figure 8: Anchor trajectories before and after eliminating.

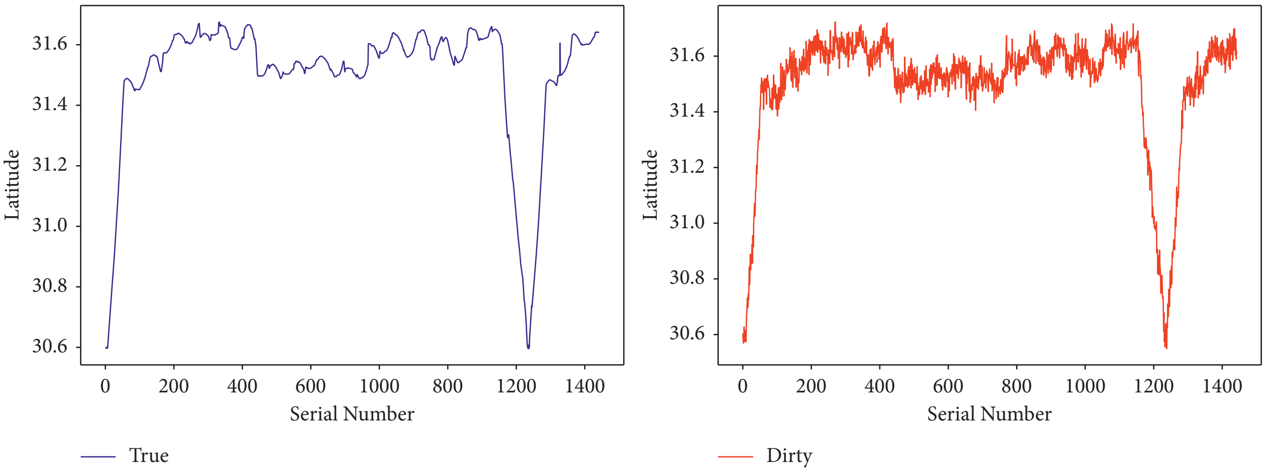


Figure 9: Latitude series with Gaussian noise.

Each average speed in V was labeled as low speed, medium speed, or high speed.

(2) We used the label obtained in step (1) to divide the 266 classes in L' into three classes. Finally, the second stage of clustering was complete.

As shown in Figure 13, clusters formed by blue dots represent low-speed trajectories, red squares represent medium-speed trajectories, and green triangles represent high-speed trajectories. This distribution shows the following obvious behavioral characteristics: when the vessel was in the initial state, its speed was slow; when the vessel entered the waterway to sail to the work area, its speed increased, and when the vessel reached its destination for operation, its speed was medium. Table 2 lists the relationship between the speed and vessel behavior. This study used the green high-speed trajectory of a vessel sailing in a waterway as an example to perform the next prediction.

4.5. Trajectory Prediction. The dataset was divided into the following three parts: training set, verification set, and test set, with a ratio of 6 : 2 : 2. The training set was used to train

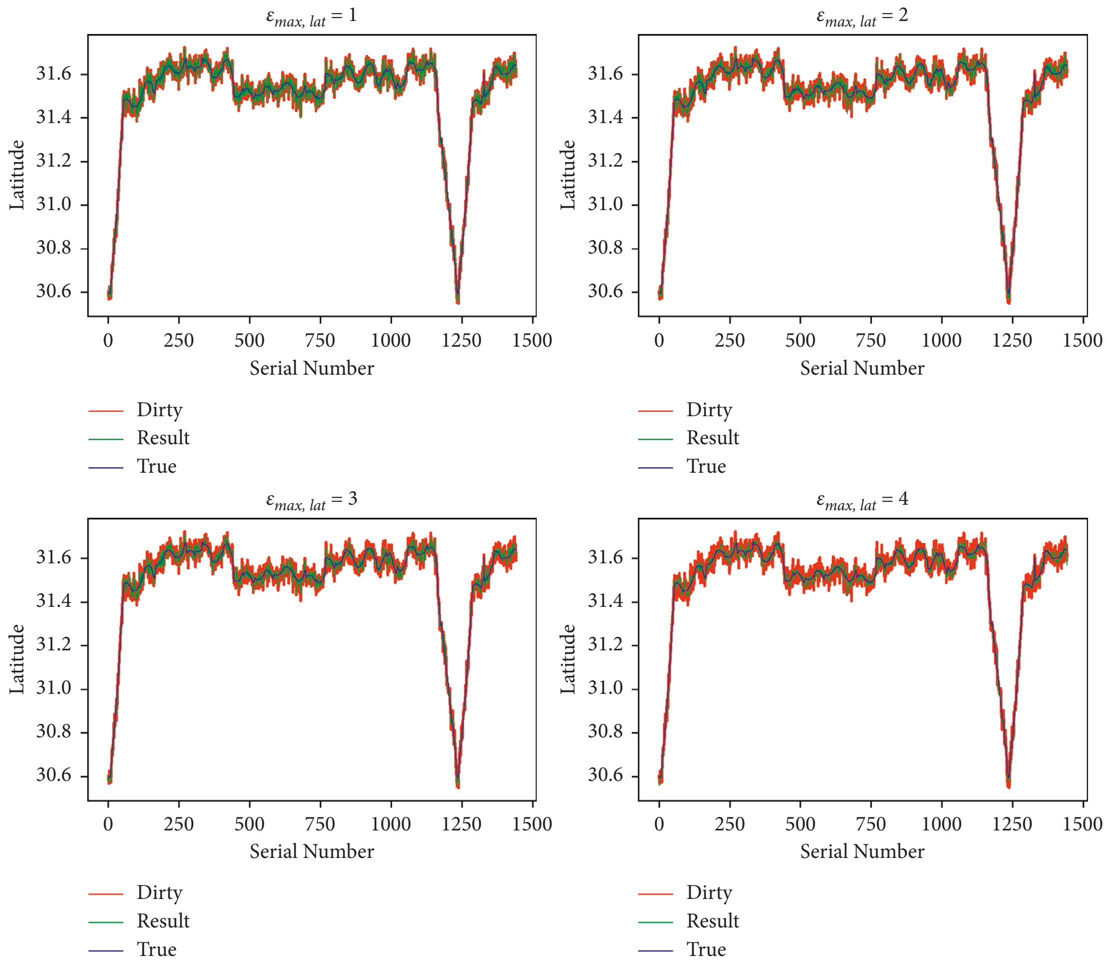


FIGURE 10: Latitude sequence repair effect changes with $\varepsilon_{\max,lat}$.

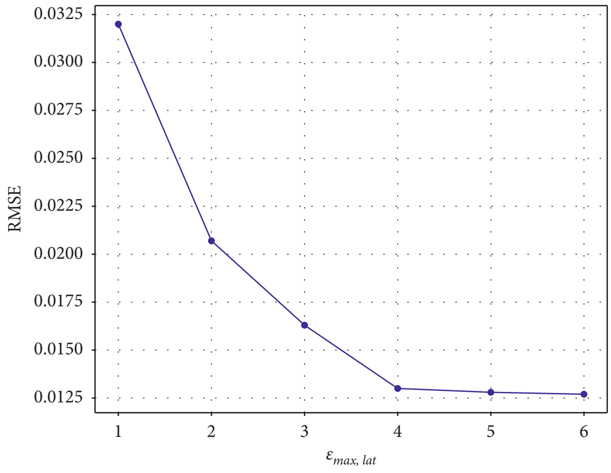


FIGURE 11: Variation of RMSE with the repair cost.

the model. In the training process, the verification set was employed to verify the performance of the model and improve its generalization ability. The test set was used to generate some prediction results. The Adam optimizer was used as the activation function of the hidden and output layers. The selectable range of the batch size was {16, 32, 64, 128, and 256}, and the experimental results of different batch size parameters are shown in Table 3.

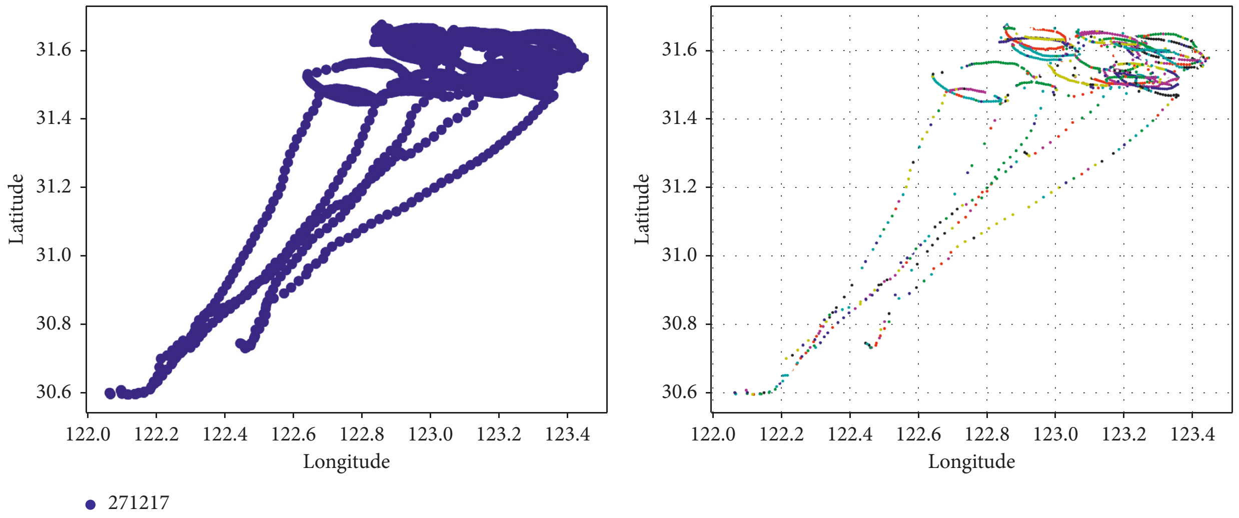


FIGURE 12: First-stage DBSCAN clustering result. First, we performed the first-step clustering of the trajectory using the DBSCAN algorithm, as follows:

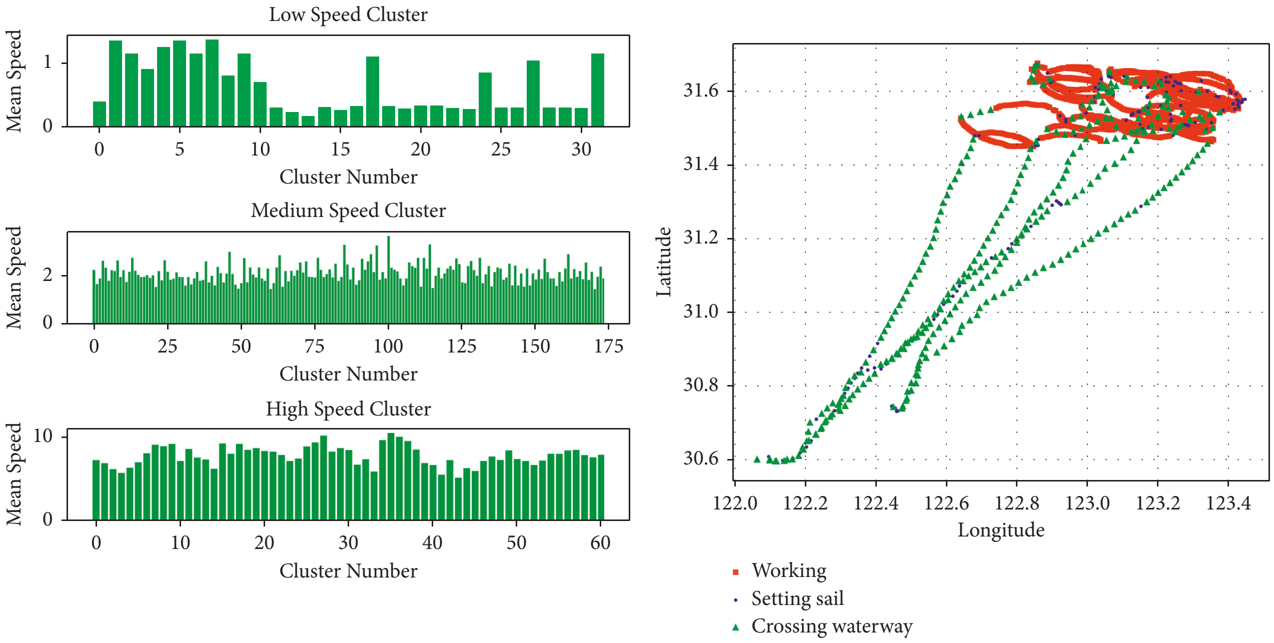


FIGURE 13: Second-stage KMeans clustering result.

TABLE 2: Comparison of data before and after anchor trajectory eliminating.

Speed status	Ground speed (knots)	Ship behavior
Low speed	$0.5 < Sog \leq 1.5$	Setting sail
Medium speed	$1.5 < Sog \leq 5.5$	Working
High speed	$5.5 < Sog \leq 12$	Crossing waterway

The training process is shown in Figure 14. The three models had fast iteration speeds in the first three rounds. When the number of training rounds was approximately 150, the models reached the extremum. The MSE of the LSTM model was 0.0037 and the MAE was 0.036; the MSE of the stacked-BiLSTM model was 0.0021 and the MAE was 0.0194; and the MSE of the stacked-BiGRU was 0.0018 and the MAE was 0.0191. The deep bidirectional structure had additional feature extraction; therefore, the stacked-BiLSTM model and the stacked-BiGRU model presented lower errors.

The stacked-BiGRU and stacked-BiLSTM models had similar losses. When the number of training rounds was approximately 10, the MSE of the stacked-BiGRU model was

Table 3: Comparison of the experimental results of different batch size parameters.

Model	Metrics	Batch size				
		16	32	64	128	256
LSTM	MSE	0.00383	0.00371	0.00472	0.00611	0.0093
	MAE	0.0378	0.0363	0.0465	0.0505	0.0721
Stacked-BiLSTMs	MSE	0.00221	0.00209	0.00313	0.00422	0.00627
	MAE	0.0202	0.0194	0.0298	0.0441	0.0602
Stacked-BiGRUs	MSE	0.00184	0.00180	0.00247	0.00317	0.0544
	MAE	0.0216	0.0191	0.0207	0.0322	0.0511

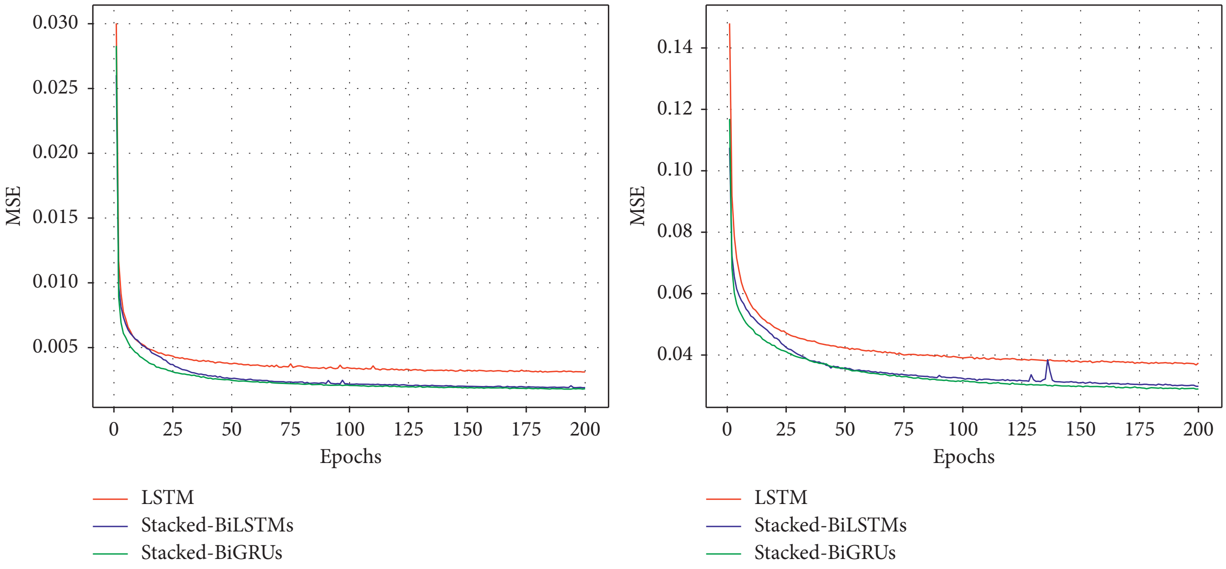


Figure 14: Comparison of three models.

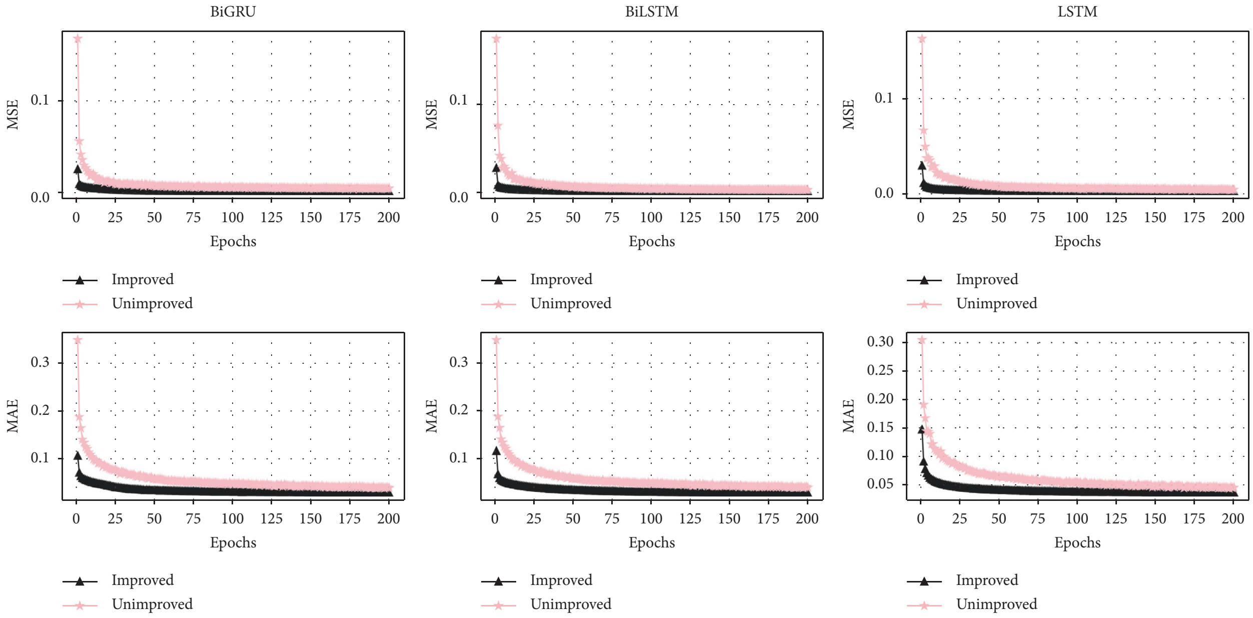


Figure 15: The impact of model improvement on prediction accuracy.

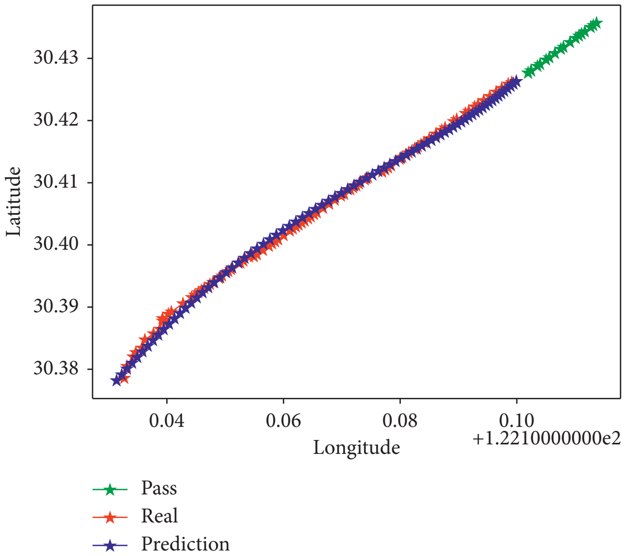


Figure 16: Comparison of the predicted trajectory and the real trajectory.

0.004, while the stacked-BiLSTM model reached this value at the 20th round. The stacked-BiGRU model converged faster mainly because the gate of the GRU unit was more simplified than the LSTM unit.

We also compared the impact of anchor trajectory elimination, outlier repair, and behavior classification improvement on different recurrent neural network models. As shown in Figure 15, the MSE of the improved model was 27% lower than that of the unimproved model on average, and the MAE was 46% lower than that of the unimproved model. The model converges after 55 epochs on average before improvement, and the improved model converges after 26 epochs. The results show that the improved model has quicker convergence rapidity and less error. This is because after improving, the abnormal data were eliminated, and the characteristics of the trajectory data were more concentrated, making it easier to analyze the inherent laws of the trajectory data.

The results of the simulation prediction are shown in Figure 16. From a path plan developed by the test, using a flexible window, the output of the previous model was used as the new trajectory data input, the planning model results at the corresponding time were repeatedly generated, and the predictions were 100 trajectories of flight trajectories. The online green line represents the historical trajectory, the blue line represents the predicted trajectory, and the red line represents the real trajectory. The predicted trajectory basically fitted the real trajectory, achieving a good trajectory prediction effect.

5. Conclusions

Trajectory prediction is a key requisite for navigation; in this research, to further improve the quality of maritime navigation in IoV, we considered the influence of anchor trajectory, trajectory abnormal points, and different vessel behavior characteristics on trajectory prediction, and designed an improved vessel trajectory prediction model based on a recurrent neural network. For the anchor trajectory in the data, an anchor trajectory elimination algorithm was proposed to detect and eliminate abnormal data. A statistical trajectory restoration algorithm was proposed to repair the abnormal points in the trajectory. The vessel behavior classification algorithm D-KMEANS realized the extraction of different vessel behavior trajectories. Finally, a Stacked-BiGRUs model was built, and the sliding window was used to iteratively predict the position of the vessel at any step length.

The experimental results of the data processing part showed that the proposed algorithm achieved the expected results in terms of anchor trajectory elimination, trajectory repair, and vessel behavior classification. The comparative experiment of prediction models proved the performance of the Stacked-BiGRUs model in terms of prediction accuracy and convergence speed. This was mainly because the bidirectional model extracted additional features of the data, and the simplified gate structure of the GRU unit improved the training efficiency. The comparative experiments to verify the accuracy of the model showed the mean square error of the improved model is 0.0018 and the mean absolute error is 0.0191, which are reduced by 27% and 46%, depicting that the improved method can effectively improve prediction accuracy. The processing eliminated anchor trajectories and repaired abnormal data, and behavior classification resulted in a higher concentration of the characteristics of the vessel trajectory data, which made it convenient for the model to mine the inherent laws of trajectory data. Owing to this, the method proposed in this study may be well suited to proactively assist collision avoidance systems in ports and offshore areas.

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 in part by the National Natural Science Foundation of China under grants J2024009 and 62072146 and in part by the Zhejiang Key Research and Development Program under grants 2019C05005 and 2021C03187.

References

[1] W. Zhang, A. M. Abdulghani, M. A. Imran, and Q. H. Abbasi, “Internet of things (IoT) enabled smart home safety barrier system,” in *Proceedings of the 2020 International Conference on Computing, Networks and Internet of Things*, pp. 82–88, Sanya, China, April 2020.

[2] J. Xiong, R. Ma, L. Chen et al., “A personalized privacy protection framework for mobile crowdsensing in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2019.

[3] J. Xiong, R. Bi, Y. Tian, X. Liu, and D. Wu, “Towards lightweight, privacy-preserving cooperative object classification for connected autonomous vehicles,” *IEEE Internet of Things Journal*, vol. 9, no. 4, 2021.

