Wireless Communications and Mobile Computing

Secure Computational Solutions for Sparse Data Challenges in the Internet of Things

Lead Guest Editor: Yan Huang
Guest Editors: Donghyun Kim, Fei Hao, Yan Huo, and Madhuri Siddula

Secure Computational Solutions for Sparse Data Challenges in the Internet of Things

Wireless Communications and Mobile Computing

Secure Computational Solutions for Sparse Data Challenges in the Internet of Things

Lead Guest Editor: Yan Huang
Guest Editors: Donghyun Kim, Fei Hao, Yan Huo, and Madhuri Siddula

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5340116, 21 pages
https://doi.org/10.1155/2021/5340116

Research Article

BFR-SE: A Blockchain-Based Fair and Reliable Searchable Encryption Scheme for IoT with Fine-Grained Access Control in Cloud Environment

Hongmin Gao,[1] Shoushan Luo,[1] Zhaofeng Ma,[1] Xiaodan Yan,[2] and Yanping Xu[3]

[1] *Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *Beihang University, Beijing 100191, China*
[3] *School of Cyberspace Security, Hangzhou Dianzi University, Hangzhou, Zhejiang Province 310018, China*

Correspondence should be addressed to Xiaodan Yan; yanxiaodan@buaa.edu.cn

Received 29 July 2021; Revised 1 September 2021; Accepted 29 October 2021; Published 24 November 2021

Academic Editor: Yan Huang

Due to capacity limitations, large amounts of data generated by IoT devices are often stored on cloud servers. These data are usually encrypted to prevent the disclosure, which significantly affects the availability of this data. Searchable encryption (SE) allows a party to store his data created by his IoT devices or mobile in encryption on the cloud server to protect his privacy while retaining his ability to search for data. However, the general SE techniques are all pay-then-use. The searchable encryption service providers (SESP) are considered curious but honest, making it unfair and unreliable. To address these problems, we combined ciphertext-policy attribute-based encryption, Bloom filter, and blockchain to propose a blockchain-based fair and reliable searchable encryption scheme (BFR-SE) in this paper. In BFR-SE, we constructed an attribute-based searchable encryption model that can provide fine-grained access control. The data owner stores the indices on SESP and stores some additional auxiliary information on the blockchain. After a data user initiates a request, SESP must return the correct and integral search results before the deadline. Otherwise, the data user can send an arbitration request, and the blockchain will make a ruling. The blockchain will only perform arbitrations based on auxiliary information when disputes arise, saving the computing resources on-chain. We analyzed the security and privacy of BFR-SE and simulated our scheme on the EOS blockchain, which proves that BFR-SE is feasible. Meanwhile, we provided a thorough analysis of storage and computing overhead, proving that BFR-SE is practical and has good performance.

1. Introduction

With the continuous development of Mobile Internet, 5G, and some other advanced technologies, especially the Internet of Things, people and machines are always generating massive amounts of data. Most IoT devices produce large amounts of data with a limited storage capacity, so the data owners like to use cloud storage services to reduce the burden of maintenance costs and local storage overhead. Cloud services provide users with great convenience, enabling users to access their data anytime and anywhere, instead of using a specific machine. But these data, especially the data generated by specific IoT devices such as smart homes and intelligent wristbands, often contains sensitive information to the user. To prevent the disclosure, users encrypt their data before uploading it to the cloud server [1–8]. However, encryption will weaken the ability of users to search for data.

Searchable encryption technology was first proposed by Song et al. [9], which allows a party to store his data in encryption on the cloud server to protect his privacy while retaining his ability to search for data. A searchable encryption scheme typically includes three participants: the data owner (DO), the data user (DU), and the cloud server. The DO encrypts his data together with the corresponding keywords and uploads them to the cloud server. The cloud server maintains these ciphertexts and provides search services for data users. A data user will initiate a search request using a search token generated based on the keywords, and the matching search results will be sent to him by the cloud server. Finally, the data user can decrypt the ciphertext locally to obtain the data. The whole process will not expose any information related to the data itself. Nowadays, many researchers have proposed various searchable encryption algorithms, such as asymmetric searchable encryption [10, 11], multikeyword searchable encryption [12, 13], and fuzzy keyword searchable encryption [14, 15]. Most of the above studies focus on searchable encryption's privacy and performance in different scenarios and assume that the cloud server is curious but honest. However, this is not the case, which will cause problems in the fairness and reliability of searchable encryption:

(1) On the one hand, after the user pays, the cloud server cannot provide satisfactory search services, resulting in the user's economic losses. On the other hand, after the user obtains the desired search results, he will slander and deny the cloud server's service and deceptively refuse to pay the service fee

(2) The cloud server is not always reliable. To save costs, it may delete data that is not often used at ordinary times to save space. When users search, it will send part of the search results or even send fake data to users

According to the above point of view, in addition to the privacy of keywords and search algorithms' efficiency, practical searchable encryption is highly expected to be fair and reliable. To solve these problems, we urgently need such a searchable encryption scheme, in which the service provider is always to provide reliable search service, and the users pay for it. There is no credible third party in this scheme but will not cause any economic disputes. Fortunately, with the emergence and development of Bitcoin [16], as a decentralized cryptocurrency, its underlying technology blockchain can gracefully help us to achieve this goal. In this paper, we proposed a fair and reliable searchable encryption scheme (BFR-SE) based on blockchain. The main contributions of our research are as follows:

(1) We constructed an attribute-based searchable encryption algorithm (ABSE) and combined it with blockchain and Bloom filter to propose a fair and reliable searchable encryption scheme. While the DO stores the data indices in the SESP, some additional auxiliary information used for verification is uploaded to the blockchain. In the event of disputes between DU and SESP, the blockchain will arbitrate, and the dishonest participant will be punished financially

(2) BFR-SE supports users' multikeyword search for ciphertexts. By utilizing ABSE, the DO realizes fine-grained access control for their data search, which means that only the users whose attributes satisfy the specific policy can search and obtain the correct search results

(3) Not the same as other blockchain-based searchable encryption schemes, BFR-SE only stores a small amount of auxiliary information on-chain and performs possible arbitration when disputes occur, which dramatically saves storage and computing resources on-chain

(4) We simulated and implemented BFR-SE on the EOS blockchain and showed implementation details of smart contracts and algorithms. Together with the security analysis, it proves that our scheme is feasible

(5) We used 6 MacBook Pros to build an EOS private chain in a laboratory environment and simulated our scheme. The storage and computing overhead proves that BFR-SE is practical and has good performance

The rest of this paper is organized as follows: Section 2 consists of related works. Section 3 reviews some preliminary knowledge used throughout this paper. In section 4, we have an overview of our scheme. Section 5 describes specific implementation details. In Section 6, we analyze the security and performance. Finally, we present the conclusion and future direction.

2. Related Work

2.1. Verifiable Searchable Encryption. To ensure the reliability of searchable encryption and prevent the cloud server from returning partial or even wrong search results, users need to have the ability to verify the correctness of search results. Earliest in 2012, Chai and Gong [17] proposed a verifiable keyword search scheme, in which the cloud server needs to prove that the returned results are correct. Kurosawa and Ohtaki [18] proposed the first UC-secure verifiable symmetric searchable encryption, which can verify whether the search results are modified or deleted, and the computational cost of verification has a linear relationship with the number of files. Zhu et al. [19] constructed a verifiable fuzzy keyword search scheme to support dynamic data using Bloom filter and locality sensitive hash function. The single-keyword verifiable searchable encryption will return many irrelevant results that cause the waste of transmission bandwidth and computing resources, so the verifiable searchable encryption proposed by Azraoui et al. [20] supports multikeyword search or combined search. However, the above verifiable searchable encryption schemes are only suitable for a small number of users, and it is challenging to meet the user's dynamic requirements in the cloud environment. The number of users growing will cause the burden of key management and cannot achieve fine-grained access control. In 2014, Zheng et al. [21] proposed a novel cryptographic primitive named verifiable attribute-based keyword search. This primitive allows DO to control the search and outsource his encrypted data to the cloud server based on an access policy. Simultaneously, it allows legitimate users to outsource the search operation (usually expensive) to the cloud server and verify whether the cloud server loyalty performs it. Ameri et al. [22] combined hierarchical identity-based multidesignated verifier signature (HIB-MDVS), hierarchical identity-based broadcast encryption (HIBBE) and Bloom filter to propose a generic construction for verifiable attribute-based keyword search. The VBKS scheme proposed by Sun et al. [23] can realize the revocation of user attributes and utilize proxy reencryption and lazy reencryption to transfer the heavy update work during attribute revocation to a semitrusted cloud server while supporting the multikeyword search.