[4] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, “Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles,” *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.

[5] S. Song, A. Zhang, J. Wang, and P. S. Yu, “SCREEN: Stream Data Cleaning under Speed constraints,” in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 827–841, Victoria, Australia, May 2015.

[6] E. Tu, G. Zhang, S. Mao, L. Rachmawati, and G.-B. Huang, “Modeling Historical AIS Data for Vessel Path Prediction: A Comprehensive treatment,” 2020, https://arxiv.org/abs/2001.01592.

[7] B. Li, Z. Cai, M. Kang et al., “A trajectory restoration algorithm for low-sampling-rate floating car data and complex urban road networks,” *International Journal of Geographical Information Science*, vol. 35, pp. 1–24, 2020.

[8] F. Gao, S. Song, S. Song, and J. Wang, “Time series data cleaning under multi-speed constraints,” *International Journal of Software and Informatics*, vol. 11, no. 1, pp. 29–54, 2021.

[9] S. Kanarachos, S.-R. G. Christopoulos, A. Chroneos, and M. E. Fitzpatrick, “Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and Hilbert transform,” *Expert Systems with Applications*, vol. 85, pp. 292–304, 2017.

[10] Z. Cheng, Z. Jiang, X. Chu, and L. Liu, “Inland vessel trajectory restoration by recurrent neural network,” *Journal of Navigation*, vol. 72, no. 6, pp. 1359–1377, 2019.

[11] H. Xue, “Fractional-order gradient descent with momentum for RBF neural network-based AIS trajectory restoration,” *Soft Computing*, vol. 25, no. 2, pp. 869–882, 2021.

[12] J. Mao, Q. Song, C. Jin, Z. Zhang, and A. Zhou, “Online clustering of streaming trajectories,” *Frontiers of Computer Science*, vol. 12, no. 2, pp. 245–263, 2018.

[13] J. Xiong, J. Ren, L. Chen et al., “Enhancing privacy and availability for data clustering in intelligent electrical service of IoT,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2018.

[14] B. Liu, E. N. de Souza, S. Matwin, and M. Sydow, “Knowledge-based clustering of vessel trajectories using density-based approach,” in *Proceedings of the 2014 IEEE International Conference on Big Data (Big Data)*, pp. 603–608, IEEE, Washington, DC, USA, Octomber 2014.

[15] M. Sun and J. Wang, “An approach of ship trajectory clustering based on minimum bounding rectangle and buffer similarity,” *IOP Conference Series: Earth and Environmental Science*, vol. 769, no. 3, Article ID 032017, 2021.

[16] X. Han, C. Armenakis, and M. Jadidi, “Dbscan optimization for improving marine trajectory clustering and anomaly detection,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 455–461, 2020.

[17] Y. Tian, Z. Zhang, J. Xiong, L. Chen, J. Ma, and C. Peng, “Achieving graph clustering privacy preservation based on structure entropy in social IoT,” *IEEE Internet of Things Journal*, vol. 9, p. 1, 2021.

[18] N. Budimirovic and N. Bacanin, “Novel algorithms for graph clustering applied to human activities,” *Mathematics*, vol. 9, no. 10, p. 1089, 2021.

[19] Y. T. Wen, C. H. Lai, P. R. Lei, and W. C. Peng, “Routeminer: mining vessel routes from a massive maritime trajectories,”vol. 1, pp. 353–356, in *Proceedings of the 2014 IEEE 15th international conference on mobile data management*, vol. 1, pp. 353–356, IEEE, Brisbane, Australia, July 2014.

[20] D. Peel and N. M. Good, “A hidden Markov model approach for determining vessel activity from vessel monitoring system data,” *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 68, no. 7, pp. 1252–1264, 2011.

[21] M. J. Riveiro, *Visual Analytics for Maritime Anomaly detection*, Örebro universitet, Örebro, 2011.

[22] K. Jaskólski, “Automatic identification system (AIS) dynamic data estimation based on discrete kalman filter (KF) algorithm,” *Scientific Journal of Polish Naval Academy*, vol. 211, no. 4, pp. 71–87, 2017.

[23] S. Qiao, N. Han, X. Zhu, L. Chen, J. Ma, and C. Pen, “A dynamic trajectory prediction algorithm based on Kalman filter,” *Acta Electonica Sinica*, vol. 46, no. 2, p. 418, 2018.

[24] F. Mazzarella, V. F. Arguedas, and M. Vespe, “Knowledge-based vessel position prediction using historical AIS data,” in *Proceedings of the 2015 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1–6, IEEE, Bonn, Germany, Octomber 2015.

[25] T. Xiaopeng, C. Xu, S. Lingzhi, M. Zhe, and W. Qing, “Vessel trajectory prediction in curving channel of Inland river,” in *Proceedings of the 2015 International Conference on Transportation Information and Safety (ICTIS)*, pp. 706–714, IEEE, Wuhan, China, June 2015.

[26] S. Qiao, C. Tang, H. Jin et al., “PutMode: prediction of uncertain trajectories in moving objects databases,” *Applied Intelligence*, vol. 33, no. 3, pp. 370–386, 2010.

[27] H. Rong, A. P. Teixeira, and C. Guedes Soares, “Ship trajectory uncertainty prediction based on a Gaussian Process model,” *Ocean Engineering*, vol. 182, pp. 499–511, 2019.

[28] S. Anderson, T. D. Barfoot, C. H. Tong, and S. Särkkä, “Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression,” *Autonomous Robots*, vol. 39, no. 3, pp. 221–238, 2015.

[29] S. J. Qiao, K. Jin, N. Han, C. J. Tang, and G. L. Gesangduoji, “Trajectory prediction algorithm based on Gaussian mixture model,” *Journal of Software*, vol. 26, no. 5, pp. 1048–1063, 2015.

[30] B. R. Dalsnes, S. Hexeberg, A. L. Flåten, B. -O. H. Eriksen, and E. F. Brekke, “The neighbor course distribution method with Gaussian mixture models for ais-based vessel trajectory prediction,” in *Proceedings of the 2018 21st International Conference on Information Fusion (FUSION)*, pp. 580–587, IEEE, Cambridge, UK, July 2018.

[31] S. Mao, E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, and G.-B. Huang, “An automatic identification system (ais) database for maritime trajectory prediction and data mining,” in *Proceedings of the ELM-2016*, pp. 241–257, Springer, Cham, 2018.

[32] B. Murray and L. P. Perera, “A dual linear autoencoder approach for vessel trajectory prediction using historical AIS data,” *Ocean Engineering*, vol. 209, Article ID 107478, 2020.

[33] D.-W. Gao, Y.-S. Zhu, J.-F. Zhang, Y.-K. He, K. Yan, and B.-R. Yan, “A novel MP-LSTM method for ship trajectory prediction based on AIS data,” *Ocean Engineering*, vol. 228, Article ID 108956, 2021.

[34] D. D. Nguyen, C. Le Van, and M. I. Ali, “Vessel trajectory prediction using sequence-to-sequence models over spatial grid,” in *Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems*, pp. 258–261, Hamilton, New Zealand, June 2018.

[35] J. Chen, G.-Q. Zeng, W. Zhou, W. Du, and K.-D. Lu, “Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization,” *Energy Conversion and Management*, vol. 165, pp. 681–695, 2018.

[36] Y. Suo, W. Chen, C. Claramunt, and S. Yang, “A ship trajectory prediction framework based on a recurrent neural network,” *Sensors*, vol. 20, no. 18, p. 5133, 2020.

[37] H. Xiao, C. Wang, Z. Li et al., “UB-LSTM: a trajectory prediction method combined with vehicle behavior recognition,” *Journal of Advanced Transportation*, vol. 2020, Article ID 8859689, 12 pages, 2020.

[38] L. Zhang, J. Zhang, J. Niu, Q. M. J. Wu, and G. Li, “Track prediction for HF radar vessels submerged in strong clutter based on MSCNN fusion with GRU-AM and AR model,” *Remote Sensing*, vol. 13, no. 11, p. 2164, 2021.

[39] K. U. Jaseena and B. C. Kovoor, “Decomposition-based hybrid wind speed forecasting model using deep bidirectional LSTM networks,” *Energy Conversion and Management*, vol. 234, Article ID 113944, 2021.

[40] H. Xue, D. Q. Huynh, and M. Reynolds, “Ss-lstm: A Hierarchical Lstm Model for Pedestrian Trajectory prediction,” in *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1186–1194, IEEE, Lake Tahoe, NV, USA, March 2018.

[41] M. Robards, G. Silber, J. Adams et al., “Conservation science and policy applications of the marine vessel Automatic Identification System (AIS)-a review,” *Bulletin of Marine Science*, vol. 92, no. 1, pp. 75–103, 2016.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 4992957, 13 pages
https://doi.org/10.1155/2022/4992957

Research Article

Adaptive Bandwidth Prediction and Smoothing Glitches in Low-Latency Live Streaming

Dapeng Wu, [1,2,3] Linfeng Cui, [1,2,3] Tong Tang, [1,2,3] and Ruyan Wang [1,2,3]

[1] *School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[2] *Advanced Network and Intelligent Interconnection Technology Key Laboratory of Chong Qing Education Commission of China, Chongqing 400065, China*
[3] *Chongqing Key Laboratory of Ubiquitous Sensing and Networking, Chongqing 400065, China*

Correspondence should be addressed to Tong Tang; tangtong@cqupt.edu.cn

Received 17 January 2022; Accepted 13 April 2022; Published 9 May 2022

Academic Editor: Qi Li

HTTP adaptive streaming (HAS) technologies such as dynamic adaptive streaming over HTTP (DASH) and common media application format (CMAF) are now used extensively to deliver live streaming services to large numbers of viewers. However, in dynamic networks, inaccurate bandwidth prediction may result in the wrong request of bitrate, and short-term network fluctuations may produce glitches, causing unnecessary bitrate switching, thereby degrading clients' Quality of Experience (QoE). To tackle this, we propose adaptive bandwidth prediction and smoothing glitches in low-latency live streaming (called APSG) in this article. Concretely, firstly, the size of random bandwidth fluctuations is exploited as the weight of exponentially weighted moving average (EWMA) for adaptive bandwidth prediction; in addition to bandwidth prediction and buffer occupancy, glitches phenomena under a stable network environment are taken into account to enhance the viewing experience of clients. Finally, experimental results show that compared to traditional ABR algorithms under a stable network environment, APSG could reduce the number of bitrate switches and latency by up to 72.6% and 27.3%, respectively; under a dynamic network environment, APSG could reduce the number of bitrate switches and latency by up to 53.8% and 23.6%, respectively.

1. Introduction

In recent years, the development of mobile networks and streaming technologies has enabled clients to watch live streaming on their mobile devices at any time, with video accounting for 67% of global traffic in 2016 and expected to reach 80% by 2022, according to Cisco's Annual Visual Networking Index Report [1]. Today, live streaming platforms, such as Huya and Douyu, attract millions of active clients, and video content providers have become more interested in live streaming as client's engagement will directly increase commercial revenue. This trend means that high-quality videos with fewer switches, lower latency, less rebuffering, and higher bitrate need to be provided to clients.

In DASH, the video is divided into multiple segments, each with a duration of approximately 2 to 10 seconds, and encoded at different bitrates and resolutions [2]. On the client's side, the ABR algorithm takes into account network environments or buffer occupancy to pick the right bitrate for the clients and fetch it from the server. In a traditional video on demand (VoD) scenario, DASH has a large end-to-end latency due to the fact that the entire segment is completely downloaded before it is added to the playback buffer and queued for playback. If the buffer content is empty, then the rebuffering will occur. In live streaming, the latency is generated by the process of capturing video from the anchor to the server and decoding it by the client. The latency is proportional to the size of the segment, and if the duration of the segment is reduced to achieve the purpose of reducing the latency, then the number of requests and the round-trip time (RTT) will increase significantly. In order to achieve target latency without reducing the duration of the segments, CMAF is a method [3]. CMAF can divide the segments into smaller chunks and then transmit them by

HTTP. When the coding of the chunk is completed, it will be sent to the client, and the remaining chunks of the segment will be sent without additional requests; it is unnecessary to send them to the client only after the coding of the entire segment is completed. CMAF significantly reduces latency.

CMAF could reduce latency but bring new challenges; bandwidth measurement becomes nonnegligible. The biggest difference between live streaming and VoD is that live content is generated in real time. Bandwidth measurement usually uses the size of the segment divided by the download time of the segment in the VoD. However, HTTP does not provide a download start time for each chunk within the segment. If the requested chunk is not encoded, it is necessary to wait until the chunks coding is completed to send it to the client. During this period, there must be idle times between the two chunks. VoD's bandwidth measurement method will underestimate the download rate [4–6]. The ABR algorithm will choose a low bitrate, directly reducing the client's QoE. To solve this problem, Bentaleb et al. proposed the first solution to calculate the bandwidth through the sliding window moving average (SWMA) bandwidth measurement method [7]. When the chunk download rate is close to the average download rate of the segment, this chunk must be disregarded. Although the problem of bandwidth underestimation is solved, the bandwidth will be overestimated, and it is more likely to rebuffer when watching live streaming. In order to solve the problem of bandwidth overestimation, Ozcelik and Ersoy considered the whole segment and subtracted the download end time of the consecutive chunks [8]. If the value is less than the average download time of the segment, then it is considered that there are no idle times between the two chunks. These chunks are used to approximate the download time of the remaining chunks within the segment and reduce the impact of idle times.

Once the exact bandwidth has been measured, the two most important parts are bandwidth prediction and bitrate selection. Traditional bandwidth prediction methods based on time series models are weighted to historical data, which can be estimated online in real-time. Fixed parameters or weights are difficult to apply to all network situations, a smaller number of samples may produce unstable prediction values when the bandwidth changes drastically, and the correlation between premature historical data and current bandwidth is weak. When the bandwidth is at a certain point, there is a glitches phenomenon (i.e., the stable network suddenly changes and then gets back to the original network state), and the bitrate changes accordingly. This unnecessary bitrate switching will directly affect the client's viewing experience.

To address the above issues, firstly, this article proposes an adaptive bandwidth prediction method, which calculates the network stability factor based on historical data, designs the weight values based on the network stability factor, and obtains the adaptive bandwidth prediction values by EWMA. Then APSG algorithm is proposed for smoothing glitches. The bandwidth is differentiated into a stable and dynamic network environment by calculating the network stability factor. The glitches phenomenon is smoothed under a stable network environment. This article takes into account both bandwidth prediction and buffer occupancy and adopts corresponding strategies to select the appropriate bitrate. APSG algorithm is implemented under the DASH.js reference player [9] and extensive experiments have been done under different network environments. The experiments enable APSG to compare with two traditional algorithms in terms of live latency, average bitrate, and the number of bitrate switches.

The rest of this article is organized as follows. Related work is provided in Section 2, followed by the details for the APSG scheme in Section 3. Section 4 presents the experimental evaluation, and Section 5 concludes the article.

2. Related Work

In the past decade, many ABR algorithms have been proposed, which can be divided into four main categories: (1) available bandwidth-based adaptive bitrate algorithms; (2) playback buffer-based adaptive bitrate algorithms; (3) mixed adaptive bitrate algorithms; and (4) data-driven adaptive bitrate algorithms.

(1) Available bandwidth-based adaptive bitrate algorithms: in this type of scheme, the most important thing is to accurately predict bandwidth. Jiang et al. proposed an algorithm called FESTIVE, which mitigates bandwidth jitter caused by stop-and-wait mechanisms by optimizing video chunks scheduling and uses harmonic mean to predict bandwidth [10]. The PANDA proposed by Li et al. uses an EWMA with a weight of 0.2 for bandwidth prediction [11]. Bentaleb et al. implemented a CMAF-based bandwidth measurement algorithm for live streaming and recursive least-squares- (RLS-) based bandwidth prediction [7]. However, the bandwidth overestimation problem occurs when the idle time increases. Ozcelik and Ersoy further addressed the problem of bandwidth overestimation due to idle times based on [7] and used an EWMA method with a weight of 0.9 to calculate the available bandwidth for the next segment [8]. van der Hooft et al. proposed an HTTP/2-based algorithm that discards unimportant frames within a segment when the selected bitrate does not match the available bandwidth [12]. Existing work has shown that selecting the next bitrate based on inaccurate bandwidth prediction values can cause low-quality video or playback rebuffering.

(2) Playback buffer-based adaptive bitrate algorithms: in this type of scheme, clients use the playout buffer occupancy as a criterion to select the next segment bitrate during video playback. Huang et al. proposed a buffer-based bitrate selection algorithm called BBA, which selects the bitrate based on a linear function aimed at maximizing the average video quality and avoiding unnecessary rebuffering events [13]. Spiteri et al. designed a buffer-based online control algorithm that uses Lyapunov optimization techniques to minimize rebuffering and maximize

video quality [14]. Essentially, these two algorithms are mapping the current buffer occupancy. Huang et al. developed a QoE model, including rebuffering, the number of bitrate switches, and video quality, and formulated the problem as a nonlinear stochastic optimal control problem [15]. A dynamic buffer-based controller is designed for DASH using control theory to determine the bitrate of each segment. Qin et al. proposed a framework for PIA by further analyzing ABR video streaming based on proportional-integral-derivative (PID) control and combining several ABR algorithms to address various business requirements [16]. Both of these approaches use a PID controller to control the buffer occupancy. The adaptive bitrate algorithm based on the playback buffer has many limitations, the most serious being that in low-latency live streaming scenarios, the size of the buffer that can be used is drastically reduced, especially under long-term bandwidth fluctuations; there are problems of overall low QoE, unstable selection of bitrates, and too many bitrate switches.

(3) Mixed bitrate algorithms: in this type of scheme, clients select bitrate based on the combination of metrics, including available bandwidth and buffer occupancy. Pioneering this critical work was Yin et al., who modelled bitrate adaptation as a stochastic optimal control problem, proposing a model predictive control (MPC) approach to model cache dynamics and then select the bitrate by optimizing the overall QoE function based on bandwidth prediction and current buffer occupancy as inputs [17], but MPC is sensitive to bandwidth prediction errors and network jitter. A fuzzy logic-based bitrate adaptive algorithm and prediction mechanism was proposed that takes into account buffer occupancy and the prediction of available network bandwidth in order to be able to respond proactively to requests [18]. Reference [19] considered the joint decision of two factors and minimized video bitrate switching. Yarnagula et al. designed a segment-aware rate adaptation (SARA) algorithm by considering segment size to predict the time to download the next segment [20].

(4) Data-driven adaptive bitrate algorithms: CS2P proposed by Sun uses a hidden Markov model to design a prediction model by analyzing the evolution trajectory of download rate [21]. In [22, 23], an ABR algorithm is based on deep reinforcement learning. With the powerful approximation ability of the neural network, the best mapping between various states and bitrate selection is learned. However, when encountering untrained network environments, the overall QoE will be very poor. Another disadvantage is that it is difficult to reproduce these ABR algorithms based on deep reinforcement learning [24, 25].

The approach used in this paper is based on a joint decision between bandwidth prediction and current buffer occupancy. On the one hand, the adaptive bandwidth prediction method is used to improve bandwidth prediction accuracy; on the other hand, the APSG method is used to solve the glitches phenomenon, thus significantly improving the quality of service experience for clients.

3. Proposed APSG

In this article, APSG is designed to improve bandwidth prediction accuracy and eliminate glitches caused by bandwidth fluctuations under different network environments. Figure 1 shows the components of the APSG in DASH.js [9], which contains five parts: (1) bandwidth measurement module; (2) bandwidth prediction module; (3) ABR control module; (4) logger module; and (5) playback speed control module. This section will introduce each module in the DASH.js player and elaborate on the details. The list of notations used in APSG is given in Table 1.

3.1. APSG Process. This article selects the appropriate bitrate for clients to match the current network environments. Firstly, the download rate is measured based on the history of the segment after removing the idle times; then, the proposed prediction method is used to get the predicted value of the download rate of the next segment. Then the current buffer occupancy and bandwidth prediction are combined to jointly determine the bitrate of the next segment to be requested and subsequently place the downloaded video in the playback buffer and determine the buffer status and whether the playback speed control module needs to be invoked.

3.2. APSG Design. The next section of this article describes the core functional design and implementation details of the APSG.

3.2.1. Bandwidth Measurement Module. The module uses a heuristic bandwidth measurement method that is able to reduce the impact of idle times on the calculation of download rates. As mentioned earlier, the chunks are encoded and transmitted at the same time, chunks that are not encoded to completion become unavailable, and unavoidable idle time is generated between two consecutive chunks. Therefore, this article uses the bandwidth measurement method of [8]. The method is shown as follows.

Firstly, this article expresses the size and download time of each chunk in a segment as a sequence: $L_i = \{x_i^1, x_i^2, x_i^3, \ldots, x_i^n\}$; there are n chunks in each segment, t_i^j denotes the download end time of the *ith* segment's *jth* chunk, and s_i^j denotes the size of the *ith* segment's *jth* chunk. We can know the download end time but do not know the download start time of each chunk, so we use the download end time of consecutive chunks subtracted from each other as the download time of each chunk. The average arrival time of successive chunks in each segment is as follows:

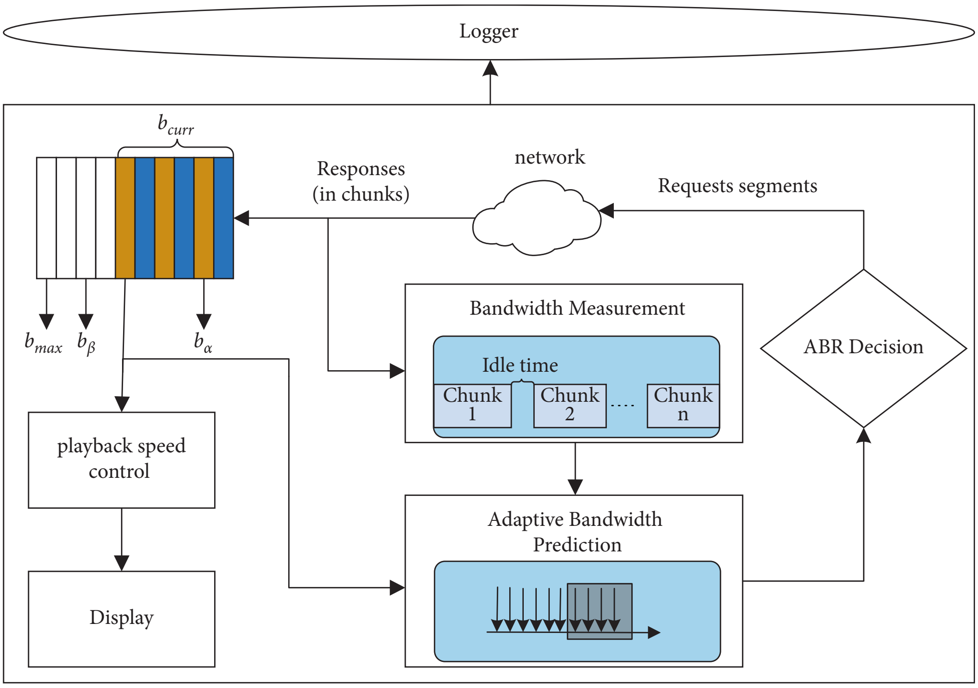


Figure 1: APSG overview in DASH.js reference play.

Table 1: List of notations.

Notation	Meaning
B_i	Download rate of the *ith* segment
$\bar{t}_i$	Average download time of the *ith* segment
t_i^j	Download time of the *jth* chunk of the *ith* segment
x_i^j	The *jth* chunk of the *ith* segment in the extra sequence
s_i^j	Size of the *jth* chunk of the *ith* segment
T_i	Download time of the *ith* segment
T_i'	Download time of the *ith* segment without idle times
ρ_i	Approximate coefficient of the *ith* segment
α	Network stability factor
θ	Network stability factor threshold
$\widehat{B}_i$	The predicted bandwidth of the *ith* segment
r_i	Bitrate of the *ith* segment
k	*k* segments in the video sequence

$$\bar{t}_i = \frac{t_i^n - t_i^1}{n-1}. \tag{1}$$

Secondly, these chunks are considered to be unaffected by the encoding side and has already been encoded at the time of request without idle times if the chunks are downloaded faster than the average arrival time. Place these chunks in an additional sequence L_i':

$$L_i' = \left(x_i^j\right),\ \text{s.t.} t_i^j - t_i^{j-1} \le \bar{t}_i \forall x_i^j \in L. \tag{2}$$

Then use the chunks in L_i' to calculate an approximate download rate ρ_i:

$$\begin{aligned}\rho_i &= \frac{\sum_{x_i^j \in L'} s_i^j}{T_i'},\\ T_i' &= \sum_{x_i^j \in L'} t_i^j - t_i^{j-1}.\end{aligned} \tag{3}$$

Once the approximate download rate is obtained, the total size of the remaining chunks and ρ_i are used to estimate the effective download time, reducing the effect of idle times in this step:

$$T_i = \frac{\sum_{x_i^j \in (L/L')} s_i^j}{\rho_i}. \tag{4}$$

Therefore, the average download rate of the *ith* segment is calculated according to $T_i' + T_i$ of (3) and (4):

$$B_i = \frac{\sum_{x_i^j \in L} s_i^j}{T_i' + T_i}. \tag{5}$$

3.2.2. Bandwidth Prediction Module. This module dynamically and adaptively predicts bandwidth in the APSG. It consists of two phases: Phase 1, which calculates the size of network fluctuation; Phase 2, which adaptively predicts the bandwidth of the next segment based on phase 1.

At phase 1, firstly, it is necessary to distinguish whether the network environments are transient jitter or long-term changes, and this article introduces a network stability factor α, the ratio of the standard deviation to the mean of a set of data to indicate the size of the fluctuation. α is calculated by the actual download rate of the last m segments of the sliding window and constructing a set $\{B_{i-m+1}, B_{i-m+2}, \cdots, B_i\}$. The network stability factor is calculated as shown in the following equation:

$$\alpha = \frac{\sqrt{\sum_{j=i-m+1}^{i}\left(B_j - 1/m\sum_{j=i-m+1}^{i} B_j\right)^2}}{1/m\sum_{j=i-m+1}^{i} B_j}. \tag{6}$$

α characterizes the dispersion of the download rate of the nearest m segments. When the value of α is small, it means that there is little fluctuation in bandwidth during the period, but there may be glitches, and those transient changes can seriously affect the client's viewing experience, so the network stability factor is designed to find and eliminate glitches. The network stability factor threshold θ is defined to measure the fluctuation of the bandwidth. The current network environment is considered to be in a stable network environment if α belongs to $(0, \theta)$; otherwise, it is considered to be in a dynamic network environment.

Bandwidth prediction for chunked video streams is not easy, and to solve this problem, traditional bandwidth prediction methods are as follows:

(1) Segment-based last bandwidth: the last successfully downloaded segment is used to predict the next segment

(2) Sliding Window Moving Average (SWMA) [7]: using the last three successfully downloaded segments, find their average

(3) Exponentially Weighted Moving Average (EWMA) [8]: exponentially weighted average bandwidth of the last four segments

(4) Harmonic mean: the total size of the last five segments is divided by the total download rate

The four prediction methods mentioned above all share a common feature, where SWMA and Harmonic have a fixed window size and EWMA has fixed weights, so there cannot predict different network environments. For example, when bandwidth fluctuations are small, there is high prediction accuracy, but when the fluctuations become larger, too early measurements are less relevant to the current network environment. The player will take a long time to download the selected chunks of a high prediction value and therefore may run out of content in the buffer during the download process, causing rebuffering. In VoD scenarios, the ABR algorithm usually has enough cached content to absorb errors, but the playback buffer is small in live streaming scenarios; the wrong request of bitrate causes rebuffering, which can seriously affect the client's viewing experience. Fixed parameters are difficult to apply to all network environments.

So, in phase 2, an adaptive bandwidth prediction method is designed in this article. An initial weight α^j is set for each segment based on the network stability factor, $j \in \{0, 1, \cdots, m-1\}$; m is the window size. Normalizing these weights to their geometric sum, the final weights for each segment are expressed as follows:

$$\alpha_j = \frac{\alpha^j (1-\alpha)}{1-\alpha^m}. \tag{7}$$

Considering that there is a strong correlation between the bandwidth prediction value and the bandwidth fluctuation, this article needs to choose the appropriate weight and prediction formula according to the size of the network stability factor. The bandwidth prediction formula is expressed as follows:

$$\widehat{B}_{i+1} = \begin{cases} \sum_{j=0}^{m} \alpha_j B_{i-j}, & 0.6 < \alpha < 1, \\ \sum_{j=0}^{m} \alpha_{m-j} B_{i-j}, & \alpha > 1, \\ \frac{1}{m} \sum_{j=0}^{m} B_{i-j}, & \text{others}. \end{cases} \tag{8}$$

When the network stability factor is less than 0.6, it means that the network is in a stable state. The detrimental effects of the glitch's phenomenon can be well reduced using m segments average for prediction. When α is in the other two ranges, it means that the network changes greatly or drastically, the average value of historical data as the bandwidth prediction values may result in a wrong request of bitrate. The segment closer to the current moment has a greater impact on predicting the next segment, so it is given a greater weight value. From (7), it can be obtained that the size of the weights varies with the drastic changes in the network, and we can make predictions adaptively according to the stability of the network.

3.2.3. ABR Control Module. The ABR algorithm is the core of the process and the bitrate chosen directly determines the client's viewing experience. In different scenarios, ABR algorithms have different objectives. In the VoD scenario, there is no buffering requirement. To ensure a better viewing experience for the clients, it will increase or decrease step by step rather than changing suddenly because the buffer is large enough to allow this. However, live streaming requires fewer switches, lower latency, less rebuffering, and higher bitrate within the constraints of a small buffer, and an appropriate bitrate is very difficult to choose.

As with typical bitrate adaptation algorithms, the APSG selects the most appropriate one from the set of available bitrates. The downloaded segments are placed in the playback buffer, and this article defines four buffer thresholds and the maximum playback buffer, which are b_I b_α, b_β, $b_{\max}$. b_{curr} represents the current buffer occupancy. As shown in Figure 2, all thresholds are defined in terms of time. The proposed algorithm uses the joint decision of buffer occupancy and bandwidth prediction to select the bitrate for the next segment.

The details of the APSG algorithm are shown in Algorithm 1. R: $\{r^0, r^1, \cdots, r^{\max}\}$ defined as the set of bitrates available for video in the server. The buffer threshold b_I b_α, b_β, $b_{\max}$, the network stability factor threshold θ, and the set of bitrates available R are initialized. These parameters do not change during each experiment. The bitrate of the next segment is chosen with knowledge of the following parameters: Download the bitrate of the current segment (r^{curr}), the current buffer occupancy(b_{curr}), network stability factor α, the size corresponding to the next available segment of bitrate $s_{i+1}(f) = \{s_{i+1}(0), s_{i+1}(1), \cdots, s_{i+1}(\max)\}$, and the adaptive bandwidth prediction value $\widehat{B}_i$ calculated by (8).

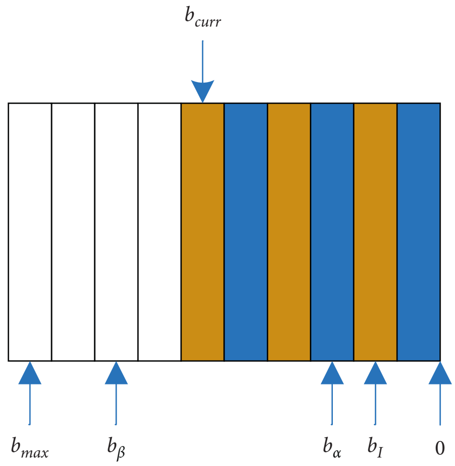


Figure 2: Video playback buffer model.

According to the current buffer occupancy and playback time, the proposed algorithm is divided into four stages, which are described as follows:

Stage 1 ($b_{curr} \le b_I$ or $i < 4$): there is nothing in the playback buffer when the video starts to play; the lowest bitrate is chosen to ensure low initial latency and also to fill the buffer quickly to avoid rebuffering. The lowest bitrate is chosen during the initial playback stage of the video for two purposes: firstly, the currently known sample values for download rates are too few to use the proposed bandwidth prediction method; secondly, it prevents clients from giving up watching the video due to rebuffering.

Stage 2 ($b_{curr} \le b_\alpha$): the algorithm enters stage 2 when the current buffer occupancy is less than b_α. At this stage, the selected bitrate is incremented by one level and is no longer played at the lowest bitrate, improving the quality of the video.

Stage 3 ($b_\alpha < b_{curr} \le b_\beta$): when the playback buffer occupancy is between b_α and b_β, this is the desired stage in this article. At this stage, the dynamic changes in the network environment can be used to determine whether the current network is in a stable state, and if it is in a stable state, the glitches phenomenon can be eliminated based on the choice of bitrate, reducing unnecessary switching of bitrate and improving the viewing experience of the client.

Stage 4 ($b_\beta < b_{curr} \le b_{\max}$): when the playback buffer occupancy exceeds b_β, There is a risk of video content overflow, resulting in a lost frame. The video download is paused until the playback buffer occupancy returns to stage 3 in the traditional method. The bandwidth resources are wasted, and the proposed bandwidth prediction method cannot be used because the download rate cannot be obtained during the pause. So, a playback speed control module is used to avoid causing overflow and wasted bandwidth resources.

3.2.4. Logger Module. The module regularly records various metrics such as bitrate, buffer occupancy, rebuffering, measured bandwidth values, and predicted bandwidth values.

3.2.5. Playback Speed Control Module. The client has a catch-up function that adjusts the playback rate to pull the player back to the target real-time edge. The player can keep itself close to the target latency by controlling the playback rate, which relies on the assumption that changes in playback rate are not significant enough for the end client at 25% or less [26]. It, therefore, speeds up or slows down the playback rate within a range of (0.75, 1.25) depending on the difference between the target latency and the current latency. To determine the actual value in this range, this article relies on the default implementation in the DASH.js player, which uses a sigmoid function.

4. Experimental Result

4.1. Experimental Design

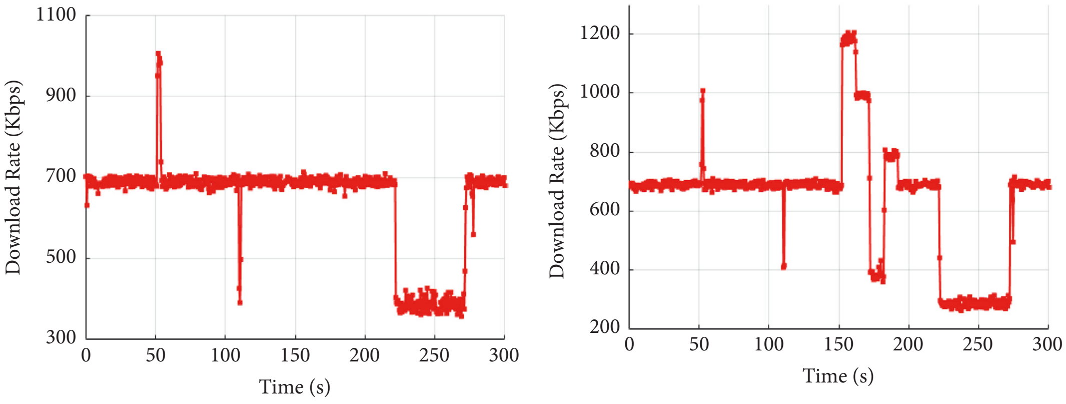
4.1.1. Test Set. As with the existing CMAF-based live servers, this article uses the video sequence Big Buck Bunny [27], encoded using x264 [28] into three different bitrates{360p@200 Kbps, 480p@600 Kbps, 720p@1000 Kbps}. The encoded video was then segmented using MP4Box [29] into 0.5-second segments for DASH and into 0.5-second segments with 33-millisecond chunks for CMAF-DASH. The resulting segments/chunks were used in the DASH.js framework. The whole video sequence intercepted the first 300 seconds of Big Buck Bunny.

4.1.2. Test Platform. In order to build an end-to-end live streaming system, two Ubuntu 20.04 virtual machines were built using personal computers, the first running the DASH.js player in the Google Chrome browser (v97) [30]. Another virtual machine is used to enable the CMAF wrapped FFmpeg encoder and put it into the server. To simulate the network, the network bandwidth is controlled on the server PC using the TC [31] network traffic shaping tool, a module of the Linux kernel, whose control principle is to restrict the transmission of packets at the transport layer so that the data traffic can be shaped.

4.1.3. Bandwidth Configuration. This article is a live streaming scenario with a single client exclusive link bandwidth, and the bandwidth variation is modelled by two scenarios: a stable network and a dynamic network, the details of which are depicted in Figure 3. In a stable network, there are two glitches at 50 and 110 seconds and a sudden decrease and increase in bandwidth at 220 and 270 seconds, respectively. There are two glitches during the dynamic network, and the network fluctuates dramatically after 150 seconds.

```
Data:
$R$: $\{r^0, r^1, \cdots, r^{\max}\}$: Set of available bitrates
$b_I$, $b_\alpha$, $b_\beta$, $b_{\max}$: Buffer thresholds
$\theta$: Network stability factor threshold
INPUT:
$r^{curr}$: Bitrate of the most recently downloaded segment
$b_{curr}$: Current playback buffer occupancy (in seconds)
$\alpha$: Network stability factor
$s_{i+1}(f) = \{s_{i+1}(0), s_{i+1}(1), \cdots, s_{i+1}(\max)\}$ are the $(n+1)th$ segment sizes for bitrates $\{r^0, r^1, \cdots, r^{max}\}$
$\widehat{B}_{i+1}$: The Adaptive bandwidth prediction of the $(i+1)th$ segment
Initialization
if $i < 4$ or $b_{curr} < b_I$;
then
  $l_{i+1} = r^0$;
else
    if $b_{curr} < b_\alpha$;
    then
      $l_{i+1} = r^1$;
  else if $b_{curr} < b_\beta$;
    then
      if $\alpha < \theta$;
      then
        $l_{i+1} = l_i$;
      else
        $l_{i+1} = \max\{r^f | r^f \in R, s_{i+1}(f)/\widehat{B}_{i+1} \le b_{curr} - b_I\}$;
    else if $b_{curr} > b_\beta$;
    then
      $l_{i+1} = \max\{r^f | r^f \in R, s_{i+1}(f)/\widehat{B}_{i+1} \le b_{curr} - b_\alpha\}$;
    end
end
Result:
$l_{i+1}$: The bitrate of the next segment to be downloaded
```

Algorithm 1: Adaptive prediction and smoothing glitches algorithm.



(a)

(b)

Figure 3: Bandwidth configuration. (a) Stable network; (b) dynamic network.

4.1.4. ABR Comparison Algorithm. The proposed APSG was compared with two traditional algorithms, the available bandwidth-based bitrate adaptation algorithm FESTIVE [10] and the buffer-based bitrate adaptation algorithm (BBA) [13]. Bandwidth is predicted by an average of 5 historical data, and then the maximum bitrate less than the predicted value is selected in FESTIVE. The BBA uses a linear mapping function where the bitrate is chosen based on the buffer occupancy.

4.1.5. Performance Metrics. The following performance metrics are prediction errors and QoE parameters.

(1) Prediction errors: the bandwidth prediction errors model is based on Root Mean Square Error (RMSE), which is calculated based on the difference between the bandwidth prediction and the actual bandwidth measurement, as follows:

$$RMSE = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\left(\frac{\widehat{B}_i - B_i}{B_i}\right)^2}. \tag{9}$$

(2) QoE parameters: after each segment is downloaded, this article considers three evaluation metrics of video average bitrate, the number of bitrate switches, and latency to analyze the performance of the proposed algorithms; the rebuffering problem was not considered because these three algorithms did not show cache underflow during the experiment.

Bitrate is one of the most important metrics of the client viewing experience; the higher bitrate brings a better viewing experience for the clients.

$$Q_i^v = \frac{1}{k}\sum_{i=0}^{k} r_i. \tag{10}$$

From the client's perspective, frequent bitrate switching is undesirable. Video is easily abandoned because bitrate switches from high to low; the following formula determines whether the bitrate switching occurs:

$$Q_i^s = \sum_{i=1}^{k-1}\left|r_i - r_{i-1}\right|. \tag{11}$$

VoD streaming has more relaxed latency requirements and can use large playback buffers, whereas live streaming cannot. To maintain interactivity, the most important requirement is low latency. The latency parameter T_d can be obtained directly in the DASH.js player.

In summary, the QoE model is as follows:

$$QoE_i = \mu Q_i^v - \gamma Q_i^s - \varsigma T_d, \tag{12}$$

where μ, γ, ς are the weights used to calculate the QoE. Each weight is given the following values: μ = segment duration; since this article focuses on bitrate switching, a large weight is given to γ, $\gamma = 20$; the latency is expressed in milliseconds in Dash.js, $\varsigma = 5000$. To simplify the representation of QoE, a normalized QoE with a value between 0 and 1 is used, N-QoE (QoE/QoEMAX).

4.2. Bandwidth Prediction Accuracy. In this section, the accuracy of traditional bandwidth prediction methods [10] is compared with that of the adaptive prediction methods in this article. Under the same network environment and parameters, the same ABR algorithm and playback speed control module is used in two prediction methods. Ten experiments were conducted to take the average of each ABR algorithm. However, this article cannot conclude from the observations that the proposed prediction method is better than the traditional method, so there are other aspects to prove that the proposed prediction is more accurate.

Bandwidth prediction error can be calculated by (9), which can directly measure the prediction accuracy at each time. Figure 4 shows the error values at each moment.

According to Figure 4, it can be seen that adaptive prediction produces lower errors in the stable networks at 50 and 110 seconds and 220 and 270 seconds, while at other times, these two errors overlap, shown in blue, because the mean prediction method is used when the network stability factor is less than θ. The same results can be clearly seen under a dynamic network. The proposed prediction method is calculated to reduce the error by 2% compared to the traditional mean prediction error under a stable network and by 5% under a dynamic network. If the network changes more and more dramatically, then the proposed prediction method will be more effective. The proposed prediction method is able to withstand small network changes as well as large ones, in contrast to traditional algorithms that are slower to respond to bandwidth changes because they only consider a fixed sample of historical measurements. Therefore, there is evidence that our prediction method outperforms the traditional method.

4.3. The Glitches Phenomenon. In this section, we verify that APSG can eliminate the glitches phenomenon. Figure 5 shows the bitrate selected results of the three algorithms in the stable network with bandwidth shown in Figure 3(a).

As can be seen from Figures 5(a) and 5(b), the lowest bitrate was chosen as the initial playback bitrate of all three algorithms at the beginning of playback. The glitches phenomenon was eliminated at 50 and 110 seconds, and the same bitrate as the previous segment was selected by the APSG algorithm. In Figure 3(a), the network suddenly changes immediately back to the original network environment, the bitrate selected by the FESTIVE algorithm is switched at 50 and 110 seconds, and small fluctuations occurred in 280 seconds, with bandwidth falling below 600 Kbps, resulting in lower bitrate requested. Figure 5(c) shows that the bitrate is selected by the BBA algorithm and it can be seen that although the glitches are eliminated, it is switched several times between 220 and 270 seconds when the network changes. In a live streaming scenario, the buffer is quite small. Once the network environment changes, the content of the buffer will continue to increase or decrease, resulting in fluctuations around the threshold, and the selected bitrate will be switched many times.

Figure 6 shows the bitrate selected results of the three algorithms in the dynamic network with bandwidth shown in Figure 3(b). As can also be seen in Figures 6(a) and 6(b), the proposed algorithm APSG is able to remove the glitches phenomenon and avoid bitrate switching due to transient

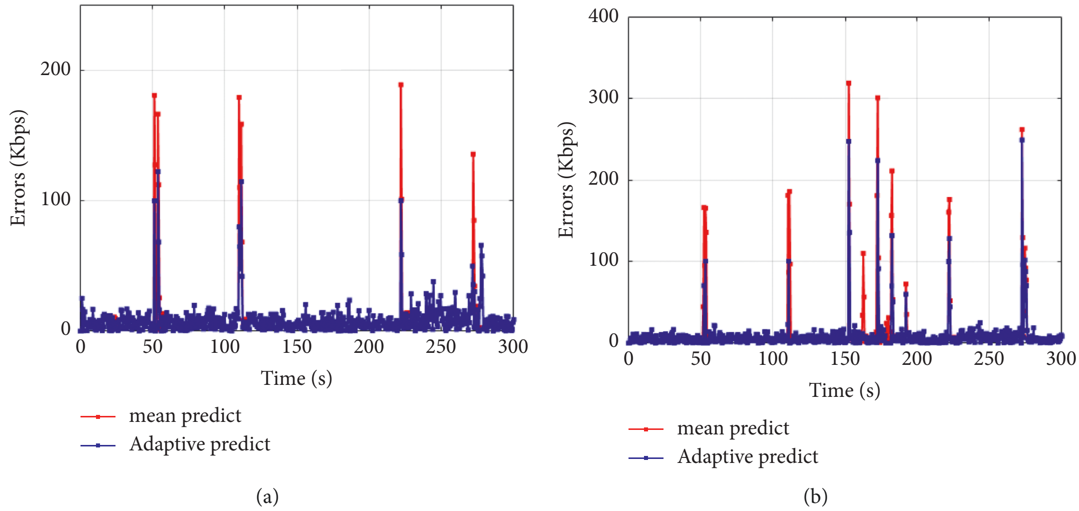


Figure 4: Prediction errors for different prediction methods under two network environments. (a) Stable network prediction errors; (b) dynamic network prediction errors.