As mentioned above on verifiable searchable encryption, the research enables users to verify the search results' correctness and integrity. However, as a more practical searchable encryption scheme, this is far from enough because when a dishonest server is detected, it cannot continue to punish the dishonest server without a third-party trusted organization, which cannot be genuinely reliable.

2.2. Blockchain-Based Searchable Encryption. In recent years, some researchers utilized blockchain to solve the fairness problem in searchable encryption. In 2017, Li et al. [24] used blockchain to construct symmetric searchable encryption (SSE-using-BC). In their scheme, users publicly store all data on the Bitcoin through transactions. As long as the participant does not execute honestly, he will lose his BTC. In their subsequent work [25], they also improved the scheme and adopted the Fabric blockchain, which significantly improved the performance. Hu et al. [26] explored the potential capabilities of the Ethereum blockchain and constructed a decentralized, privacy-protected search model. The scheme designed a financially fair smart contract to replace the centralized server so that all participants are treated equally and motivated to perform correct operations. Cai et al. [27] also used a smart contract to record encrypted search records on the blockchain and designed a fair protocol to deal with disputes and payment issues. They used a dynamic, efficient searchable encryption scheme, which retained the search capability and inspired the service provider to make a real effort. Tang [28] extended the original searchable encryption, storing some necessary information on-chain, in which blockchain only serves as a proper judicial function. If there is no dispute, it will perform little operations on-chain, reducing the blockchain's burden. Chen et al. [29] stored the indices and complex logical structure of EHRs on the blockchain. They believed that only utilizing blockchain for propagation can the data owner have complete control over their data. Blockchain ensures the integrity, unforgeability, and traceability of the indices. Jiang et al. [30] proposed a Bloom filter-enabled multikeyword search protocol with enhanced efficiency and privacy preservation. In the protocol, a low-frequency keyword is selected using the Bloom filter to filter the database when performing a multikeyword search.

In summary, although the above searchable encryption based on the blockchain can solve the fairness problem in the payment process, there are still some shortcomings:

(1) These schemes are products of the combination of blockchain and symmetric searchable encryption that can only achieve a single one-to-one scenario that is difficult for a large number of users and meeting dynamic requirements in a cloud environment, not to mention fine-grained access control

(2) The main idea of these schemes is to store index information of the encrypted data on-chain. Although encrypted, the symmetric searchable encryption is generally a deterministic function. It will be noticed when the user searches for the same keyword multiple times. It will lead to the establishment of some statistics, making it possible to infer some private information

(3) Both file storage and search algorithm execution are all processed on-chain, which increases the storage and computing overhead of blockchain. Compared with the traditional way, because the blockchain requires parallel storage and calculation of multiple miners, resource waste is bound to become noticeable. Individual schemes put this part off-chain but caused functional defects, such as the participants need a lot of offline communications

3. Preliminary

3.1. Bilinear Pairing. Let G_0 and G_1 be cyclic groups of order p and g be a generator of G_0. We call $e : G_0 \times G_0 \longrightarrow G_1$ is a bilinear paring if it is a map with the following properties:

(1) Bilinear: for all $g_1, g_2 \in G_0$ and $a, b \in Z_p$, there will be $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$

(2) Nondegenerate: there exists $g_0 \in G_0$, such that $e(g_0, g_0) \neq 1$

(3) Computable: there is an efficient algorithm to compute $e(g_1, g_2)$ for all g_1 and g_2

3.2. Linear Secret Sharing Scheme (LSSS). Let $P = \{P_1, \cdots, P_n\}$ be a set of participants, and (A, ρ) be an access structure. In the structure, A is an $l \times k$ matrix with ρ mapping its rows. An LSSS is composed of two polynomial-time algorithms:

(1) share$((A, \rho), s)$: to share a secret value s, it selects randomly $v_1, \cdots, v_{k-1} \in Z_p$. Let $\vec{v} = (s, v_1, \cdots, v_{k-1})^T$, and A_i be the vector as the ith row of matrix A, then, the secret share $\sigma_i = A_i \vec{v}$ belongs to party $\rho(i)$

(2) recover$(\omega, \{\sigma_i\}_{i\in\omega})$: it takes $\omega \in A$ and corresponding secret shares as inputs. If any $L \in \{i \mid \rho(i) \in \omega\}$ satisfies the access structure, a set of recovery coefficients $\{\mu_i\}_{i\in L}$ can be calculated so that $\sum_{i\in L} u_i \sigma_i = s$

3.3. Bloom Filter. Bloom filter is a space-efficient probabilistic data structure, proposed by Burton Howard Bloom in 1970 [31] that through an individual error rate in exchange for space-saving and query efficiency. A standard l_{BF}-bit Bloom filter includes a vector $\mathbf{V}$ with a length of l_{BF}, all bits of which are initialized to 0, k independent hash functions

$\{h_1, \cdots, h_k\}$, and each hash function has a uniform value in the range of $[0, l_{\mathrm{BF}} - 1]$. For each element $w_i (1 \le i \le n)$ in set $\mathbf{W} = \{w_1, \cdots, w_n\}$, set the corresponding position of $H_j(w_i)(1 \le j \le k)$ in the vector to 1. It is only necessary to determine whether an element all $H_j(w_i)$ in $\mathbf{V}$ are 1 or not to judge whether an element w is in the set $\mathbf{W}$. If not, there must be $w \notin W$; otherwise, there is a high probability $w \in \mathbf{W}$. (It should be noted that a high probability means that there is a false-positive rate of nonzero, but this possibility can be minimized by appropriate setting the value of l_{BF} and k.) A Bloom filter is composed of two algorithms:

(1) $\mathrm{BFGen}(\{h_1{}', \cdots, h_k{}'\}, \{w_1, \cdots, w_n\}) \longrightarrow \mathrm{BF}$: this algorithm generates an l_{BF}-bit Bloom filter by hashing $\{h_1, \cdots, h_k\}$ the data set $\mathbf{W} = \{w_1, \cdots, w_n\}$

(2) $\mathrm{BFVerify}(\mathrm{BF}, w, \{h_1{}', \cdots, h_k{}'\}) \longrightarrow (0, 1)$: this algorithm verifies whether the element w is in the set $\mathbf{W}$. If it returns 1, $w \in W$. Otherwise, $w \notin W$

3.4. ^{Blockchain}. The blockchain concept originated from Nakamoto's Bitcoin white paper [16], whose foundation is cryptography and P2P networks. Then, it organizes the data with a specific structure into blocks in a certain way and links these blocks into a chain in chronological order. Cryptography and consensus mechanisms together ensure the security and unforgeability of data. In short, as the underlying technology of cryptocurrency like Bitcoin, the blockchain is a trusted ledger with distributed computing capabilities that can process business credibly without a third-party organization.

3.5. Smart Contract. Initially, when it comes to blockchain, the only well-known applications are cryptocurrencies such as Bitcoin and Litecoin. What brings a qualitative change to the blockchain is that in 2013 Vitalik Buterin established the first public chain platform named Ethereum with a built-in Turing complete language [32] and inaugurated smart contract for the blockchain. Szabo defined smart contract as "a computerized transaction protocol that executes the terms of a contract" [33]. The smart contract in the blockchain is a piece of program code stored on the chain, which can be executed securely and reliably. On the one hand, the blockchain can utilize a programmable smart contract to implement more complex business logic. On the other hand, the blockchain can provide a trusted environment for executing a smart contract. The operating mechanism of the smart contract in the blockchain is shown in Figure 1.

As shown in the figure, the blockchain can be seen as a state machine triggered by transactions, and the ledger is a public world state starting from the Genesis Block. Users can create a transaction and broadcast it to the blockchain network from any node. All block producers will perform corresponding operations after receiving the transaction, and the consensus mechanism makes all nodes finally get a consistent result and update the world state.