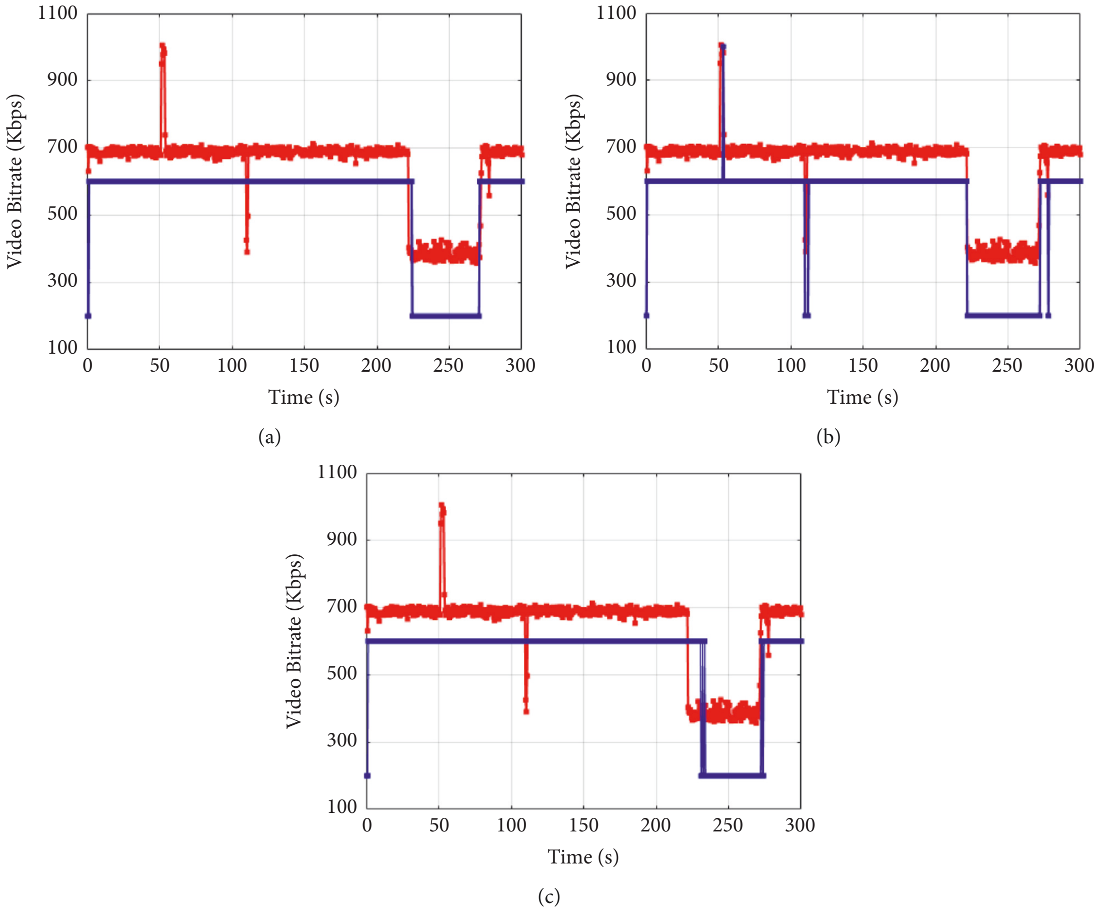
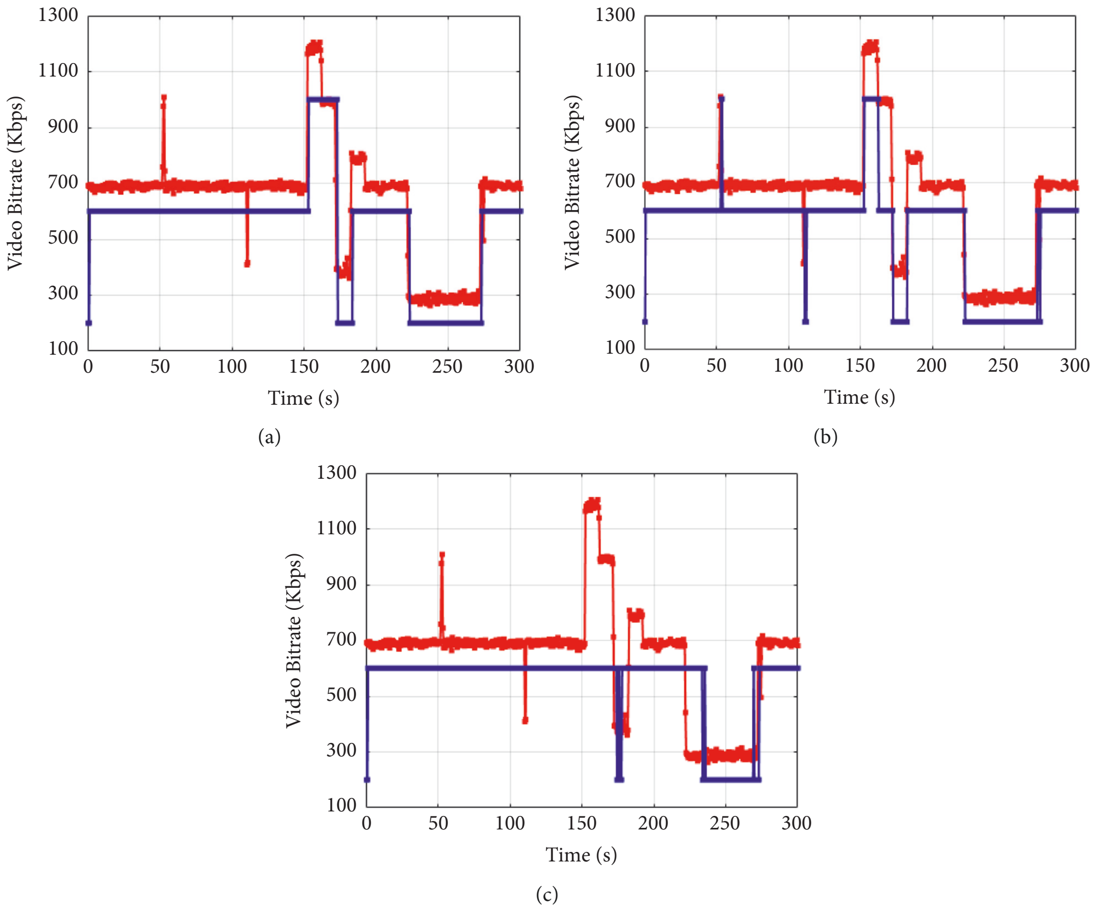


Figure 5: The bitrate selection results during the stable network; the red line represents the bandwidth and the blue line represents the selected bitrate. (a) APSG; (b) FESTIVE; (c) BBA.

bandwidth fluctuations. At about 160 seconds, the network environment becomes worse and the bitrate selection by the FESTIVE algorithm decreases; however, the APSG algorithm still maintains the original bitrate. This is because the bitrate changes less during this period, and the proposed bandwidth prediction method is closer to the real network

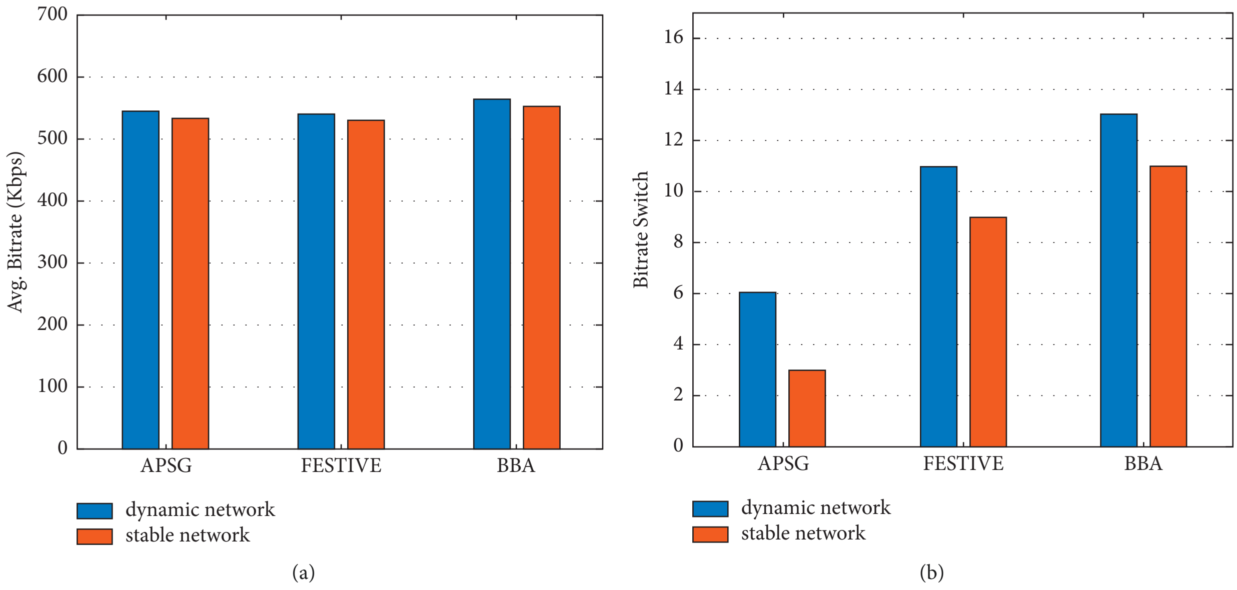


(a)

(b)

(c)

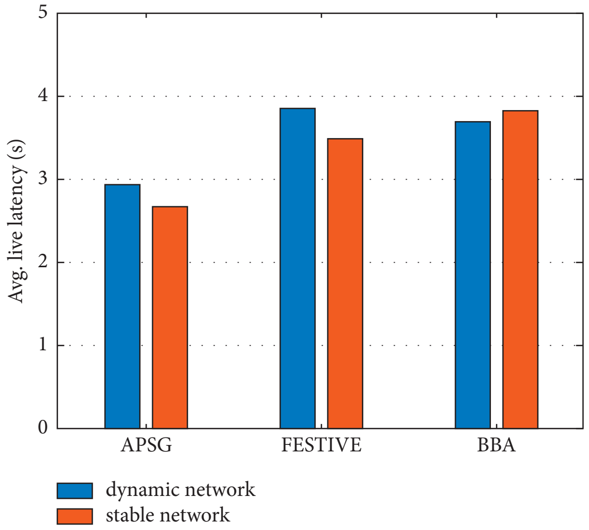
Figure 6: The bitrate selection results during the dynamic network; the red line represents the bandwidth and the blue line represents the selected bitrate. (a) APSG; (b) FESTIVE; (c) BBA.



(a)

(b)

Figure 7: Continued.



(c)

Figure 7: Average bitrate, number of switches, and live latency for different ABR and different networks over 10 runs. (a) Average bitrate; (b) average number of switches; (c) average live latency.

Table 2: Average bitrate, live latency, and QoE with its metrics.

	Avg.Bitrate (Kbps)	Avg.Live Latency (s)	Avg. Switches	Avg.N-QoE
		Stable network		
APSG	531.334	2.68	3	0.98
FESTIVE [10]	528.667	3.47	9	0.85
BBA [13]	552.000	3.69	11	0.85
		Dynamic network		
APSG	544.697	2.94	6	0.98
FESTIVE [10]	540.667	3.85	11	0.86
BBA [13]	562.000	3.83	13	0.85

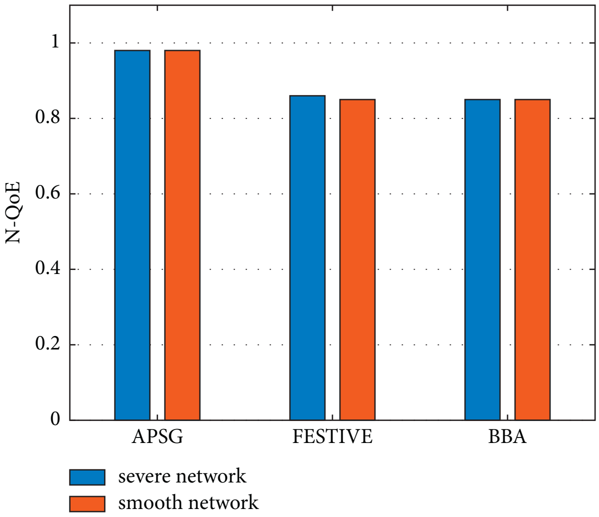


Figure 8: Average N-QoE.

environments, so there is no bitrate switching until the bandwidth is reduced again in 170 seconds. For Figure 7(c), the bitrate selection is similar to the stable network, where the appropriate bitrate is selected based on the buffer occupancy, which tends to ignore the reasonable use of network bandwidth from the playback cache perspective, resulting in wasted bandwidth resources.

4.4. QoE Performance. This section shows the comparison and summary of APSG and two traditional algorithms in QoE metrics. Table 2 and Figure 7 are QoE indicators of the three algorithms under two network environments. In terms of average bitrate, the APSG algorithm compared with the FESTIVE algorithm has little improvement but less than the BBA algorithm, the overall change is not big, and video quality can be considered to remain unchanged. In terms of switching, the APSG algorithm reduced the number of switches relative to the FESTIVE algorithm and the BBA algorithm by 6 and 8 under a stable network and by 5 and 7 under a dynamic network, respectively. In terms of latency, the APSG algorithm reduces it by 0.79 seconds and 1.15 seconds under a stable network and by 0.91 seconds and 0.75 seconds under a server network, respectively. Because this article references the playback speed control module of the DASH.js player, it is able to speed up or slow down the playback speed within range. As can be seen from the above results, unnecessary switching is reduced on the basis of guaranteed bitrate by the APSG algorithm.

Figure 8 shows the N-QoE of different algorithms under different network environments. It can be seen that the algorithm proposed has a higher QoE, while the other two traditional algorithms have a lower overall QoE due to the excessive number of switches. The number of bitrate switches is a key concern in this article, and the QoE obtained by the two traditional algorithms would be lower if the item is given a higher weight.

5. Conclusion

In this article, an adaptive bitrate scheme called APSG is proposed to improve prediction accuracy and eliminate the glitches phenomenon caused by bandwidth fluctuations. The ABR decision relies on three main components: (1) bandwidth measurement with idle time removed; (2) adaptive bandwidth prediction based on the size of network fluctuation; (3) a joint decision algorithm based on bandwidth prediction and buffer occupancy. Results showed that compared to traditional ABR algorithms stable network environment, APSG could reduce the number of bitrate switches and latency by up to 72.6% and 27.3%, respectively, under a dynamic network environment; APSG could reduce the number of bitrate switches and latency by up to 53.8% and 23.6%, respectively, achieving a better video service experience.

Although this work shows good performance in removing glitches, there is room for improvement. Next, we will consider making full use of the various network conditions' scenarios to improve the model and consider the human subjective factor to improve the quality of experience.

Data Availability

The datasets of this work are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflicts of interest in publishing this article.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grants U20A20157, 61771082 and 61871062, the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJQN201900604, the General Program of Chongqing Natural Science Foundation under Grant cstc2021jcyj-msxmX0032, the Natural Science Foundation of Chongqing, China (cstc2020jcyj-zdxmX0024), and the University Innovation Research Group of Chongqing (CXQT20017).

References

[1] V. Cisco, "Cisco visual networking index: forecast and trends, 2017–2022[J]," *White Paper*, vol. 1, no. 1, 2018.

[2] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A survey on bitrate adaptation schemes for streaming media over HTTP," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 562–585, 2019.

[3] K. Hughes and D. Singer, *Information Technology–Multimedia Application Format (MPEG-A)–Part 19: Common media Application Format (CMAF) for Segmented media*, pp. 23000–19, ISO/IEC, Geneva, Switzerland, 2017.

[4] D. Wu, R. Bao, Z. Li, H. Wang, H. Zhang, and R. Wang, "Edge-cloud collaboration enabled video service enhancement: a hybrid human-artificial intelligence scheme," *IEEE Transactions on Multimedia*, vol. 23, pp. 2208–2221, 2021.

[5] D. Wu, X. Han, Z. Yang, and R. Wang, "Exploiting transfer learning for emotion recognition under cloud-edge-client collaborations," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 479–490, 2021.

[6] T. Tang, L. Li, X. Wu et al., "TSA-SCC: text semantic-aware screen content coding with ultra low bitrate," *IEEE Transactions on Image Processing*, vol. 31, pp. 2463–2477, 2022.

[7] A. Bentaleb, C. Timmerer, A. C. Begen, and R. Zimmermann, "Performance analysis of ACTE," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 2s, pp. 1–24, 2020.

[8] I. M. Ozcelik and C. Ersoy, "Low-latency live streaming over HTTP in bandwidth-limited networks," *IEEE Communications Letters*, vol. 25, no. 2, pp. 450–454, 2021.

[9] JS. Dash, "DASH Reference Player," 2019, https://reference.dashif.org/dash.js/.

[10] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 326–340, 2014.

[11] Z. Li, X. Zhu, J. Gahm et al., “Probe and adapt: rate adaptation for HTTP video streaming at scale,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719–733, 2014.

[12] J. van der Hooft, S. Petrangeli, T. Wauters et al., “HTTP/2-Based adaptive streaming of HEVC video over 4G/LTE networks,” *IEEE Communications Letters*, vol. 20, no. 11, pp. 2177–2180, 2016.

[13] Te-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, “A buffer-based approach to rate adaptation: evidence from a large video streaming service,” in *Proceedings of the 2014 ACM Conference on SIGCOMM (SIGCOMM ’14)*, pp. 187–198, NY, USA, October 2014.

[14] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, “BOLA: near-optimal bitrate adaptation for online videos,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1698–1711, 2020.

[15] W. Huang, Y. Zhou, X. Xie, D. Wu, M. Chen, and E. Ngai, “Buffer state is enough: simplifying the design of QoE-aware HTTP adaptive video streaming,” *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 590–601, 2018.

[16] Y. Qin, R. Jin, S. Hao et al., “A control theoretic approach to ABR video streaming: a fresh look at PID-based rate adaptation,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2505–2519, 2020.

[17] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, “A control-theoretic approach for dynamic adaptive video streaming over HTTP,” *ACM SIGCOMM - Computer Communication Review* in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM ’15)*, vol. 45, no. 4, pp. 325–338, NY, USA, October 2015.

[18] A. Sobhani, A. Yassine, and S. Shirmohammadi, “A video bitrate adaptation and prediction mechanism for HTTP adaptive streaming,” ACM trans. Multimedia comput,” *Commun. Appl*, vol. 13, no. 2, 2017.

[19] C. Wang, R. Amr, and M. Zink, “SQUAD: a spectrum-based quality adaptation for dynamic adaptive streaming over HTTP,” in *Proceedings of the 7th International Conference on Multimedia Systems (MMSys ’16)*, Association for Computing Machinery, NY, USA, May 2016.

[20] H. K. Yarnagula, P. Juluri, S. K. Mehr, V. Tamarapalli, D. Medhi, and D. Medhi, “QoE for mobile clients with segment-aware rate adaptation algorithm (SARA) for DASH video streaming,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 15, no. 2, pp. 1–23, 2019.

[21] Y. Sun, “CS2P: improving video bitrate selection and adaptation with data-driven throughput prediction,” in *Proceedings of the 2016 ACM SIGCOMM Conference*, Florianopolis Brazil, August 2016.

[22] H. Mao, R. Netravali, and M. Alizadeh, “Neural adaptive video streaming with pensieve,” in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM ’17)*, pp. 197–210, NY, USA, August 2017.

[23] N. Kan, C. Li, C. Yang, W. Dai, J. Zou, and H. Xiong, “Uncertainty-aware robust adaptive video streaming with bayesian neural network and model predictive control,” in *Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV ’21)*, pp. 17–24, NY, USA, July 2021.

[24] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, “A blockchain-based machine learning framework for edge services in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, 2022.

[25] J. Xiong, R. Bi, Y. Tian, X. Liu, and D. Wu, “Toward lightweight, privacy-preserving cooperative object classification for connected autonomous vehicles,” *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2787–2801, 2022.

[26] M. Kalman, E. Steinbach, and B. Girod, “Adaptive media playout for low-delay video streaming over error-prone channels,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 841–851, 2004.

[27] Ton, “Big Buck Bunny,” 2020, https://peach.blender.org/.

[28] x264, “VideoLAN, a project and a non-profit organization,” 2020, https://www.videolan.org/developers/x264.html.

[29] Gpac, “MP4Box,” 2020, https://gpac.wp.imt.fr/mp4box/.

[30] Chromium, “Chromium Web Browser 2020,” 2020, https://www.chromium.org/.

[31] Ubuntu, “TC Module 2020,” 2020, http://manpages.ubuntu.com/manpages/xenial/man8/tc.8.html.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 5832124, 10 pages
https://doi.org/10.1155/2022/5832124

Research Article

LUNAR: A Practical Anonymous Network Simulation Platform

Xianchun Zheng,[1] **Haonan Yan**,[1] **Rui Wang**,[1] **Ziwei Zhang**,[2] **and Hui Li**[1]

[1] *State Key Laboratory of Integrated Service Network, School of Cyber Engineering, Xidian University, Xi'an, China*
[2] *School of Cyber Science and Engineering, Wuhan University, Wuhan, China*

Correspondence should be addressed to Hui Li; lihui@mail.xidian.edu.cn

Received 25 February 2022; Revised 1 April 2022; Accepted 17 April 2022; Published 4 May 2022

Academic Editor: Jinbo Xiong

With the expansion of cyberspace and the increasing importance of users for privacy protection, anonymous network research has been further developed, especially for the numerous internet of things (IoT) devices. However, when we repeat the existing experiment in an anonymous network, there are many problems such as too old version, low realism, poor operability, and so on. In this paper, we analyzed the design requirements and topology of the new experimental platform. Two topologies with different levels of complexity are designed. We also set up a practical anonymous network simulation platform called LUNAR with virtualization, software-defined networking (SDN), and other technologies to solve those problems. The platform we proposed supports multiprotocol and reproducible complex networks with centralized management. Finally, we implement our simulation platform and reproduce two typical attacks, that is, time-linked Tor node reset attack and website fingerprint attacks on The Onion Router (Tor) network, to evaluate the platform. Experiments results indicate the practicality and superiority of our simulation platform in terms of anonymous network simulation.

1. Introduction

Anonymous network is a privacy protection technology that can protect the private information of the users and the service providers [1, 2]. It can hide the true identity and location of one or both parties in the communication so that the attacker cannot know the sender and receiver of the data, decrypt the plaintext information or associate the transmission of information between the sender and the receiver. The global anonymous network represented by Tor [3] and I2P [4] is widely used in the fields of anonymous access, anonymous communication, and hidden services [5, 6]. These networks have millions of daily users and thousands of relay nodes. Users use anonymous network such as Tor to conduct web pages, browse, online communication, and virtual trade. Due to the characteristics of anonymity, the dark web [7] has two sides. On the one hand, it can be used to protect the privacy of Internet users; on the other hand, it can also be concealed criminal traces or other malicious behavior [8].

1.1. Related Work. Since the birth of the anonymous network, the research on its security about the simulation platform has never stopped [9–15]. For example, [16] proposes a method for connecting IoT devices in a client-server configuration to utilize the Tor network for addressing and secure communication between IoT and CPS devices. Reference [17] introduces a new approach to exploiting Tor's anonymous communication to handle distributed attacks against smart devices on the Internet and demonstrates the effectiveness of Tor in IoT devices. Reference [18] designs a higher bandwidth Onion IoT gateway to provide robust security protection for vulnerable IoT devices hidden behind an IoT gateway. Reference [19] develops, investigates, and evaluates the performance of machine-learning-based darknet traffic detection systems (DTDS) in IoT networks. Reference [20] presents deep learning recurrent LSTM-based technique to classify the traffic over IoT-cloud platforms. Reference [21] realizes resource-conserving access control and end-to-end security for IoT devices and deploys onion routing for the IoT within the well-established Tor network enabling IoT devices to leverage resources to achieve the same grade of anonymity as readily available to traditional devices. SDN [22] can decouple the control plane from the data plane of forwarding devices, and [23] proposes an SDN-based solution to mitigate the privacy threats by anonymizing both MAC and IP addresses. Reference [24] studies ARP spoofing attacks based on SDN technology. However, this experiment is only limited to the ARP protocol and does not discuss other protocols or networks. Reference [25] simulates the dark network scene on the OpenStack platform to detect dark network resources and obtain dark network information. Reference [26] proposes a novel methodology for constructing a real private tor network by editing and controlling Raspberry Pis. However, there is little research on anonymous network simulation platforms design, mostly focusing on improving existing platforms. Reference [27] shows an anonymous network named Crowds. They use Peersim [28] for Crowds anonymous network simulation. In this kind of network, researchers can conduct anonymous tracking. But the author did not carry out a comprehensive design for different forms of networks. Reference [29] designs and implements a new Tor network simulation model based on Shadow [30]. The current simulation platforms have more or less different shortcomings. Some existing platforms are no longer updated, or they are currently no version available to use. Almost all the platform can only simulate or emulate part of establishing a network connection, which is not suitable for various simulation requirements. At the same time, the modification cost is high and has an impact on the simulation results. Table 1 are statistics for Tor network security research.

1.2. Our Work. In this paper, we design and set up a practical anonymous network simulation platform called LUNAR. The overall design principle of LUNAR is to establish an anonymous network simulation platform with convenient operation, large scale, and a high degree of reality under feasible resource constraints. Besides, the platform can ensure the correctness of the Tor network simulation operation, ensure that the virtual nodes can be linearly expanded, and form a large-scale simulation platform.

Our main contributions are as follows:

(1) We summarize all experiments conducted on anonymous networks and classify them into five categories in Section 2.1, from which we find six network topology requirements of the simulation platform in Section 2.3.

(2) We provide five design requirements for the current new anonymous network simulation platform in Section 2.2.