Blockchain provides the following support for the execution of smart contract on-chain:

(i) Public status: every participant can inspect the smart contract's current world state on the public ledger

(ii) Timestamp server: the block height can be seen as a trusted timestamp that never stops

(iii) Trusted propagation channel: the sender can utilize the blockchain to spread the message, and the receiver will reliably receive the message shortly. The delivery traces will be recorded on-chain for auditing, and the records are credible and cannot be tampered with by anyone

3.6. Transactions of EOS. Account, address, and transaction are three essential components in the EOS blockchain [34]. Each user has an account that corresponds to multiple ECDSA key pairs expressed as (pk, sk), and each key pair represents different operation permission of the account. The private and public keys are used by users to sign and verify a transaction. Our definition of a transaction is consistent with our previous work [35, 36]:

$$\begin{aligned}\mathrm{Tx} = (&\mathrm{Ref}_{\mathrm{block}}, t, \mathrm{Sig}_U(\mathrm{Chain\_ID}, \mathrm{Tx})),\\ &\mathrm{Action}(\mathrm{Code}, \mathrm{Name}, \mathrm{Auth}_U, \mathrm{Data})),\end{aligned} \tag{1}$$

where $\mathrm{Ref}_{\mathrm{block}}$ refers to the height and id of a recently generated block, which prevents the transaction from being packaged on the fork chain and t is the expiration time of the transaction. $\mathrm{Sig}_U(\mathrm{Chain\_ID}, \mathrm{Tx})$ is the signature signed by the sender. Action is the operation performed by the transaction in which Code is the name of the smart contract, Name is the method to be invoked in the smart contract, Auth_U is used to verify whether the sender has the authority to call the method, and Data are the parameters. There may be multiple actions in one transaction. Smart contracts can also send actions to each other to call methods of other contracts, which is called inline communication, and the corresponding execution authority is the same as the original transaction.

3.7. Data Persistence of EOS. After the smart contract is executed, the occupied memory will be released, and all variable data in the program will be lost, so it is necessary to persist the data in smart contract. In the smart contact of Ethereum, data can only be stored in key-value pairs, which is difficult to meet more complex requirements. EOS imitates multi-index containers in Boost library and develops a C++ class: *eosio::multi_index* (from now on referred to as multi_index). Each multi_index can be regarded as a table in the traditional database. Each row of the table can store an object, and the object's attributes can be any C++ data type. Therefore, the table constructed by multi_index in EOS is no less flexible than traditional databases. A significant feature of multi_index is that a primary key can be set as the main index and 16 secondary indices. Users can obtain any of these indices and use the *emplace*, *erase*, *modify*, and *find* functions of the index to insert, delete, update, and select data.