(3) We devise two kinds of an anonymous network topology for our simulation platform, that is, a basic one for a single small-scale experiment and a more complex one for a complex anonymous network experiment in Section 3.1. Our proposed simulation platform is also compared with five previous ways to conduct anonymous network experiments in Section 2.1.

(4) We implement our highly scalable simulation platform and reproduce two types of typical attacks, that is, time-linked Tor node reset attacks and website fingerprint attacks on the Tor network, on our simulation platform in Section 4. Experiments indicate our simulation platform can provide a practical anonymous network environment.

2. Problem Description

2.1. Compare with Existing Platform. The existing methods for conducting anonymous network experiments are roughly divided into several categories. Table 2 is the comparison of the advantages and disadvantages of five experimental platforms.

(1) Experiments in real networks. The advantage is that the threshold for conducting small-scale experiments is low, and the realism is high. But it can only provide local experiment results, observation angles, and control methods, and it is impossible to implement a global experiment scenario [31].

(2) Experiments with a large research network such as PlanetLab [32]. It consists of more than 1,000 servers distributed around the world. It can apply to several servers to form a network segment and deploy experiment software on it. But the operability is still limited, and experiments can only be carried out in a limited time frame and scale [31].

(3) Simulation experiments, such as ExperamenTor [33]. It uses ModelNet [34] as a virtual machine and network simulation to realize the experiment environment. Although the efficiency is partially improved, because of the age, we cannot download the available version.

(4) Emulation test, such as TorPs [35]. It simulates the process of the Tor network selection relay node to establish a link. TorPs is suitable for experiments related to improving or changing the link selection algorithm, but not applicable to operational experiments.

(5) Semi-simulation and semi-emulation, such as Shadow [30]. It implemented a network layer emulation and application layer simulation test environment. However, the use of Shadow as a research platform requires a certain amount of modification cost, and the modification will directly affect the efficiency, accuracy, and intuitiveness of the experiment results.

Compared with our design, these simulation platforms make different trade-offs in extensibility, global control, operability, and cost, which cannot meet the higher requirements of modern new experiments. For example, Shadow implements a low-level takeover to meet scalability, but it is less efficient.

Table 1: Statistics for Tor network security research.

Research	Attack	Defense
Link attack	Censorship, BGP attack, path selection, bridge attack	Censorship avoidance, timing-based avoidance, path selection algorithm, guard node selection algorithm
Traffic analysis	Bridge discovery attack, replay attack, man-in-the-middle attack, traffic correlation attack	Data mining, trust mechanism
Website fingerprinting attack	Based on website characteristics, based on website cache	Service site redesign, cache mask
Others	Side channel attack, DDoS, information leakage	Privacy information protection

Table 2: Comparison of the advantages and disadvantages of five experimental platforms.

Methods	Typical representative	Advantages	Disadvantages
Real networks	Real tor network	Low threshold and high realism	Cannot control the whole, high cost of large-scale case
Global research network	PlanetLab	Apply for several servers to form a network segment and deploy experiment software	Operability, time frame, and scale are limited
Simulation experiments	ExperamenTor	High overall control and low experimental deployment cost	No available version
Emulation test	TorPs	Suitable for experiments related to improving or changing the link selection algorithm	Unable to analyze traffic and test security
Semi-simulation and semi-emulation	Shadow	Offering a network layer emulation and application layer simulation test environment	High modification cost and modification affect results

2.2. Design Goals. Combined with the existing simulation platform analysis, the new platform needs to meet the following design requirements:

(1) Support multiprotocol. The experiment platform should support the environment reproduction of Tor, I2P, Freenet, and other anonymous networks designed or modified by themselves and can test and demonstrate the functions, performance, and security of these protocols.

(2) Support native code. The simulation platform should directly install and run the original code or installation package of each anonymous network software and its modified variants. Ensure the accuracy of the simulation results.

(3) Reproduce network conditions. The simulation platform can simulate different link bandwidths and network congestion, support background traffic, and support network monitoring, interference, and control at different levels (node level, network segment level, and self-made domain level). This satisfies the researchers' needs for each dimension of the network layer.

(4) Save resources. Although the current hardware resource cost is getting lower and lower, the simulation platform's node size has also been greatly improved. As a result, it is still necessary to prevent the hardware cost from exploding with the scale expansion, making the demand linear with the node size.

(5) Centralized control and observation. In order to achieve the global perspective simulation, the platform needs to have centralized management configuration means and a unified result observation method, which can be used by researchers to use the platform to carry out experiments to lower the threshold and provide convenience.

2.3. Platform Functions. Experiments on standard anonymous networks can be summed up in the following types.

2.3.1. Performance Analysis. Tor's anonymous network has been criticized for its network performance due to the particularity of its protocol [36, 37]. Researchers have been looking for ways to improve the Tor network's overall performance, either redesigning the protocol and architecture, optimizing Tor's own weighted link selection algorithm, or enhancing its handling of network congestion, packet encryption, and decryption at the application layer.

2.3.2. Passive Listening. In passive listening, the attacker can control the critical location nodes such as the malicious guard node and exit node, hijack the traffic, carry out the relevant time attack, fingerprint attack, and other attack means to achieve the goal of deanonymization. Traffic feature extraction and matching are often vital in passive listening.

2.3.3. Active Interference. Active interference refers to the attacker's artificial change of the link establishment process, traffic path selection, and other key points to remain anonymous. For example, an attacker at the AS (autonomous system) level can change the network traffic of the Internet to achieve more granular traffic filtering by implementing BGP hijacking and asymmetric traffic analysis to obtain more accurate analysis results.

2.3.4. Denial of Service. The denial of service is an attack on the Tor network itself. The target of these attacks is often the relay node. Since the overhead of the relay node processing the data packet is much larger than the overhead of generating the data packet, the attacker can continuously send the CREATE data packet to consume the relay node's computing resources, thereby reducing the performance of the entire Tor network.

2.3.5. Application Layer Induction. In addition to the underlying protocol in the overall composition of the Tor network, the entire application layer system also exists in the surface web website. Browsers may have loopholes and problems, so it is also necessary to test and audit the anonymous network's service provider. Security vulnerabilities may occur on it.

3. Our Proposed Platform

3.1. Topology Design. We summarize the common anonymous network attack and defense experiment scenarios, extract the rule conditions that satisfy them, and design a standard network topology, which can project various simulation scenes into the topology without distortion. We use a white spot as the terminal node and use a black spot as the basic network node. Therefore, we design a basic topology that meets different conditions.

The topology shown in Figure 1 consists of N routing nodes and N terminal nodes. The routing nodes are connected end to end to form a ring network. Each routing node is externally connected with a terminal node. The routing node acts as the control node of the termination node. The control node of the point can realize monitoring and flow control. Adjacent routing nodes can form an autonomous system to reproduce the characteristics of the anonymous network in the home domain. After the ring network is destroyed on a link, another half of the line can be dynamically routed. This topology simplifies the network structure and node relationships but retains the various network features required for anonymous network experiments and can be used for a single small-scale experiment.

In addition, we have designed a more complex network topology, as shown in Figure 2. The topology separates the routing node used to monitor the terminal traffic from the basic routing node used to transmit information and is connected by an N-route node in a star topology to form a basic transmission network, each of which is on the basic routing node. A routing node is dedicated to the control node, and M terminal nodes are connected to the control node to form an autonomous system. In the experiment, the basic routing nodes are not touched, and only the terminal nodes and control nodes required by the simulation environment are operated. This topology is closer to the real Internet environment, ensuring the availability of the underlying network for complex anonymous network experiments.

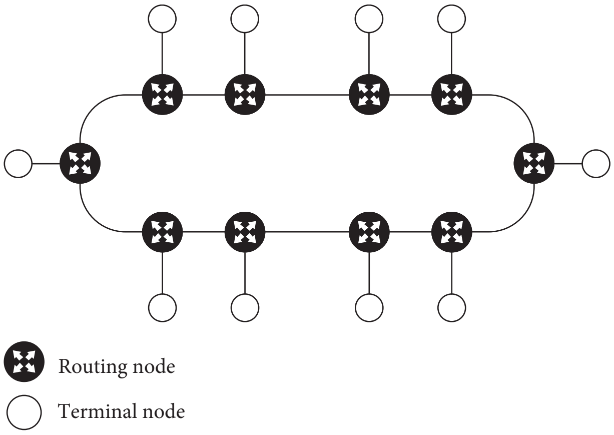


Figure 1: The basic topology of our platform.

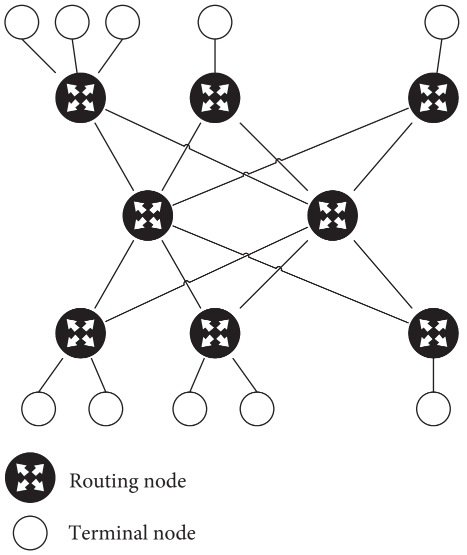


Figure 2: A more complex topology of our platform.

The focus of our design is not on traffic collection and stratification but on the design of the topology. Traffic in the protocol layer is the problem of upper-layer applications. Further research applications such as Tor can be deployed on our proposed topology.

3.2. Advantages. Our platform's network topology has the following advantages:

(1) It contains the basic network node responsible for transmission. The basic network node is equivalent to the infrastructure on the Internet and is only responsible for forwarding network packets, not participating in the encryption and decryption of anonymous networks.

(2) It runs anonymous network programs in the terminal nodes. Terminal nodes may assume the role of

dark web infrastructure such as hidden service directory, may also act as users of anonymous networks, run anonymous service-side programs or client services, and may also be used as a simulation of various services in the surface web so that experimenters from the anonymous network to initiate access to them, such as DNS, Web Server, etc.

(3) It interworks between any two nodes. The same experiment instance is performed in an interworking network. If multiple simulation experiments are required at the same time, numerous network instances can be pulled up.

(4) No single node can carry all the traffic in the network. The simulation platform should avoid a situation where the entire network is down due to the failure of the single node that carries all the traffic.

(5) The communication between any two terminal nodes has no less than two routing lines. The network infrastructure under the dark web (usually the Internet) has certain robustness and will not be faulty due to a single node. And it leads to the whole network.

(6) All communication traffic of any terminal node needs to flow through at least one corresponding basic network node. In the simulation experiment for the anonymous network, there may be scenarios in which the traffic of the terminal node is monitored or controlled in the system (rather than being handled by the node itself), such as being controlled by the operator. The node in this topology can implement this condition, which we call the control terminal node.

4. Evaluation

As the simulation platform is virtualized by KVM (Kernel-based virtual machine) and SDN network layer, each node's operation is consistent with that of the actual Tor network. No additional functions or redundant configuration is required. Log in to the corresponding node shell to perform steps such as compiling and running Tor, starting the Tor process, and changing the Tor configuration. This section mainly introduces the platform's feasibility and superiority by recreating the simulation approach of two typical attacks on our testing platform.

4.1. Experiment Setup. In order to achieve the design goals of the topology and simulation platform, we choose to use a host virtualization technology to create virtual nodes on a set of server clusters and use SDN technology to connect nodes according to the designed topology to form a virtual network, running different software. The network traffic sent by the virtual machine is transmitted by the SDN controller, using the same communication protocol (IP, TCP, and UDP) as the real network, and finally sent to receive the traffic. The design of the simulation platform is shown in Figure 3.

The simulation platform is built on a server cluster connected by LAN. KVM is used as the virtual machine controller; OVS (Open vSwitch) is used as the SDN switch; and CentOS is used as the operating system on both the server and the virtual machine. The foundation of the experimental platform is composed of KVM, OVS, and CentOS. The routing node is responsible for data forwarding at the network layer, and the complete Tor process is run in the terminal node. For different Tor roles, such as directory servers, we can flexibly allocate more resources to meet its computing and storage requirements.

The version of Tor selected for this paper is tor-0.3.4.8. System version is CentOS, and Linux version is 3.10.0-693.el7.x86 64. The simulation platform designed in this paper provides a related server and virtual machine images.

The platform provides a set of scripting tools, as the management layer of the simulation platform, which realizes the generation and configuration of the virtual machine and virtual network topology, pulls up and manages instances of the simulation environment, controls virtual machine terminals, and changes network bandwidth and congestion.

Another set of scripts implements the business layer of the simulation platform. It can install and configure the basic components of classic anonymous networks such as Tor, I2P, and Freenet and build an independent and complete anonymous network simulation environment. On this basis, the experimenter can also log in to the virtual machine to install other servers or client software and access the anonymous network. Figures 4 and 5 are examples of OnionRoute and HiddenService configuration.

Finally, the platform provides an observation layer. The participants can obtain the control of the corresponding terminal node or routing node according to the scenario setting and role assignment and specify the network traffic mirroring and log information of the location.

4.2. Time-Linked Tor Node Reset Attack

4.2.1. Attack Fundamentals. We divide the link in the process of communication between the client and the anonymous server into several parts for the subsequent explanation. The first is the link from the client to the rendezvous node through the 2-hop node, which we collectively call the CR link part. The link from the anonymous server to the rendezvous node through the 3-hop node is collectively referred to as the HR part.

In the dark web environment, only the hidden server's entry node knows the real IP of the anonymous service. Suppose we currently have a certain number of entry nodes, clients, and rendezvous nodes.

The attack is divided into three steps:

(1) Open the Tor2Web mode at the client we control. Select the RP that we control as the rendezvous node. The client sends a request to a specific anonymous server continuously for a certain period, and the anonymous server further establishes long-link communication with the client.

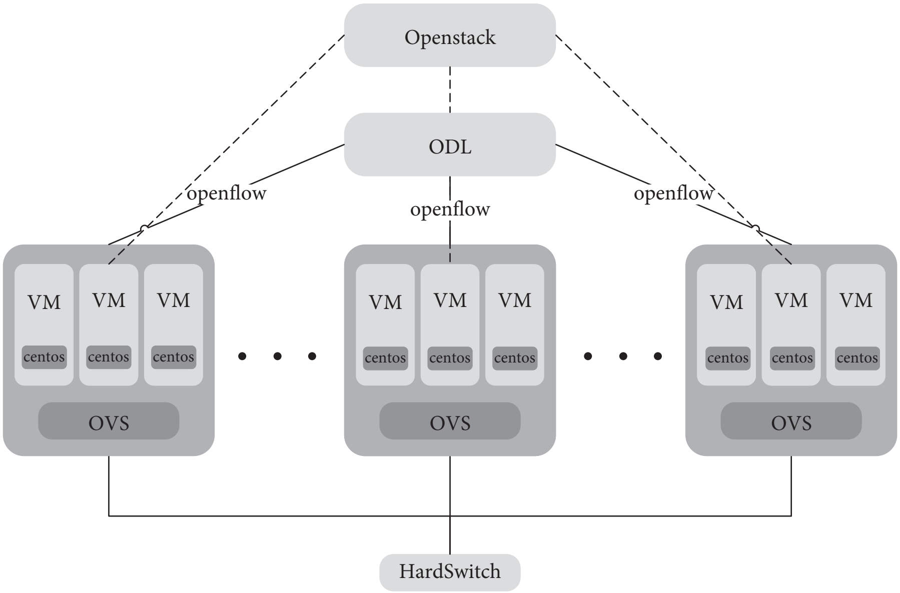
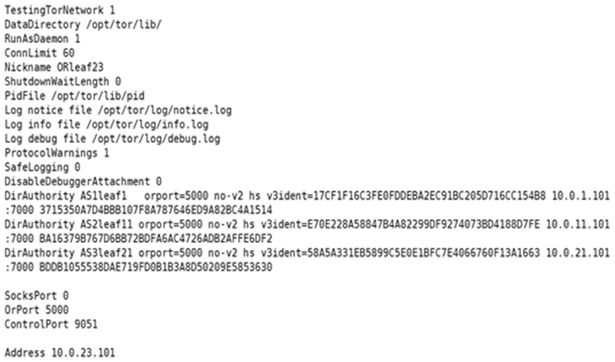


Figure 3: Simulation platform design.

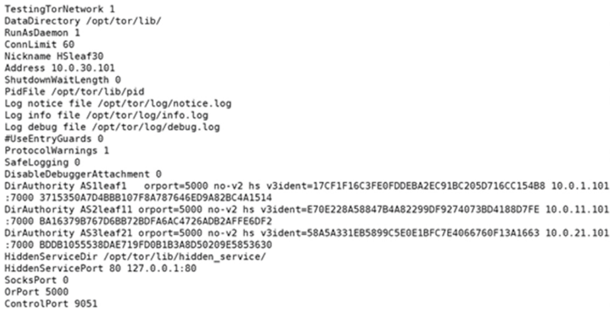


```
TestingTorNetwork 1
DataDirectory /opt/tor/lib/
RunAsDaemon 1
ConnLimit 60
Nickname ORleaf23
ShutdownWaitLength 0
PidFile /opt/tor/lib/pid
Log notice file /opt/tor/log/notice.log
Log info file /opt/tor/log/info.log
Log debug file /opt/tor/log/debug.log
ProtocolWarnings 1
SafeLogging 0
DisableDebuggerAttachment 0
DirAuthority AS1leaf1   orport=5000 no-v2 hs v3ident=17CF1F16C3FE0FDDEBA2EC91BC205D716CC154B8 10.0.1.101
:7000 3715350A7D4BBB107F8A787646ED9A82BC4A1514
DirAuthority AS2leaf11 orport=5000 no-v2 hs v3ident=E70E228A58847B4A82299DF9274073BD4188D7FE 10.0.11.101
:7000 BA16379B767D6BB72BDFA6AC4726ADB2AFFE6DF2
DirAuthority AS3leaf21 orport=5000 no-v2 hs v3ident=58A5A331EB5899C5E0E1BFC7E4066760F13A1663 10.0.21.101
:7000 BDDB1055538DAE719FD0B1B3A8D50209E5853630

SocksPort 0
OrPort 5000
ControlPort 9051

Address 10.0.23.101
```

Figure 4: Example of OnionRoute configuration.



```
TestingTorNetwork 1
DataDirectory /opt/tor/lib/
RunAsDaemon 1
ConnLimit 60
Nickname HSleaf30
Address 10.0.30.101
ShutdownWaitLength 0
PidFile /opt/tor/lib/pid
Log notice file /opt/tor/log/notice.log
Log info file /opt/tor/log/info.log
Log debug file /opt/tor/log/debug.log
#UseEntryGuards 0
ProtocolWarnings 1
SafeLogging 0
DisableDebuggerAttachment 0
DirAuthority AS1leaf1   orport=5000 no-v2 hs v3ident=17CF1F16C3FE0FDDEBA2EC91BC205D716CC154B8 10.0.1.101
:7000 3715350A7D4BBB107F8A787646ED9A82BC4A1514
DirAuthority AS2leaf11 orport=5000 no-v2 hs v3ident=E70E228A58847B4A82299DF9274073BD4188D7FE 10.0.11.101
:7000 BA16379B767D6BB72BDFA6AC4726ADB2AFFE6DF2
DirAuthority AS3leaf21 orport=5000 no-v2 hs v3ident=58A5A331EB5899C5E0E1BFC7E4066760F13A1663 10.0.21.101
:7000 BDDB1055538DAE719FD0B1B3A8D50209E5853630
HiddenServiceDir /opt/tor/lib/hidden_service/
HiddenServicePort 80 127.0.0.1:80
SocksPort 0
OrPort 5000
ControlPort 9051
```

Figure 5: Example of HiddenService configuration.

(2) Select to turn off the Tor service at the RP, and the HS will disconnect the HR link at this time.

(3) Through our client to access a specific anonymous server again to send a continuous request within a certain period of time, re-establish the CR and HR long link. However, the anonymous server at this time has already established an HR of three hops different from the previous one, and thus, the selected entry node has also changed.

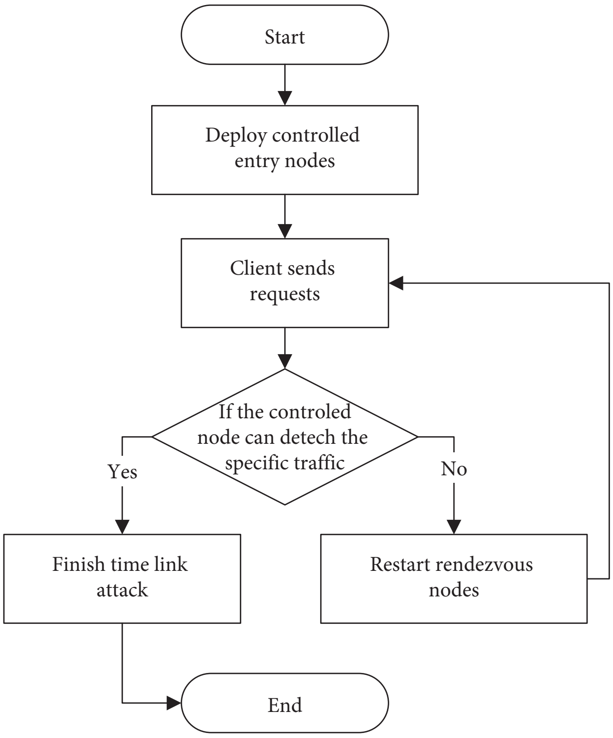
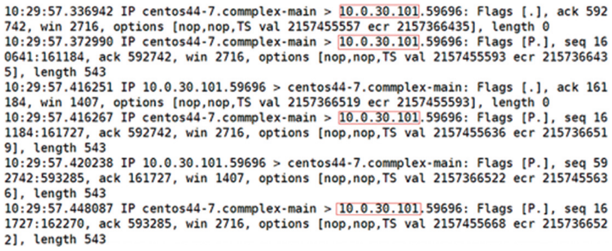

Start
Deploy controlled entry nodes
Client sends requests
If the controled node can detech the specific traffic
Yes
No
Finish time link attack
Restart rendezvous nodes
End

Figure 6: Time-linked Tor node reset attack flowchart.

Since we have a certain number of entry nodes, we repeat step 2 and 3 until a specific anonymous server connects to the entry node we control.

Finally, according to the characteristics of packet time association and continuity, the IP of the anonymous server is determined so as to achieve the effect of the anonymous server to anonymity. Figure 6 shows the overall flowchart.



```
10:29:57.336942 IP centos44-7.commplex-main > 10.0.30.101.59696: Flags [.], ack 592
742, win 2716, options [nop,nop,TS val 2157455557 ecr 2157366435], length 0
10:29:57.372990 IP centos44-7.commplex-main > 10.0.30.101.59696: Flags [P.], seq 16
0641:161184, ack 592742, win 2716, options [nop,nop,TS val 2157455593 ecr 215736643
5], length 543
10:29:57.416251 IP 10.0.30.101.59696 > centos44-7.commplex-main: Flags [.], ack 161
184, win 1407, options [nop,nop,TS val 2157366519 ecr 2157455593], length 0
10:29:57.416267 IP centos44-7.commplex-main > 10.0.30.101.59696: Flags [P.], seq 16
1184:161727, ack 592742, win 2716, options [nop,nop,TS val 2157455636 ecr 215736651
9], length 543
10:29:57.420238 IP 10.0.30.101.59696 > centos44-7.commplex-main: Flags [P.], seq 59
2742:593285, ack 161727, win 1407, options [nop,nop,TS val 2157366522 ecr 215745563
6], length 543
10:29:57.448087 IP centos44-7.commplex-main > 10.0.30.101.59696: Flags [P.], seq 16
1727:162270, ack 593285, win 2716, options [nop,nop,TS val 2157455668 ecr 215736652
2], length 543
```

Figure 7: Time-linked Tor node reset attack result.