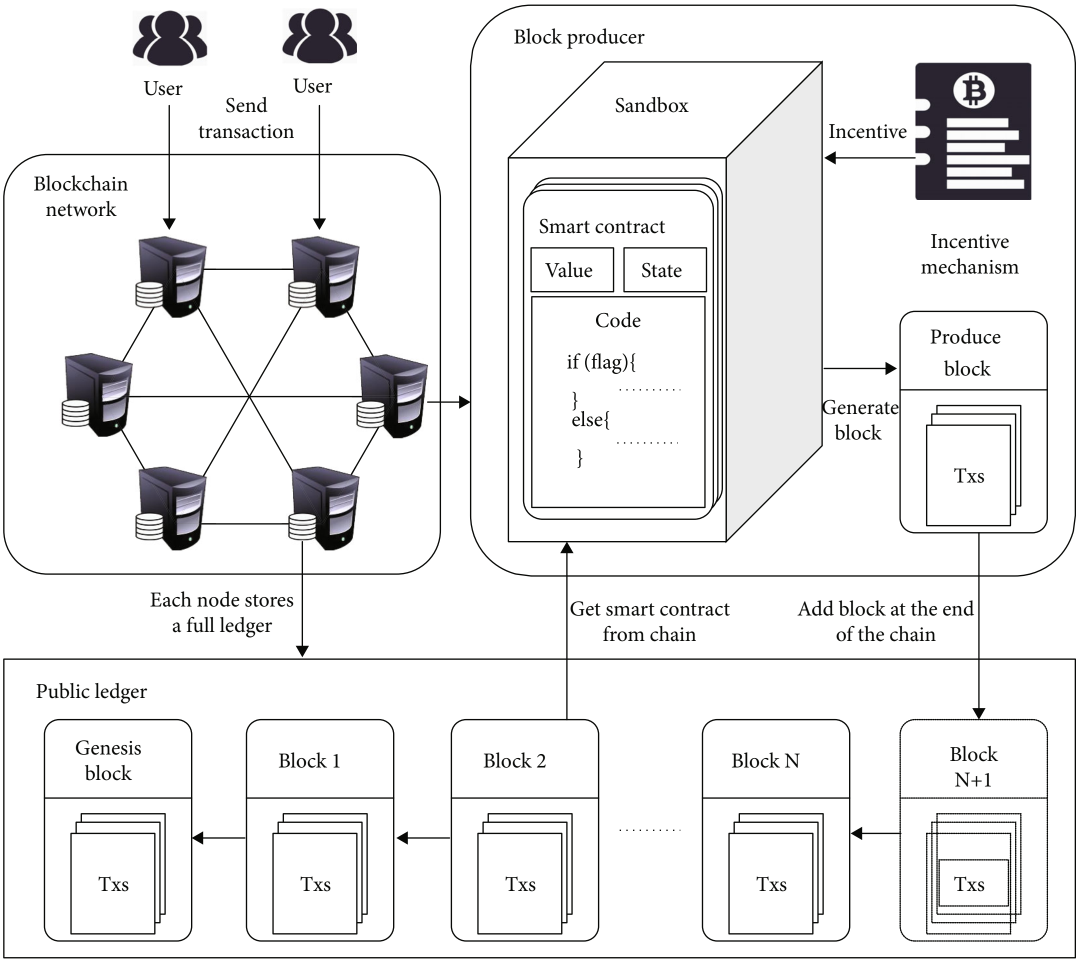


Figure 1: Operation mechanism of blockchain-based smart contract.

4. Overview of Proposed Scheme

This section will give an overview of our proposed scheme, including the system model and scheme design. The meanings of the symbols and abbreviations used in this paper are shown in Table 1.

4.1. System Model of BFR-SE. The scheme proposed in this paper is composed of four components: data owner (DO), data user (DU), searchable encryption service provider (SESP), and blockchain. The keywords and their corresponding index structure are encrypted and uploaded to the SESP after DO extracts the keyword set from the outsourced data set to prevent privacy disclosure. DO distribute keys for DUs through blockchain, and only the DUs whose attributes satisfy the access policy can search and obtain the original data. DU uses his private key to generate a search token according to the keywords he wants to retrieve. According to the search token provided by DU, SESP performs complex search calculation operations, then returns the search results to DU and obtains the revenue. The traces and additional evidence of each participant will be recorded on the blockchain, which cannot be destroyed or denied. DUs need to pay SESP for the service they use. If the SESP does not provide the correct result before the predetermined block height, DU can apply for arbitration, and the blockchain will make a judgment according to the auxiliary information and the additional evidence on-chain during the search. Then, the charge fee will be returned to the DU as compensation, together with a penalty on SESP. The specific functions and responsibilities of the four components are as follows:

(1) DO: the owner of the IoT device is also the owner of the data. Responsible for the system's initialization, including creating and deploying smart contracts in the scheme. DO needs to generate and distribute private keys for registered DUs according to their attributes. Besides, DO extracts keywords from the outsourced data files, generates the corresponding indices, and sets a reasonable access policy for the indices. DO is honest by default

(2) DU: according to the keywords he wants to retrieve, DU generates a search token with his private key and sends a search request to the SESP. At the same time, DU can judge whether the search results returned by SESP are correct or not

Table 1: The symbols and abbreviations involved in this paper.