Table 3: Extracted features.

Feature name	Feature description
pkt_length	The overall packet size
fwd_pkt_length	The size and number of packets sent
bwd_pkt_length	The size and number of packets received
fwd_pkt_num_p	The proportion of packets sent as a percentage of the overall packet
bwd_pkt_num_p	The proportion of the accepted packet stake in the overall packet
nc_length_mv	The mean variance of the length of all forward packets before the flow changes direction
ncd_num_mv	The mean variance of the number of all forward packets before the flow changes direction
pkt_length_per_sec	The length of the packet per second
fwd_first_30_pkt_length	The length of the first 30 packets sent
bwd_first_30_pkt_num	The length of the first 30 packets received
Duration	The total transfer time

4.2.2. Reproduction Process. First, we modify the client and hidden server configuration files to enable the client Tor2-web mode. Also, specify the rendezvous node. After the client and hidden server configuration are completed, continuous requests are sent to the hidden server through the client for a certain period of time. Then use tcpdump crawl traffic data at all controllable portal nodes as well as at the client. Next, reset the Tor service at the rendezvous node, forcing the anonymous server to send a DESTROY instruction to destroy the link.

Repeat the reset service, and the client sends a continuous request for the steps so that the hidden server continuously destroys, re-selects the entry node, and re-establishes the link. Observing the traffic at the controllable node, in order to make the observation effect better, it is necessary to filter out the traffic of other interference items such as the directory server synchronization descriptor. The attack is stopped until the hidden server selects the controllable entry node as a node. Eventually, we can find the real IP (10.0.30.101) of the hidden server and complete the attack. Figure 7 shows the successful attack result.

4.3. Website Fingerprint Attacks on Tor Networks

4.3.1. Attack Fundamentals. Anonymous communication hides the address of the source and destination. The layered encryption of the traffic allows the attacker to detect the content information of the traffic. However, data traffic still has other dimensions. These features can be used to attack the anonymous system.

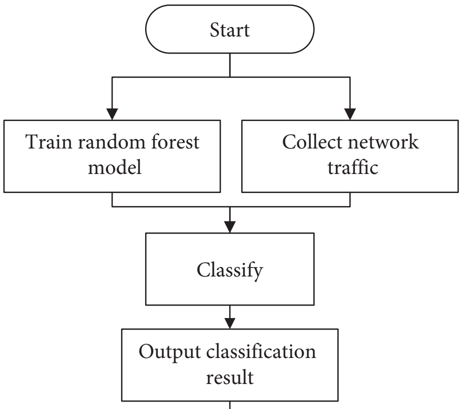


Figure 8: Website fingerprint attacks flow.

In order to achieve the purpose of the attack, the attacker first needs to obtain the fingerprint information of N websites, collect the traffic when visiting *N* websites, set up feature vectors, and train the classifier by using machine learning algorithms. When the target client accesses the website, the client can obtain the client traffic, and the import classifier determines whether the target client has accessed one or several of the *N* websites according to the classification result. Therefore, the prerequisite for a successful website fingerprint attack is to be able to collect traffic from the client. The attacker is required to be an ISP-level adversary.

Table 4: Classification test results.

Website	Number of client visits	The correct number of classifications	Accuracy
QQ	32	26	0.813
Tmall	26	22	0.85
Taobao	20	187	0.9
Sohu	8	67	0.75
JD	27	26	0.96
Weibo	6	57	0.83
Sina	33	29	0.88
360	10	87	0.8
Csdn	26	227	0.85
Bilibili	12	12	1

4.3.2. Reproduction Process. In this experiment, the Alexa China Top100 website was selected for fingerprint collection.

First, the local client opens Tor, connects to the Tor network, and accesses the above website. In the entry node, tcpdump is simulated by the ISP-level attacking attacker to grab the traffic sent and received by the client. The experiment will access each website 100 times with a random interval in between. The data set is the time delay and size of traffic data packets visiting the corresponding website. The extracted feature is a combination of these delays and sizes, such as the delay of the first 300 data, the size of the first 300 packets, and so on.

The selection of fingerprint features in this experiment is crucial for the entire attack because the selection characteristics largely determine the correct rate of classifier classification. Based on previous researchers' fingerprint attack experience on anonymous systems, this topic selects the extracted features shown in Table 3.

With the random forest algorithm using the Scikit-learn library in python, the data set is handed over to our classifier. The training is done by means of a tenfold cross-validation method. The ten-fold cross-validation method is used in the training, that is, the data set is divided into ten equal parts, one of which is used as the test set, and the rest are used for training and evaluation, so as to make the classification effect of our classifier better.

We have already built the classifier in the previous work, and then we will introduce the overall attack process. The flow chart of the attack process is shown in Figure 8. Randomly visit the website at the client to simulate normal user behavior and listen to traffic at the ingress route to simulate ISP-level adversaries. The monitored user traffic is processed by our script to extract features. The feature data is then passed to the classifier classification to obtain an output.

To test the accuracy of our classification, assume that the adversary is only interested in the Alexa Top10 website. Control the client to randomly access the website in Alexa Top10 for a total of 200 times, and the monitored traffic data is repeated. Finally, the statistical results in Table 4 are obtained.

The success rate of using fingerprint attacks on different websites to destroy client anonymity is basically above 80%, indicating that this attack is established in the Tor network.

Our design can meet the needs of different applications, and the experimental platform has good scalability. Users can quickly understand and familiarize themselves with the method of use, and the experimental platform has good ease of use. Compared with the existing research platform, our design has good generality.

5. Conclusion

This paper designs a new anonymous network research platform, which reduces the research threshold for anonymous networks and is suitable for comprehensive and efficient simulation of real anonymous network research. The research platform uses virtualization to quickly build the basic nodes in the anonymous network, avoiding deploying a large number of group hardware facilities in the network. The platform uses SDN, which can be defined and controlled by software programming, simplifying the network. The steps of network deployment meet the needs of the anonymous network for complex network environments. The platform can also deploy and run anonymous Tor source codes and other anonymous network source codes to ensure the network simulation operation's correctness and ensure that the virtual nodes can be linearly expanded. Finally, the paper highlights the platform's feasibility and superiority by processing two typical attacks on the platform. In the future, we will focus on technical research of anonymous networks in the platform, seeking to identify possible anonymous network vulnerabilities and implementing improvements.

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. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 571–588, 2002.

[2] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE*

Transactions on Industrial Informatics, vol. 17, no. 2, pp. 922–933, 2021.

[3] R. W. Gehl, *Weaving the Dark Web: Legitimacy on Freenet, Tor, and I2P*, MIT Press, Cambridge, MA, USA, 2018.

[4] P. Iacoban, *Measuring Accessibility of Popular Websites while Using the I2p Anonymity Network*, Delft University of Technology, Delft, Netherlands, 2021.

[5] A. Biryukov, I. Pustogarov, and R.-P. Weinmann, "Trawling for tor hidden services: detection, measurement, deanonymization," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, pp. 80–94, IEEE, Berkeley, CA, USA, May, 2013.

[6] D. Kavallieros, D. Myttas, E. Kermitsis, E. Lissaris, G. Giataganas, and E. Darra, "Understanding the dark web," in *Dark Web Investigation*, pp. 3–26, Springer, New york, NY, USA, 2021.

[7] K. Finklea, "Dark Web, Special Report for Congressional Research Service," R44101, 2015.

[8] J. Weber and E. W. Kruisbergen, "Criminal markets: the dark web, money laundering and counterstrategies - an overview of the 10th Research Conference on Organized Crime," *Trends in Organized Crime*, vol. 22, no. 3, pp. 346–356, 2019.

[9] J. Xiong, R. Bi, Y. Tian, X. Liu, and J. Ma, "Security and privacy in mobile crowdsensing: models, progresses, and trends," *Chinese Journal of Computers*, vol. 44, no. 09, pp. 1949–1966, 2021.

[10] W. Yu, X. Fu, S. Graham, X. Dong, and W. Zhao, "Dsss-based flow marking technique for invisible traceback," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07)*, pp. 18–32, IEEE, Berkeley, CA, USA, May 2007.

[11] X. Wang and S. Douglas, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pp. 20–29, Washington D.C. USA, October, 2003.

[12] X. Wang, S. Chen, and S. Jajodia, "Network flow watermarking attack on low-latency anonymous communication systems," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07)*, pp. 116–130, IEEE, Berkeley, CA, USA, May, 2007.

[13] D. Agrawal, D. Kesdogan, and S. Penz, "Probabilistic treatment of mixes to hamper traffic analysis," in *Proceedings of the 2003 Symposium On Security And Privacy*, pp. 16–27, IEEE, Berkeley, CA, USA, May, 2003.

[14] M. Liberatore and B. N. Levine, "Inferring the source of encrypted http connections," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 255–263, Alexandria Virginia USA, October, 2006.

[15] T. G. Abbott, K. J. Lai, M. R. Lieberman, and E. C. Price, "Browser-based attacks on tor," in *International Workshop on Privacy Enhancing Technologies*, pp. 184–199, Springer, New york, NY, USA, 2007.

[16] F. W. Baumann, U. Odefey, S. Hudert, M. Falkenthal, and U. Breitenbücher, "Utilising the tor network for iot addressing and connectivity," in *Proceedings of the 8th International Conference on Cloud Computing and Services Science*, pp. 27–34, Madeira, Portugal, March, 2018.

[17] N. Phong Hoang and D. Pishva, "A tor-based anonymous communication approach to secure smart home appliances," in *Proceedings of the 2015 17th International Conference on Advanced Communication Technology (ICACT)*, pp. 517–525, IEEE, PyeongChang, Korea (South), July, 2015.

[18] L. Yang, C. Seasholtz, B. Luo, and F. Li, "Hide your hackable smart home from remote attacks: the multipath onion iot gateways," in *European Symposium on Research in Computer Security*, pp. 575–594, Springer, New york, NY, USA, 2018.

[19] Q. Abu Al-Haija, M. Krichen, and W. Abu Elhaija, "Machine-learning-based darknet traffic detection system for iot applications," *Electronics*, vol. 11, no. 4, p. 556, 2022.

[20] S. Patil and L. A. Raj, "Classification of traffic over collaborative iot and cloud platforms using deep learning recurrent lstm," *Computer Science*, vol. 22, no. 3, 2021.

[21] J. Hiller, P. Jan, M. Dahlmanns, H. Martin, A. Panchenko, and K. Wehrle, "Tailoring onion routing to the internet of things: security and privacy in untrusted environments," in *Proceedings of the 2019 IEEE 27th International Conference on Network Protocols (ICNP)*, pp. 1–12, IEEE, Chicago, IL, USA, October, 2019.

[22] R. Amin, M. Reisslein, and N. Shah, "Hybrid sdn networks: a survey of existing approaches," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3259–3306, 2018.

[23] T. Wong, H. Cui, Y. Shen, W. Lin, and T. Yu, "Anonymous network communication based on sdn," in *Proceedings of the 2018 4th International Conference on Universal Village (UV)*, pp. 1–5, Boston, MA, USA, October, 2018.

[24] H. Aldabbas and R. Amin, "A novel mechanism to handle address spoofing attacks in sdn based iot," *Cluster Computing*, vol. 24, no. 4, pp. 3011–3026, 2021.

[25] P. Wang, H. Liu, B. Wang, K. Dong, L. Wang, and S. Xu, "Simulation of dark network scene based on the big data environment," in *Proceedings of the International Conference on Information Technology and Electrical Engineering 2018*, October, 2018.

[26] Q. Wang and W. Cao, "A tor anonymity attack experiment platform driven by raspberry pi," in *Proceedings of the 2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan)*, pp. 569–574, Jinan, China, October, 2020.

[27] Z. H. O. N. G. Ying-Shou, L. I. Nan-Fang, Y. A. N. G. Li-Li, and Xu Wang, "Locating the Source of Message Diffusion in the Anonymous Network," *DEStech Transactions on Computer Science and Engineering*, 2017.

[28] J. Mark, M. Alberto, J. Gian, and V. Spyros, "The Peersim Simulator," 2003, http://peersim.sf.net.

[29] J. Tracey, "Building a Better Tor Experimentation Platform from the Magic of Dynamic elfs,", Master's Thesis, University of Waterloo , 2017.

[30] R. Jansen and N. Hooper, "Shadow: Running Tor in a Box for Accurate and Efficient Experimentation," NDSS, 2011.

[31] D. Komosny, S. Mrdovic, P. Ilko, M. Grejtak, and O. Pospichal, "Testing internet applications and services using planetlab," *Computer Standards & Interfaces*, vol. 53, pp. 33–38, 2017.

[32] B. Chun, D. Culler, T. Roscoe et al., "PlanetLab," *ACM SIGCOMM - Computer Communication Review*, vol. 33, no. 3, pp. 3–12, 2003.

[33] K. S. Bauer, M. Sherr, and D. Grunwald, *Experimentor: A Testbed for Safe and Realistic Tor Experimentation*Georgetown University, Washington D.C. USA, 2011.

[34] K. Venkatesh Vishwanath, D. Gupta, V. Amin, and K. Yocum, "Modelnet: towards a datacenter emulation environment," in *Proceedings of the 2009 IEEE Ninth International Conference on Peer-To-Peer Computing*, pp. 81-82, IEEE, Seattle, WA, USA, September, 2009.

[35] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and S. Paul, "Users get routed: traffic correlation on tor by realistic adversaries," in *Proceedings of the 2013 ACM SIGSAC Conference on*

Computer & communications security, pp. 337–348, Berlin, Germany, May, 2013.

[36] A. Panchenko and J. Renner, "Path selection metrics for performance-improved onion routing," in *Proceedings of the 2009 Ninth Annual International Symposium on Applications and the Internet*, pp. 114–120, IEEE, Bellevue, WA, USA, July, 2009.

[37] M. AlSabah and I. Goldberg, "Performance and security improvements for tor," *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–36, 2016.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 6218094, 12 pages
https://doi.org/10.1155/2022/6218094

Research Article

E-LPDAE: An Edge-Assisted Lightweight Power Data Aggregation and Encryption Scheme

Junhua Wu,[1] **Zhuqing Xu**,[1] **Guangshun Li**,[1] **Cang Fan**,[1] **Zhenyu Jin**,[1] **and Yuanwang Zheng**[2]

[1]*School of Computer Science, Qufu Normal University, Rizhao 276826, China*
[2]*Shandong Huatong Used Car Information Technology Limited Company, Jining 272000, China*

Correspondence should be addressed to Guangshun Li; guangshunli@qfnu.edu.cn

Received 6 January 2022; Revised 1 March 2022; Accepted 15 March 2022; Published 18 April 2022

Academic Editor: Jinbo Xiong

In smart grid systems, electric utilities require real-time access to customer electricity data; however, these data might reveal users' private information, presenting opportunities for edge computing to encrypt the information while also posing new challenges. In this paper, we propose an Edge-assisted Lightweight Power Data Aggregation Encryption (E-LPDAE) scheme for secure communication in a smart grid. First, in the edge privacy aggregation model, the data of smart meters are rationally divided and stored in a distributed manner using simulated annealing region division, and the edge servers of trusted organizations perform key one-time settings. The model encrypts the data using Paillier homomorphic encryption. It then runs a virtual name-based verification algorithm to achieve identity anonymization and verifiability of the encrypted data. The experimental results indicate that the E-LPDAE scheme reduces overall system power consumption and has significantly lower computation and communication overhead than existing aggregation schemes.

1. Introduction

In recent years, with the rapid development of modern science and technology and urbanization, the combination of power systems and information technology has produced a new concept—Smart Grid [1]. Smart Grid is the intelligence of the power grid. Building a smart grid can optimize resource allocation, reduce consumption, and increase efficiency. In smart grid applications, smart meters are deployed in all households in a residential area, each smart meter can collect the user's electricity consumption data and report it to the control center periodically (for example, every 15 minutes), and the control center can perform actions based on the reported data and real-time data analysis and take corresponding measures to ensure the health of the power system. Therefore, in the process of data transmission, a large number of real-time electricity consumption data of users is interacted with and calculated on the transmission line [2].

By using container technology, edge computing [3] is able to collect heterogeneous data in real time across a wide range of devices and can provide elastic computing resources for deep learning models. The resource configuration of edge computing can satisfy offline processing and analysis of small-area data, thereby ensuring the safe transmission and processing of various data. In addition, edge computing can reduce network latency and improve the utilization of network transmission bandwidth with the help of high-speed communication technology. In the implementation process of smart grid, the introduction of edge computing has a good development prospect, as shown in Figure 1.

Interaction and calculation of real-time electricity consumption provide a great convenience for power companies to fully grasp the electricity consumption of their customers but, at the same time, pose serious security and privacy risks. As pointed out by the National Institute of Standards and Technology (NIST) in the United States, there are more and richer data in smart grid systems. While bringing convenience to services, data leakage will also bring many security threats. Once the real-time electricity consumption information is stolen by the attacker, through the

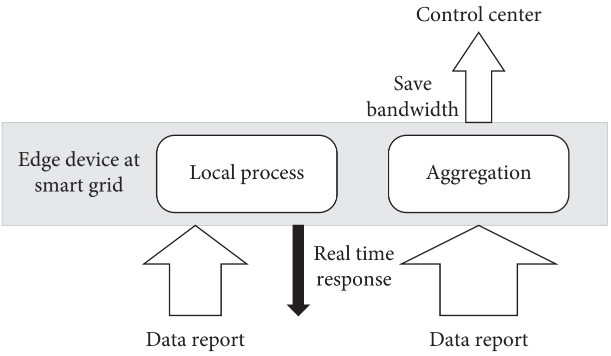


Figure 1: The edge computing paradigm extends cloud computing capabilities to the edge of the network to provide real-time response to local processes, as well as aggregated bandwidth savings.

analysis of the data, the user's detailed family life habits and other information can be obtained. Therefore, how to protect user privacy and data security in smart grids has become a research hotspot in recent years [4].

In order to overcome the above challenges, we propose an edge-assisted lightweight power data aggregation and encryption scheme. The main contributions of this paper are summarized as follows:

(i) An edge privacy aggregation model is proposed. The model uses simulated annealing (SA) to propose a segmentation algorithm for smart meters, Simulated Annealing Region Division (SARD). The algorithm can generate optimal area division according to the energy consumption of electricity meters, which is convenient for data collection and analysis of cluster electricity meters. The realization of distributed data storage is conducive to the privacy protection of smart meter data.

(ii) The Trusted Organization (TO) can set all keys in the system at one time, improve the efficiency of the smart grid, and reduce the power consumption of the system. Since a trusted organization stores a large amount of sensitive information such as keys, if it is stolen by an attacker, it will seriously threaten the data privacy and security of users. Such issues can be resolved by using edge servers, which are relatively trustworthy.

(iii) A virtual name-based authentication algorithm is proposed. The algorithm uses an encryption mechanism combining chameleon signature and Paillier cryptography to encrypt and verify the data to ensure the security of transmitted data while reducing the communication overhead; a selection strategy is developed using an attribute decision tree to improve the value of the data. Finally, the aggregated encrypted data is sent to the Cloud Power Distribution Center (CPDC). The CPDC decrypts the data in order to obtain the final result.

The rest of this paper is organized as follows. Section 2 summarizes the related work. In Section 3, we do some preparatory work. In Section 4, we describe the procedure and algorithm of the scheme. The results of the experimental analysis are reported in Section 5. Finally, the conclusions are discussed in Section 6.

2. Related Works

Although many secure communication schemes to protect the privacy of smart grid users have been introduced over the years, not many privacy-preserving aggregation schemes such as [5–8] have been proposed so far. Electricity consumption data collection is an important process in smart grid communication systems. However, a report from the Netherlands argues that frequent reading of smart meters is problematic from a legal point of view [9], violates the European Convention on Human Rights, and generates many load issues. Fortunately, integrating edge computing into smart grids and designing data aggregation schemes that protect privacy can avoid these problems. First, Pacific Northwest National Laboratory first proposed "edge computing" in an internal report in 2013. With the rapid growth of the Internet of Things, edge computing has received a lot of attention. Shi et al. summarize typical examples of the smart home and collaborative edge and present some of the challenges and opportunities in the area of edge computing [10]. It moves some of the workloads used in the cloud to the edge nodes. The security of sensitive data stored on cloud servers through edge nodes will be of great concern to users. Therefore, consideration should be given to the resource requirements of edge devices, as well as the privacy of smart grid users.

To address these issues, we use data aggregation technology to solve the transmission conflict problem of a large number of data packets for smart grids in edge computing. To improve the security of the data aggregation model, traditional secure data aggregation schemes use hop-by-hop aggregation encryption [11]. However, frequent encryption and decryption operations may affect the aggregation efficiency and increase the corresponding additional energy consumption and the delay of the data aggregation process. An efficient privacy-preserving aggregation scheme (EPPA) for smart grid communication [7] was proposed by Lu et al. They used a super-incremental sequence to construct multidimensional data and encrypted the data with Paillier homomorphic encryption [12]; however, the scheme has security flaws. Shi et al. used an untrusted aggregator to differentially aggregate multiple time slots, which is more costly based on computationally intensive systems [13]. Fan et al. [14] proposed a secure power usage data aggregation scheme for smart grids, but it critically requires a third-party trust mechanism for distribution, adding an additional burden. Li et al. proposed a distributed incremental data aggregation approach where they used homomorphic encryption to solve the repetitive regular data aggregation task [5]. Garcia and Jacobs used homomorphic encryption to ensure the privacy of users and gave a measurement method [6]. Lu et al. proposed a lightweight privacy-preserving data aggregation scheme called Lightweight Privacy-Preserving Data Aggregation (LPDA), but it cannot achieve identity anonymization [15]. Hua et al. proposed an effective smart grid aggregation scheme against malicious data mining attacks but increased the computational overhead and communication overhead [16].

3. Preparation

This section reviews the main basic concepts related to our work, including Paillier homomorphic encryption, simulated annealing region partition [17], chameleon hash function [18], and Attribute decision tree.

3.1. Paillier Homomorphic Encryption. Paillier cryptography is an additive homomorphic public key cryptography, which has been widely used in the field of encrypted signal processing or third-party data processing. Its homomorphic property is that the corresponding arithmetic operation can be performed on the ciphertext directly after encryption, and the result of the operation is the same as that of the corresponding operation in the plaintext domain. Its probabilistic property is that for the same plaintext, different ciphertexts can be obtained by different encryption processes, thus ensuring the semantic security of the ciphertext. The mechanisms used for encryption and decryption are as follows:

(1) Key generation: randomly select two large prime numbers p and q , calculate their product N and the least common multiple of $p-1$ and $q-1$, and then randomly select an integer that satisfies the following conditions:

$$\gcd\left(L\left(g^{\lambda} \bmod N^2\right), N\right) = 1. \tag{1}$$

Among them, function $L(u) = (u-1)/N$ and function $\gcd(.)$ are used to calculate the greatest common divisor of two numbers. Z_{N^2} is the set of integers less than $x \in Z_p^*$, while $Z_{N^2}^*$ is the set of integers coprime with N^2 in $Z_{N^3}^*$. (N, g) and λ are public key and private key, respectively.

(2) Encryption process: a random integer $r \in Z_i$ is selected. For any plaintext $m \in Z_w$, the corresponding ciphertext c is obtained by using public key (N, g) encryption:

$$\begin{aligned} c &= E[m, r] \\ &= g^m r^N \bmod N^2. \end{aligned} \tag{2}$$

According to the properties of the Paillier encryption system, when ciphertext $c \in Z_{N^2}^*$ is encrypted with the same public key, because the selection of ciphertext r is random, different ciphertext c can be obtained for the same plaintext m , but the same plaintext m can be restored after decryption, thus ensuring the semantic security of ciphertext.