No.	Symbol	Description
1	DO	The data owner
2	DU	The data user
3	SESP	The searchable encryption service provider
4	MSK	System master key
5	PK	System public parameters
6	$\varepsilon = (\varepsilon.\mathrm{Enc}, \varepsilon.\mathrm{Dec})$	An asymmetric encryption algorithm like ECC
7	$(Pk_{\mathrm{com}}, Sk_{\mathrm{com}})$	A pair of keys for algorithm ε
8	$(Pk_{\mathrm{sig}}, Sk_{\mathrm{sig}})$	A pair of keys for an algorithm like ECDSA
9	S	All general attribute set
10	ω	The attributes set of a specific DU
11	$\mathscr{P}$	Access policy
12	Sk_{ω}	The private attribute key of DU whose attributes set is ω
13	FID	The set of file identities
14	KW	The set of keywords
15	CKW	The set of ciphertext for keywords
16	I	The set of indices between keywords and data files
17	CI	The set of ciphertext for indices
18	AI	Auxiliary information
19	TOK	Search token
20	COMM	DO's commitment to his search request

(3) SESP: it has powerful computing capabilities and stores the ciphertext of indices provided by DO. It will perform complex search calculations on the indices according to DU's search token, then return the search results to DU. SESP may be malicious and may return partial or even wrong search results to save resources and defraud profits

(4) Blockchain: it stores the auxiliary information of the indices and intermediate evidence information. In the event of a dispute, arbitration can be conducted according to this information, and the malicious parties can be punished economically. In the absence of a third-party authoritative and trusted organization, the blockchain is the cornerstone of trust in the scheme. Additionally, blockchain can provide a reliable broadcast channel for each participant, which can be used by each party for information dissemination

The system model of our proposed scheme is shown in Figure 2.

Our scheme's searchable encryption algorithm is inspired by the scheme named VABKS (verifiable attribute-based keyword search) proposed by Zheng et al. [21]. We have optimized and extended it to support a multi-keyword search. The detailed description of each step in the flowchart is as follows:

(i) DO creates and deploys smart contracts on the blockchain. BFR-SE includes two smart contracts: PMContract and SEContract

(ii) DO generates the system master key and public parameters, as well as a pair of signature keys. Then, DO publishes the public parameters and public key for the signature to the smart contract, while the system master key and the private key for signature keep secret

(iii) SESP is registered in the SEContract, and a definite amount of deposit is required when registering. If SESP has fraudulent behavior, it will deduct part of the deposit as punishment

(iv) DU applies for registration in the PMContract, and he needs to provide his EOS account and a public key of ECC in which the EOS account is used for receiving compensation when SESP is dishonest

(v) DO generates the attribute key for DU according to his attributes set, then uses his public key to encrypt it and broadcast it to the blockchain. The ciphertext of the attribute key will be stored in the PMContract

(vi) DU obtains the ciphertext of his attribute key from PMContract and decrypts it locally by the corresponding private key

(vii) DO encrypts his data files and outsources them. (It can be uploaded to the cloud server, or IPFS, which is beyond the scope of this paper.) The returned address and the corresponding decryption key