(3) Decryption process: decrypt ciphertext c with private key n to get the corresponding plaintext m.

$$\begin{aligned} m &= D[c] \\ &= \frac{L\left(c^{\lambda} \bmod N^2\right)}{L\left(g^{\lambda} \bmod N^2\right)} \bmod N. \end{aligned} \tag{3}$$

3.2. Simulated Annealing Region Division

3.2.1. Regional Division. For a given smart meter, the division of area Q is expressed as follows:

$$Q \equiv \sum_{s=1}^{s_Q} \left[\frac{l_s}{L} - \left(\frac{d_s}{2L} \right)^2 \right]. \tag{4}$$

s_Q is the number of regions, L is the number of links between smart meter nodes in the smart grid, l_s is the number of regions in region Q, L is the number of links between smart meter nodes in the smart grid, l_s is the number of links between smart meter nodes in region Q, and d_s is the sum of degrees of smart meter nodes in region Q. First, we use equation (4) to randomly place smart meters on the device layer into the area. Finally, we use a simulated annealing algorithm to find the optimal partition.

3.2.2. Simulated Annealing Algorithm. It is a general probabilistic algorithm that is used in our scheme to find the optimal solution to the zoning problem, where one can find low-cost smart meter regions, but not local minima for high-cost smart meter regions. We introduce the energy consumption T_e of smart meters to achieve this. Starting from high T_e, it gradually decreases and the system gradually approaches the minimum cost, avoiding the high-cost local minima.

The purpose of identifying modules is to maximize the use of modules, where costs $C = -Q$ and Q are the areas defined in equation (4). We update each energy consumption randomly, and the probability is expressed as

$$p = \begin{cases} 1 C(S') \leq C(T_e), \\ \exp\left(-\dfrac{C(S') - C(T_e)}{T}\right) C(S') > C(T_e), \end{cases} \tag{5}$$

where $C(S')$ is the cost after the update and $C(T_e)$ is the cost before the update, $\Delta C = C(S') - C(T_e)$.

3.3. Chameleon Hash Function. Traditional cryptographic hash functions are difficult to find collisions. But the chameleon hash function can artificially set up a “back door”: if you master it, you can easily find collisions. This breaks the collision resistance of the hash function, but for most people, these properties remain, and the hash is still secure. Accenture applied the characteristics of the chameleon hash function and applied for a patent on an editable blockchain.

Although the decentralization and irrevocability of the blockchain are damaged to a certain extent, on the other hand, it also expands the application scenario of the blockchain and meets part of the needs of the government's regulatory requirements [19].

Principle description: suppose there exist two prime numbers p, q , and $q = kp + 1$ is large enough. The private key of the chameleon hash function is $x \in Z_p^*$, Z_p^* is the group of order q, and g is its generating element. The public key is $h = g^x \bmod p$. Given an arbitrary message m with random value $r \in Z_p^*$, now tampering the content m to m', it is now desired to find a random number r' such that $H(m\prime) = H(m)$. By the exponential property $g^a * g^b = g^{(a+b)}$, $(g^a)^b = g^{(ab)}$. The solution procedure for $r\prime$ is as follows:

$$H(m) = g^m h^r \bmod p = g^m g^{xr} \bmod p = g^{(m+xr)} \bmod p,$$
$$H(m\prime) = g^{m\prime} h^{r\prime} \bmod p = g^{m\prime} g^{xr\prime} \bmod p = g^{(m\prime+xr\prime)} \bmod p. \tag{6}$$

Therefore, m, $m\prime$, x, and r are known, $r\prime = (m + xr - m\prime)/x \bmod p$.

3.4. Attribute Decision Tree. The attribute decision tree is modeled after the access control tree and is set up according to the needs of the data collector. The leaf nodes of the attribute decision tree represent various attributes, and the intermediate nodes and roots are replaced by AND and OR. When an attribute of the data satisfies the requirements of the attribute decision tree, it is passed and the next calculation is performed; if not, other calculations or steps are performed.

For example, Mr. Li is a professor in the school of computer science of a university, so his attribute set matches the attribute strategy, as shown in Figure 2. Miss Wang is a professor in the school of information security of a university. Her attribute set does not match the attribute policy, as shown in Figure 3.

4. Edge-Assisted Lightweight Power Data Aggregation Encryption Scheme

4.1. Edge Privacy Aggregation Model. The edge privacy aggregation model contains four subjects: the User's Smart Meter (USM), the Marginal Power Services Institutions (MPSI), the Cloud Power Distribution Center, and the trusted organization. First, the USM encrypts data and divides it into optimal regions according to the change of energy consumption at different moments using a simulated annealing region partitioning algorithm, and as the energy consumption of USM changes at different moments, the number and location of clustered meters also change, thus realizing distributed data storage, which is conducive to the privacy protection of user data. Secondly, MPSI aggregates data with user identity anonymized and without affecting the privacy of any party. Finally, CPDC performs secure decryption, and TO performs key generation and distributes the key to the system. The model is shown in Figure 4.

User's Smart Meter. Smart meters use TPM chips to securely store and encrypt data. The SARD algorithm is executed using the handheld unit (including the sensor). Divide the smart meters of all users to meet the power load balance of the meters. The cluster meter regularly sends the collected data to the edge server. Perform data encryption calculation and chameleon signature calculation.

Marginal Power Services Institutions(MPSI). It consists of edge servers. The edge server performs chameleon signature aggregation and verification calculations and data aggregation calculations.

Cloud Power Distribution Center. The cloud server receives the aggregated data and decrypts it.

Trusted Organizations. The real identities of all users are virtualized to form virtual names and distribute system parameters and all private keys, and the distribution channels are all secure channels. The three parties of cloud, edge, and smart meter collaborate with trusted organizations to generate all private keys, as shown in Figure 5. Compared with existing solutions, our private keys require only a one-time setup between the three parties, which is beneficial for resource-limited systems. In addition, the private keys owned by TO are involved in decrypting the ciphertext and verifying the ciphertext, confusing the attacker, and making it impossible to tamper with the ciphertext.

4.2. Scheme Construction. The scheme proposed in this paper realizes the security and integrity of real-time power consumption data transmission between the smart meter and power server. The steps are as follows.

4.2.1. Initialization. TO inputs safety parameter (1^λ) and gets related parameter $(q_{1,} G_1, G_2, G_r, g_1, g_2, \omega, e)$, where q_1 is a large prime, G_1 and G_2 are two additive cyclic groups, G_r is a multiplicative cyclic group, q_1 is the order of the cyclic group, g_1 and g_2 are the generators of groups G_1 and G_2, respectively, satisfying that $\omega(g_2) = g_1$ and ω is an isomorphic mapping, e: $g_1 \times g_2 \longrightarrow g_r$ is bilinear mapping, and the storage list is established. TO chooses a system master key $s \in Z_p^*$, Z_p^* is a multiplication cycle group, and $y = g_2^s$ is the system public key. Two hash functions $H_1(.)$: $\{0,1\}^* \longrightarrow G_1$ and $H_2(.)$: $\{0,1\}^* \longrightarrow G_2$.

TO publishes system parameters and functions, selects a security parameter for the Paillier encryption algorithm, and sends it to the smart meter for initialization of the Paillier encryption algorithm. TO generates other parameters of the Paillier encryption algorithm: select two large prime numbers p and q, where $|p| = |q| = k$. The smart table computes $n = pq$ and chooses $g \in Z_{n^2}^*$ as the generator to use (n, g) as the public key of the Paillier encryption algorithm. CPDC computes the private key of the Paillier encryption algorithm $\lambda = lcm(p-1, q-1)$.

For the initialization of the chameleon signature, TO selects an element g_3 of order q in Z_p^* and an arbitrary index

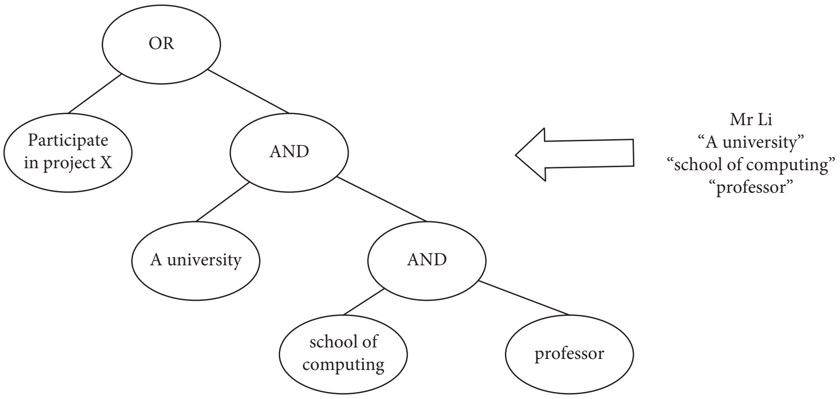


Figure 2: Schematic diagram of successful matching of policy and attribute collection.

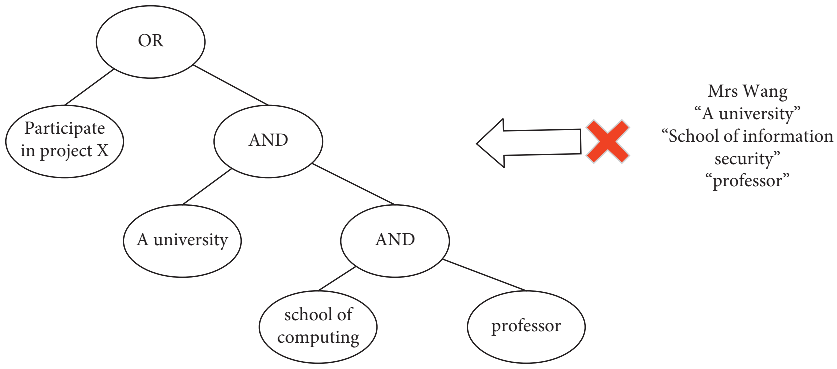


Figure 3: Schematic diagram of policy and attribute collection mismatch.

x , then the private key of the chameleon signature is $CK = x$, and the public key is $HK = g_3^x$.

TO sets the regularized attribute set F as a multiplicative cyclic group; then, any attribute f in the attribute set F is any element in the multiplicative cyclic group. The attribute set F is sent to the smart meter. Similarly, if TO sets the attribute set A of the attribute decision tree as a multiplicative cyclic group, then any attribute a in the attribute set A is any element in the multiplicative cyclic group, and the set attribute set A is sent to CPDC.

4.2.2. User Registration. Assuming a secure channel between TO and the user, in order to complete the user registration, the operation steps between the user and TO are as follows:

User i sends ID, serial number of smart meter to TO.

TO sends a Cert to user i after confirmation.

User i uses the Cert to get permission to request the parameters and key of the algorithm from TO.

TO sends the signature key etc. to the smart meter of user i.

TO calculation:

$$\begin{aligned} DP\ SI\ D &= H(I\ D, t)^{\text{Cert}}, \\ pid_{i,0} &= H(DP\ SI\ D, 0), \\ pid_{i,1} &= H(DP\ SI\ D, 1). \end{aligned} \tag{7}$$

TO calculates the signature key of user i:

$$\begin{aligned} S_{i,0} &= pid_{i,0}^s, \\ S_{i,1} &= pid_{i,1}^s. \end{aligned} \tag{8}$$

TO sends the signature key $S_i = (S_{i,0} S_{i,1})$, the real-time virtual name $DP\ SI\ D$ to the smart meter of user i.

4.2.3. Data Processing. Within data acquisition time t, the smart meter of user i encrypts the data with the Paillier homomorphic encryption and signs the encrypted data with the chameleon hash function which is referred to as chameleon signature for short. The cluster meter j collects data within the divided area. Finally, the real-time encrypted data and signatures are sent to MPSI. The steps are as follows.

The smart meter of user i selects a random number $a \in Z_{n^2}^*$ and encrypts data m_i.

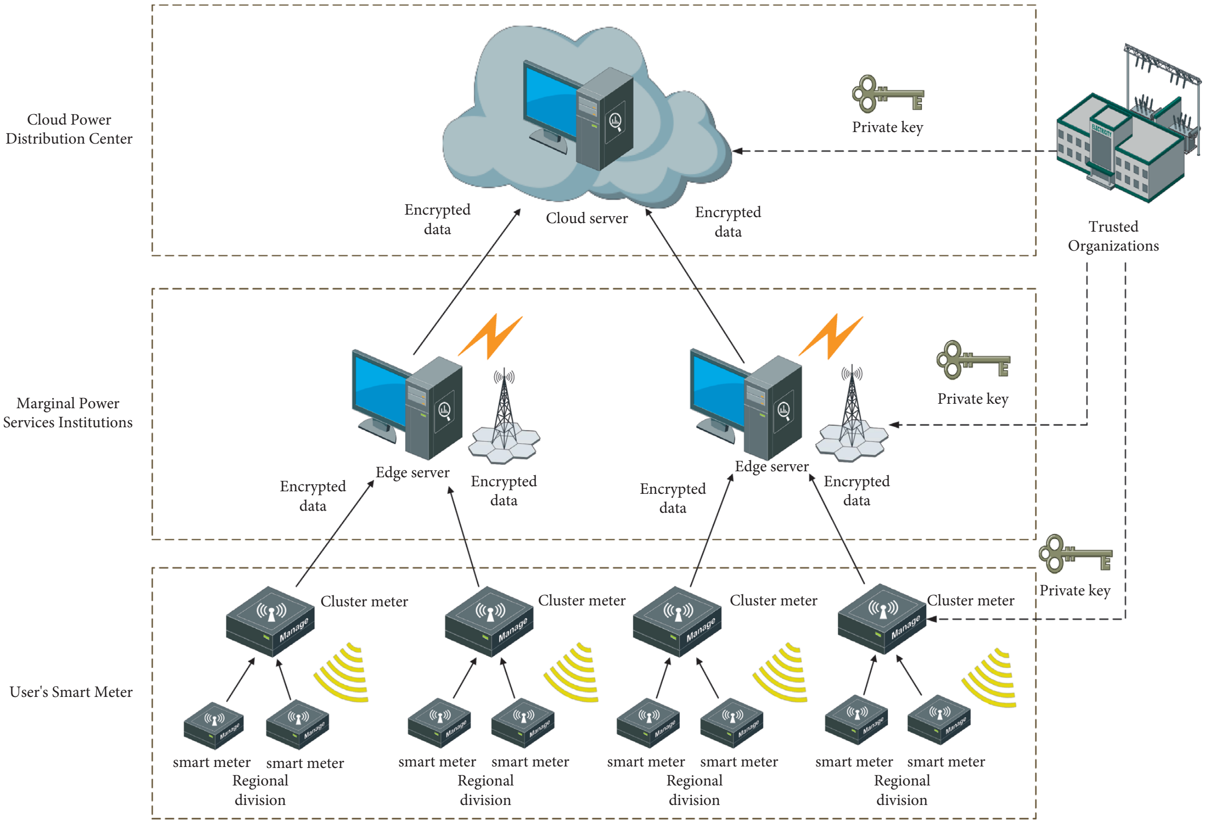


FIGURE 4: Edge privacy aggregation encryption model.

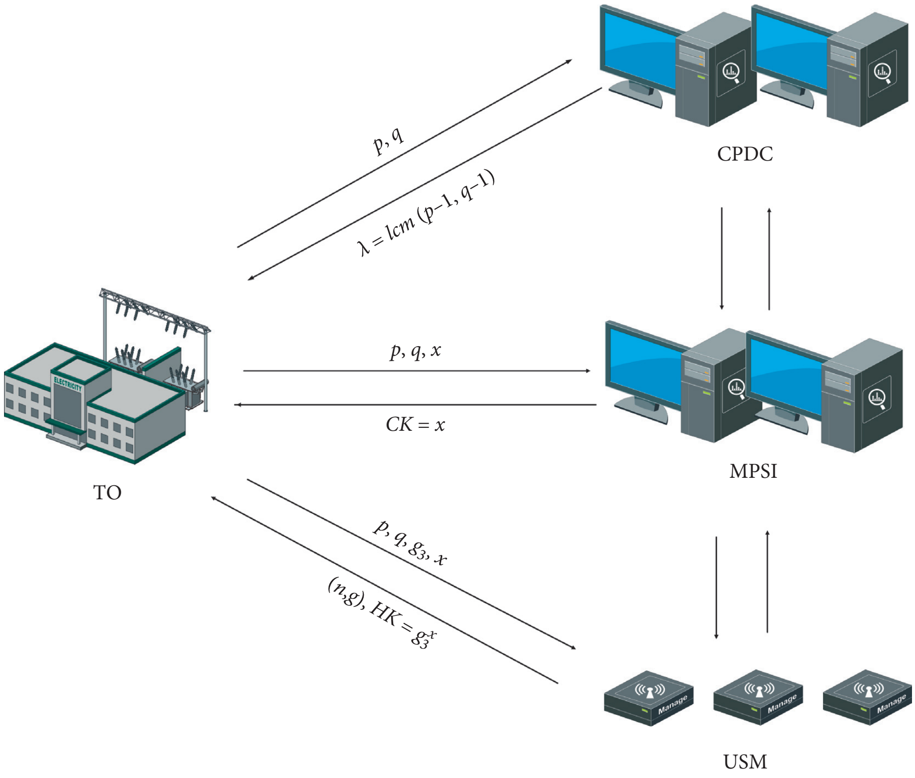


FIGURE 5: Key generation.

$$c_i = E(m_i) = g^{m_i} a^n \bmod n^2. \tag{9}$$

The smart meter of user i uses signature key $S_i = (S_{i,0} S_{i,1})$, virtual name, and attribute set to sign encrypted data by the chameleon hash function and finally send it to the cluster meter j.

$$\begin{aligned} h_i &= \text{Chamelelon}.H(c_i, HK, DP\ SI\ D, f), \\ \sigma_i &= s_{i,0} s_{i,1}^{h_i}. \end{aligned} \tag{10}$$

Cluster meter j sends $(c_i, \sigma_i, DP\ SI\ D)$ to MPSI.

MPSI receives the information and runs the virtual name-based verification algorithm as shown in Algorithm 1.

The algorithm first aggregates chameleon signatures. After verification, the attribute set f of the data is obtained, and the attribute set A of the data decision tree is matched in turn, and the data satisfying the data decision tree can be data aggregated with other data satisfying that decision tree for the data aggregation operation.

$$\begin{aligned} c &= \prod_{i=1}^{n} c_i \bmod n^2 \\ &= \prod_{i=1}^{n} g^{m_1} \ldots g^{mw} a^n \bmod n^2 \\ &= \prod_{i=1}^{n} g^{m_1+m_2+\ldots+m_w} a^n \bmod n^2 \end{aligned}. \tag{11}$$

After aggregation, MPSI sends the aggregated data to the CPDC through the secure channel. Data decryption: CPDC decrypts the encrypted aggregate data.

$$m_i = \frac{L\left(c^{\lambda} \bmod n^2\right)}{L\left(g^{\lambda} \bmod n^2\right)} \bmod n. \tag{12}$$

$m_i = m_1 + m_2 + \ldots\ldots + m_w$, CPDC stores data for power grid operation and puts forward decisions.

4.2.4. Track. While making power consumption analysis and decision-making, CPDC may find that some power consumption values do not meet its predetermined range or abnormal conditions. At this time, CPDC will start the tracking process, and the steps are as follows:

CPDC sends the command to the edge server that submits the relevant abnormal power consumption: let MPSI send the stored power consumption and virtual name at that time to CPDC.

CPDC first decrypts each encrypted data received, detects and finds the abnormal power consumption, and locks its $DP\ SI\ D$.

CPDC sends the virtual name of the abnormal power consumption determined by it to TO and applies for identity tracking.

TO can query the real identity of the users who send out abnormal electricity consumption. TO sends the real identity to CPDC, and CPDC processes the user and his power consumption accordingly.

4.3. Safety Analysis

4.3.1. User Identity Privacy Protection. Before sending data to the CPDC, the USM registers with the TO to obtain a virtual name and signing key. The USM uses the virtual name as the identity of the data transfer in the architecture and performs encryption, signing, and other actions based on it. The USM has a tamper-proof storage device. This storage device can be thought of as a “black box” that can read and write data, but only by the USM; no other device can read or write information. According to the one-way and collision-free characteristics of the hash function, even if the attacker obtains the virtual name, it cannot crack the real identity. This scheme can effectively protect user identity and prevent illegal intrusion.

4.3.2. Security Analysis of Chameleon Signature. The chameleon signature is a preferable designated verifier signature. Compared to other signatures, the chameleon signatures are more suitable for lightweight aggregated encryption schemes due to their ability to transmit data efficiently and reduce computational overhead. Chameleon signatures are also nontransmissible, nonforgeable, and nonrepudiation, which also ensure data security and meet the security requirements of the system.

4.3.3. User Fine-Grained Data Privacy Protection. USM encrypts the electricity consumption data using the Paillier encryption algorithm, sends it to MPSI, which does not have the ciphertext decryption key, and sends the ciphertext to CPDC after successful verification. CPDC mainly receives aggregated numbers of electricity consumption data, so it protects the user’s fine-grained data privacy, while CPDC can get the complete electricity consumption data.

5. Experimental Analysis

5.1. Simulated Annealing Region Division. Intraregional connectivity and participation: each region is divided into relatively balanced regions from one or several fully centralized regions based on the energy consumption of smart meters to achieve a balanced electrical load in each region. We define the intraregional connectivity, in order to measure whether the smart meter u is well connected to other smart meters in the region.

$$Z_u = \frac{k_u - \bar{k}_{s_u}}{\sigma_{k_{s_u}}}, \tag{13}$$

where k_u is the number of links from the smart meter u to other smart meters in zone s_u, $\overline{k}_{s_u}$ is the average number of links from all smart meters in the zone s_u, and $\sigma_{k_{s_u}}$ is the standard deviation of all links in the zone.

Of course, we also need to consider unexpected situations. For example, a smart meter u may not be connected to its own area. Therefore, we define the participation degree p_u of a smart meter u.

$$p_u = 1 - \sum_{s=1}^{s_M} \left(\frac{k_{us}}{k_u} \right)^2, \tag{14}$$

where k_{us} is the number of links from the smart meter u to smart meters in zone s , and k_u is the total number of degrees of the smart meter u. According to equation (14), if the connections of the smart meter u are evenly distributed in all areas, then the participation degree of the smart meter u is close to 1. If all its connections are in its own area, the participation degree is 0.

We use a MATLAB environment with a Dell laptop (i5-6200u, CPU 2.40 GHz, Windows 10 OS) for simulation experiments. Assuming that 100 smart meters are randomly distributed in a $1.0 * 1.0$ km smart grid, and each smart meter has a random electricity consumption $N(T_e)$, a zoning model is established. First, the 100 randomly distributed smart meters are generated as a subset of the neighborhood of electricity consumption $N(T_e)$ Download the open-source dataset from the website Open Energy Data Initiative (OEDI) and randomly select the electricity consumption information from 100 apartments with no missing points and a time granularity of 15 minutes. The average value is calculated based on the electricity load of 100 users at different times of the day, as shown in Figure 6. 14:00–20:00, the user's electricity load continues to grow, with 20:00 reaching the highest peak of the day; 20:00–24:00, the user's electricity load continues to fall to a stable value. After reasonable analysis, we divide the average value of the electricity load of 100 users in different time periods of a day into 6 electricity consumption states. A power consumption state of $S(k)$ is randomly selected for the regional division scheme, and the next power consumption state of S' is randomly selected as the candidate scheme for the next regional division scheme. Calculate $\Delta C = C(S') - C(T_e)$; if $\Delta C < 0$, accept S' for the next region division scheme; otherwise, we judge the random update probability $p = \exp(-\Delta C/cT) > \alpha$, $\alpha \in (0,1)$; if true, accept S' for the next region division scheme; namely, $S(k+1) = S'$, $k = k+1$. Then, we check whether the connectivity and participation in the region satisfy equations (13) and (14). Finally, we use $S(k+1)$ for the region partition scheme and return the SARD algorithm.

Figures 7(a)–7(f) show the experimental process of the SARD algorithm. We performed six rounds of state calculation, divided the six power consumption states into different regions, and terminated the algorithm. Cluster meters in each area are used to collect data and process the data accordingly to realize power load balancing under different power consumption states.

First, the power consumption of smart meters increases with the increase of users in the smart grid. Since all the data eventually needs to be sent to the cloud server of CPDC for processing, the power consumption of the cloud server also increases with the increase of data, as shown in Figure 8. Then, we introduce edge computing into the smart grid, and the power consumption of MPSI increases with the increase of edge servers. This layer processes a large amount of data and then sends it to the CPDC. Since the CPDC does not need to process a large amount of data, the power consumption of the cloud server in the CPDC does not fluctuate much, as shown in Figure 9. Comparing the experiments in the two figures, the introduction of edge servers to process large amounts of data in the edge privacy aggregation model of the smart grid effectively reduces the power consumption of the CPDC and the total system power consumption.

5.2. Total Computing Overhead. The computational overheads of this scheme and the LPDA scheme mainly involve the following three operations: bilinear pair operation, exponential operation, and Paillier homomorphic encryption and a decryption operation, and other operations are neglected. The bilinear pair operation and the exponential operation are C_b and C_e, respectively, and the encryption and decryption of the Paillier algorithm are C_A and C_B, respectively, and the other computational overheads are neglected. The AMDM scheme is mainly multiple operations, C_{pe} is the multiplication operation in the cyclic group $Z^*_{N^2}$, C_{pm} and C_m are the multiplication operation in $Z^*_{p'}$, C_e is the exponential operation, and C_{gm} is the multiplication operation in the group G_1 because C_{pm} and C_m produce little effect negligible.

The scheme uses the MATLAB environment of a Dell laptop (i5-6200u, CPU 2.40 GHz, Windows 10 OS) for simulation experiments. The simulation measures the amount of time needed by the Dell laptop to perform basic operations in the experimental environment. It takes 1.1 ms to calculate a single C_e, 3.1 ms to calculate C_b, 4.5 ms to calculate C_{pe}, and 2.1 ms to calculate C_{gm}. Since all three scenarios in this paper have only one pair of encryption and decryption operations, we first disregard C_A and C_B .

The scenarios in this paper consider the computational overhead of each of the three participants, Smart Meter, MPSI, and CPDC, and compare them with other scenarios, as shown in Table 1. The total computational overhead of all the solutions is the total computational overhead of the three participants. As can be seen from the table, this paper is significantly more efficient than the other two schemes.

As shown in Figure 10, the computing energy consumption of the scheme in this paper is significantly lower than the aggregated encryption schemes of the remaining two schemes, where the AMDM scheme resists malicious attacks and requires more computing energy and is significantly higher than the LPDA scheme and the scheme in this chapter, while the scheme in this chapter does not cause additional computation while ensuring data security due to the use of the chameleon signature, so the total computation overhead is lower, and it can be said that the scheme in this chapter is better than the LPDA scheme and the AMDM scheme.

5.3. Total Communication Overhead. The total communication overhead of this scheme mainly refers to all the communication data that needs to be transmitted in the system. The output data length of the hash function is

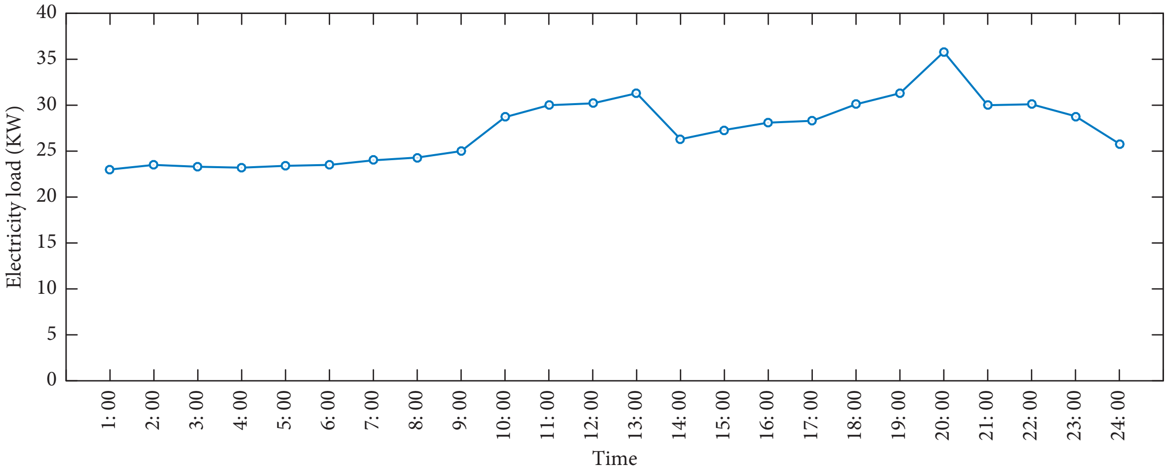
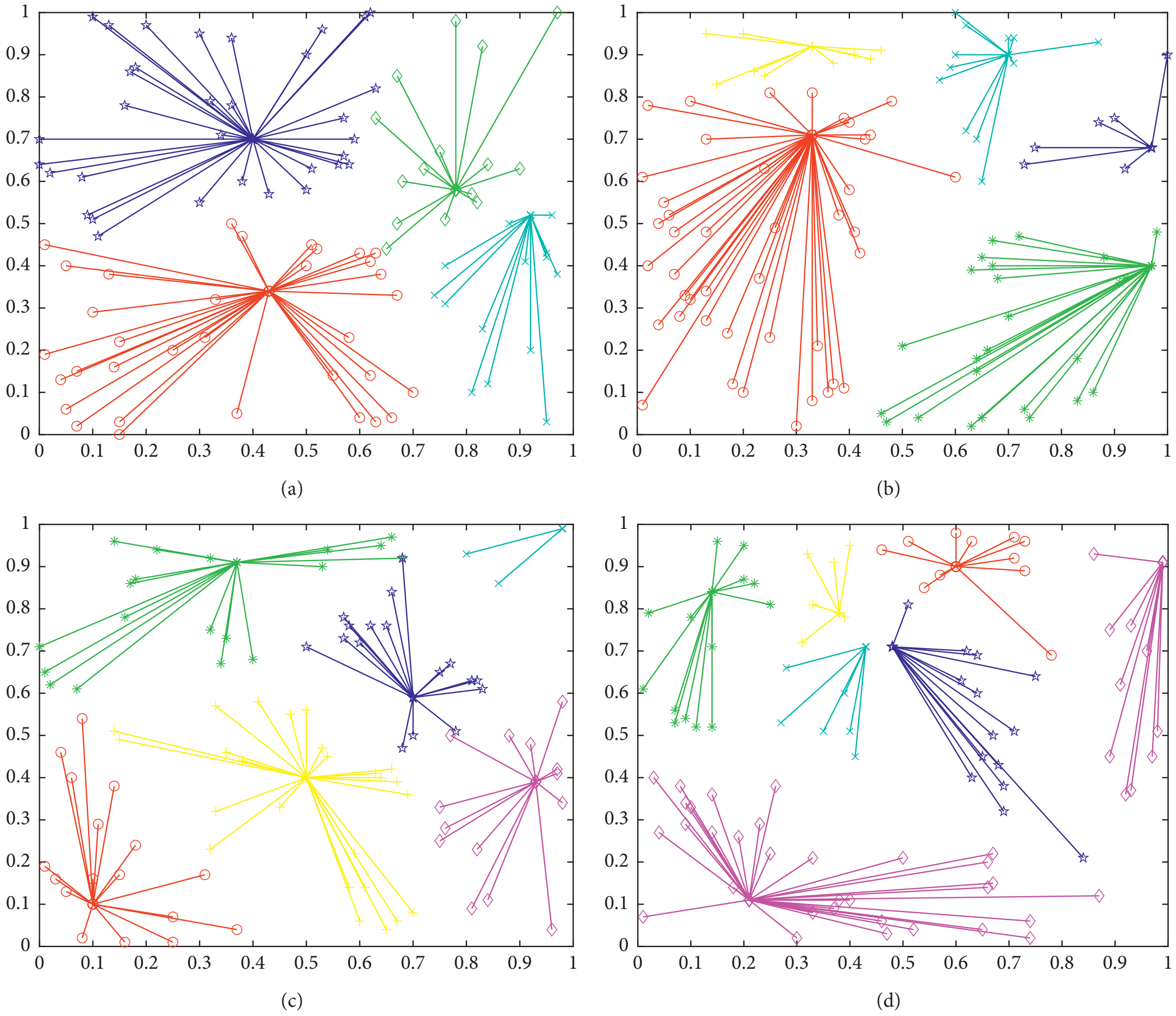


FIGURE 6: Edge privacy aggregation encryption model.



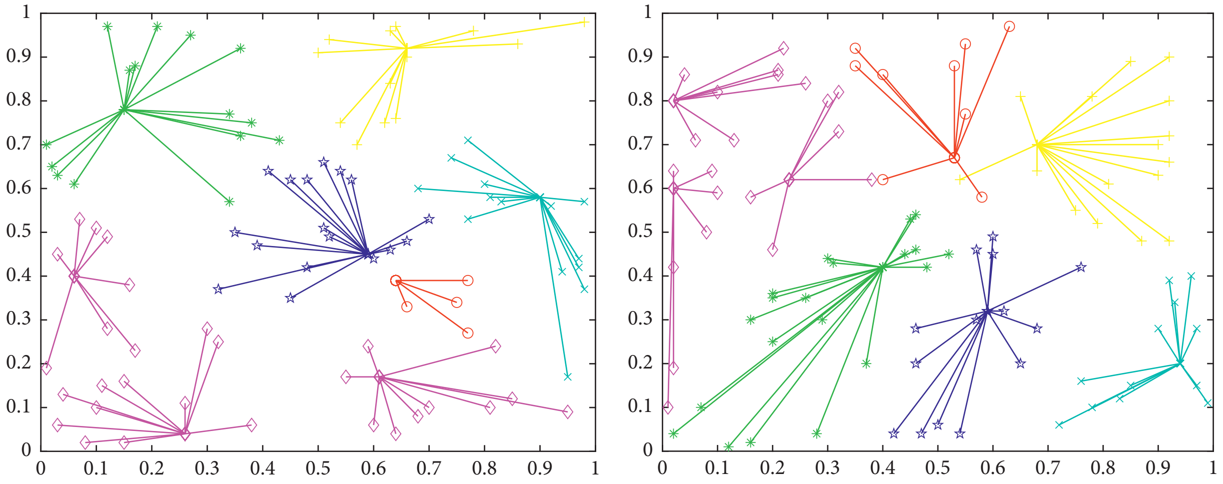
(a)

(b)

(c)

(d)

FIGURE 7: Continued.



(e)

(f)

Figure 7: Regional division based on electricity consumption state. (a) Partition of 0:00–7:00. (b) Partition of 13:00-14:00. (c) Partition of 7:00–11:00. (d) Partition of 11:00–13:00. (e) Partition of 14:00–20:00. (f) Partition of 20:00–24:00.

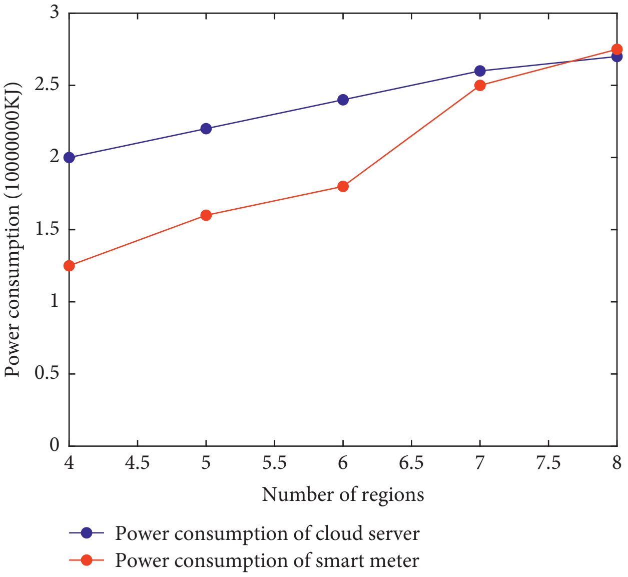


Figure 8: Energy consumption of cloud network.

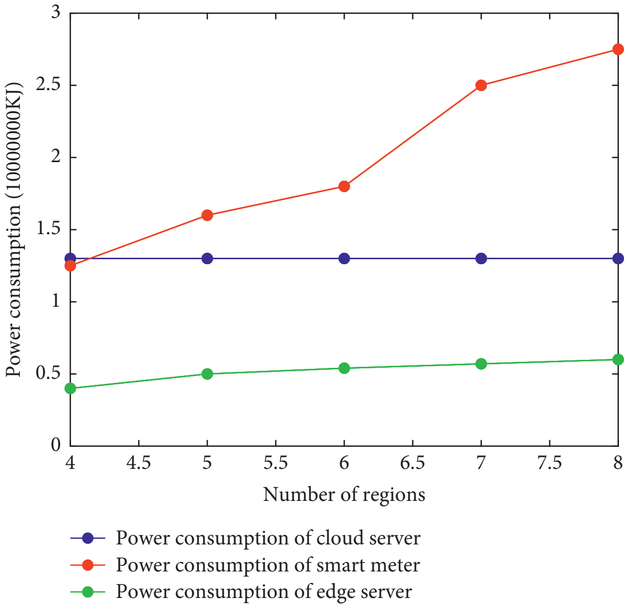


Figure 9: Energy consumption of edge computing network.

160 bits. Suppose the length of n in Paillier encryption algorithm is 512 bits, the length of group G_1 element is 161 bits, the length of $DP\ SI\ D$ is 32 bits, the length of attribute set f is 32 bits, and the length of σ_i is 32 bits. The total communication data volume of this scheme consists of two parts: the first part is from SM to MPSI, and the data transmitted is $(c_i, \sigma_i, DP\ SI\ D)$; the second part is from MPSI to CPDC, and the data transmitted is c . The total traffic of the LPDA scheme consists of two parts. The first part is from SM to ESP, which transmits 2048 bits through calculation, and the second part is from ESP to CC, which transmits 2048 bits through calculation. The total traffic of the AMDM scheme consists of two parts. The first part is SM to GW, which transmits 3264 bits through calculation, and the second part is GW to CC, which transmits 3264 bits through calculation. The comparison between this scheme and other schemes is shown in Table 2.The simulation experiment is carried out using MATLAB, and the results are shown in Figure 11.

We use the smart meter data of a year in London on the Open Energy Data Initiative (OEDI) website to simulate the total communication cost per day. As shown in Figure 12, different colors represent different communication situations; that is, when the number of edge servers and smart meters changes, the communication cost also changes. Based on the actual privacy requirements and cost requirements of the customer, we implement appropriate electricity usage data delivery mechanisms in the actual area.

```
Input: $c_i, \sigma_i$, $DP\ SI\ D$
Output: $c$
for $i = 1; i < n; i++$ do
    $\Omega = \prod_{i=1}^{n} \sigma_i$;
    $h_i$ = Chamelelon.$H(c_i', CK, DP\ SI\ D, f\prime), c_i', f\prime \in Z_p^*$;
    $f = f' - c_i - c_i'/x \bmod p$;
    $f$ match $A$;
    $pid_{i,0} = H(DP\ SI\ D, 0), pid_{i,1} = H(DP\ SI\ D, 1)$;
    $e(\Omega, g) = e(\prod_{i=1}^{n} pid_{i,0} pid_{i,1}^{h_i}, y)$;
    $c = \prod_{i=1}^{n} c_i \bmod n^2$;
end for
MPSI sends $c$ to CPDC;
```

Algorithm 1: Verification algorithm based on the virtual name.

Table 1: Analysis of computational complexity.

Scheme	SM	MPSI (ESP, DCP)	CPDC (CC)
Our scheme	$3C_e + C_A$	$NC_e + 2C_b$	C_B
LPDA	$C_e + C_A$	NC_e	$NC_e + C_B$
AMDM	$2C_{pe} + 2C_e + C_{gm} + C_A$	$(N+2)C_b + C_{gm}$	$2C_b + C_{pe} + 2C_e + C_B$

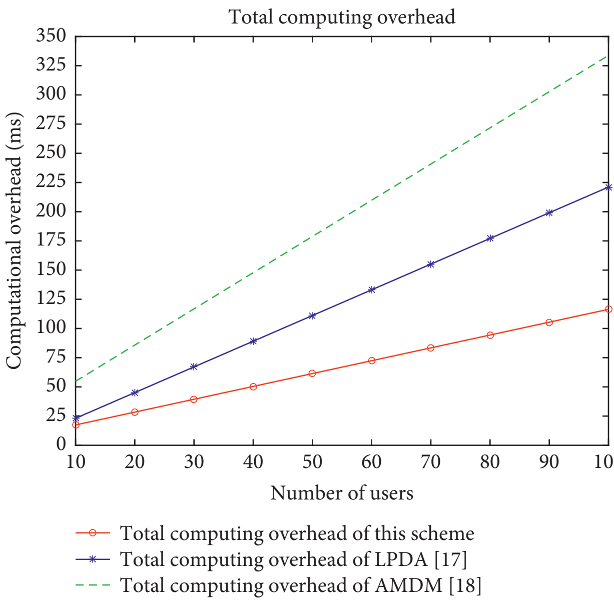


Figure 10: Total computing overhead.

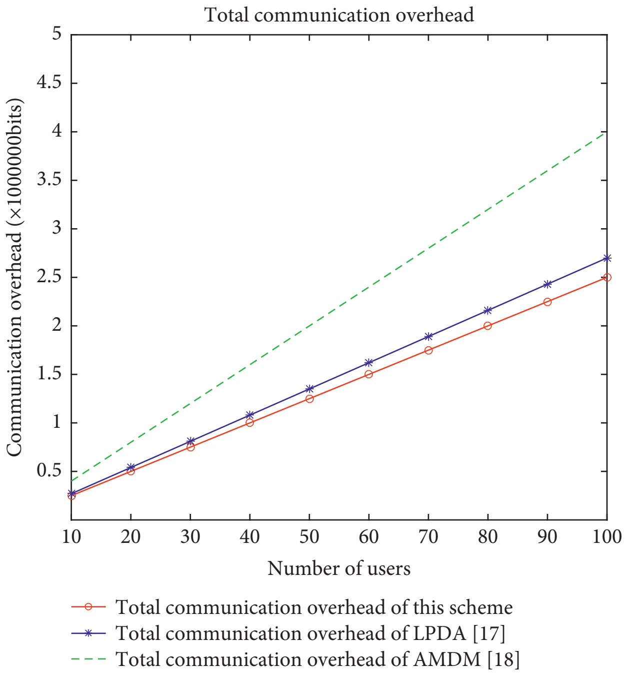


Figure 11: Total communication overhead.

Table 2: Analysis of communication complexity.

Scheme	SM (bit)	MPSI (ESP, DCP) (bit)
Our scheme	1409	1024
LPDA	2048	2048
AMDM	3264	3264

6. Conclusion

In this paper, we consider the actual smart grid, introduce edge computing, and propose an edge-assisted lightweight electricity consumption data aggregation and encryption

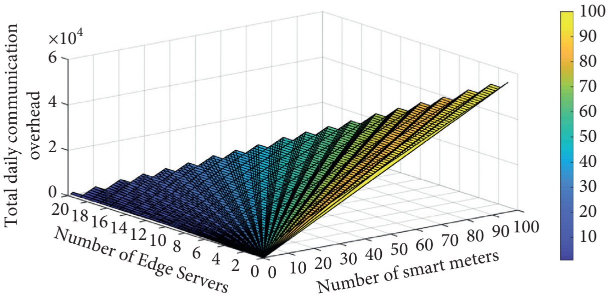


Figure 12: Total daily communication overhead.

scheme, which solves the problem of sending electricity consumption data to the cloud by users securely and efficiently. The scheme uses a simulated annealing zone partitioning algorithm to reasonably partition smart meters according to their electricity consumption energy consumption to achieve load balancing of smart grid systems; at each sending of data, licensed users apply for virtual names from trusted organizations to enable them to communicate with the grid as anonymous, which effectively protects the privacy of user identity security; in encrypting data, CPDC uses a virtual name-based verification algorithm which is used to Paillier encryption technology combined with chameleon signature to ensure authentication, integrity, and nonrepudiation of data, so that CPDC can only obtain encrypted data aggregated by MPSI, protecting the privacy of users' fine-grained data. Performance analysis shows that it is much better than the LPDA scheme and AMDM scheme in terms of communication overhead and computation overhead. In future work, we will evaluate our schemes in realistic smart grid scenarios with stronger adversarial models and study the impact of different signatures on system performance and security.

Data Availability

The research data are obtained from the corresponding author.

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 (Grant nos. 61771289 and 61832012), Major Basic Research of Natural Science Foundation of Shandong Province (Grant no. ZR2019ZD10), and Key Research and Development Program of Shandong Province (Grant no. 2019GGX101050).

References

[1] A. Saleem, A. Khan, S. U. R. Malik et al., "FESDA: fog-enabled secure data aggregation in smart grid IoT network," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6132–6142, 2020.

[2] K. Wei, S. Jian, and Pandi, "A practical group blind signature scheme for privacy protection in smart grid," *Journal of Parallel and Distributed Computing*, vol. 136, pp. 29–39, 2020.

[3] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.

[4] J. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2019.

[5] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *Proceedings of the First IEEE International Conference on Smart Grid Communications*, pp. 327–332, IEEE Press, Gaithersburg, MD, USA, October 2010.

[6] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *Proceedings of the 6th International Conference on Security and Trust Management*, vol. 67, no. 10, pp. 226–238, Springer-Verlag, Berlin, Germany, 2011.

[7] R. Rongxing Lu, X. Xiaohui Liang, X. Xu Li, X. Xiaodong Lin, and X. Xuemin Shen, "EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.

[8] R. Petrlic, "A privacy-preserving concept for smart grids," *Sicherheit in Vernetzten Systemen*, pp. B1–B14, 2010.

[9] Q. Zhou, G. Yang, and L. He, "An efficient secure data aggregation based on homomorphic primitives in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 10, no. 1, pp. 2022–2037, 2014.

[10] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[11] Y. Yang, X. Wang, S. Zhu, and G. Cao, "Sdap," *ACM Transactions on Information and System Security*, vol. 11, no. 4, pp. 1–43, 2008.

[12] P. Paillier, "A public-key cryptosystem based on composite degree residuosity classes," *Advances in Cryptology - EUROCRYPT'99*, vol. 1592, pp. 223–238, Springer-Verlag, Berlin, 1999.

[13] E. Shi, T. H. Chan, E. G. Rieffel, R. Chow, and D. Song, "Privacy preserving aggregation of time-series data," *Network and Distributed System Security (NDSS)*, vol. 2, no. 4, pp. 1–17, 2011.

[14] C.-I. Fan, S.-Y. Huang, and Y.-L. Lai, "Privacy-enhanced data aggregation scheme Against internal attackers in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 666–675, 2014.

[15] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

[16] Y. Dong, J. hen, S. Ji, Q. Rongxin, and L. Shuai, "A novel appliance-based secure data aggregation scheme for bill generation and demand management in smart grids," *Connection Science*, vol. 33, no. 4, pp. 1–22, 2021.

[17] L. Ren and L. Lin, "Simulated annealing algorithm coupled with a deterministic method for parameter extraction of energetic hysteresis model," *IEEE Transactions on Magnetics*, vol. 54, no. 11, pp. 1–5, 2018.

[18] S. A. Hua, A. Yl, X. D. Zhe, and Z. Mingwu, "An efficient aggregation scheme resisting on malicious data mining attacks for smart grid," *Information Sciences*, vol. 526, pp. 289–300, 2020.

[19] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, "A blockchain-based machine learning framework for edge services in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, 2022.