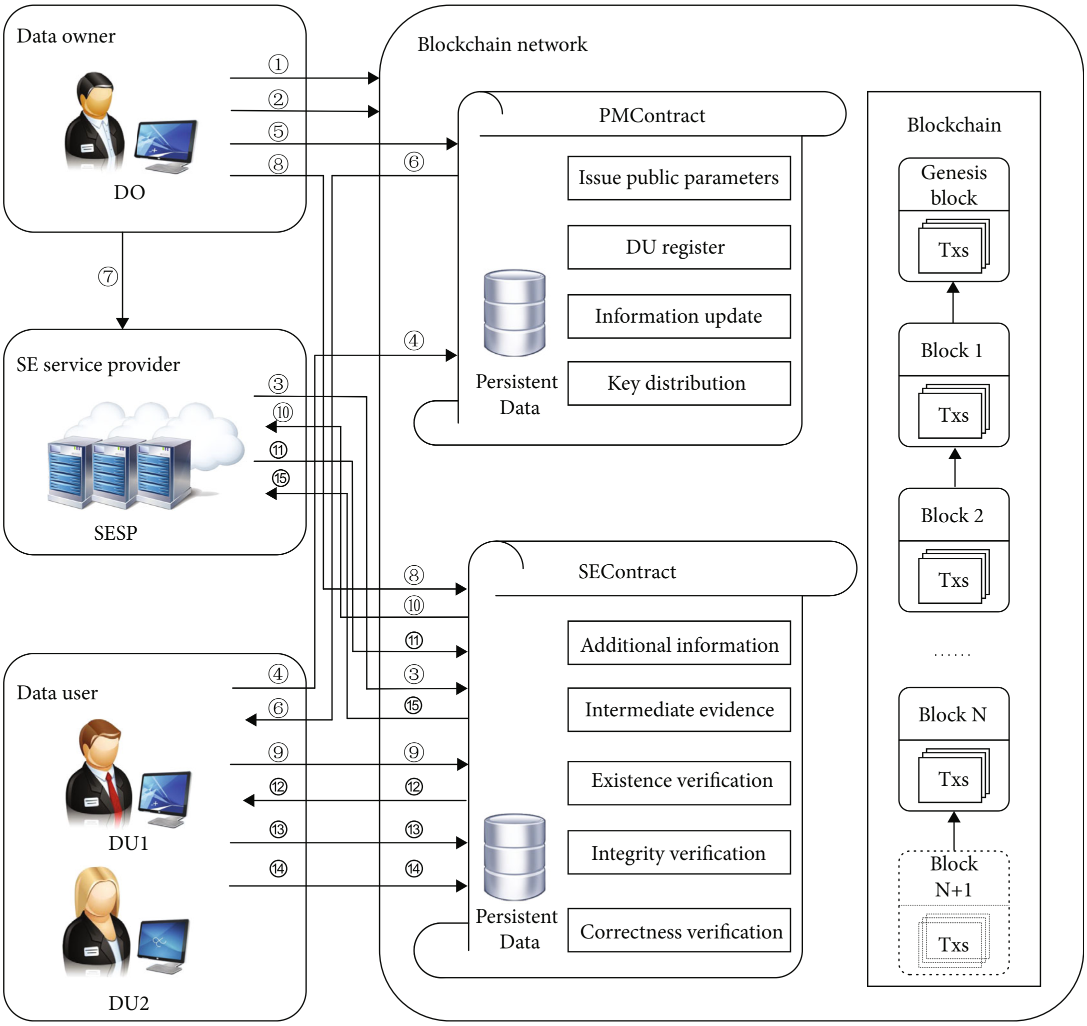


Figure 2: The system model of our scheme.

constitute the identity of the shared data. DO extracts the keywords set from the data files and builds indices for the keywords matching data files' identification, then generates auxiliary information simultaneously. DO uploads the indices to SESP

(viii) DO uploads the auxiliary information to SEContract

(ix) DU uses his private attribute key locally to generate the corresponding search token and additional commitment for his search request. Before searching, DU can check the amount of SESP deposit in PMContract. If the deposit is not enough to pay the penalty when doing evil, DU can choose not to continue

(x) SESP obtains the search request from SEContract, uses DU's search token and the indices uploaded by DO to execute the search algorithm, and returns the search results

(xi) SESP uploads search results to SEContract. It should be noted that SESP needs to complete the search and upload the results before the preagreed block height

(xii) DU gets his search results from SEContract

(xiii) If SESP does not provide any results before the specified block height, DU can withdraw his service fee through SEContract without any loss

(xiv) If DU believes that the search results returned by SESP are incorrect, DU could initiate a request for arbitration, and the blockchain will determine whether SESP has performed correctly

(xv) SESP gets his deposit from the contract

4.2. Detail Design of BFR-SE. BFR-SE consists of the following phase: initialization phase, apply and register phase, build index phase, token generation phase, search phase, verification phase, and withdraw phase. This section will describe each phase's detailed design in BFR-SE and give the corresponding relationship with the process steps in the flowchart.

(1) Initialization phase

The main work of the initialization phase is that DO creates smart contracts and deploys them on the blockchain. Then, DO generates the system master key MSK and public parameters PK locally. The core algorithm of this phase is $\mathrm{Setup}(1^{\lambda}) \longrightarrow (\mathrm{MSK}, \mathrm{PK}, Sk_{\mathrm{sig}}, Pk_{\mathrm{sig}})$, which is run by DO locally. The algorithm's input is a security parameter 1^{λ}, and the outputs are MSK, PK, and a pair of keys for signature. After that, the DO publishes PK and Pk_{sig} to smart contracts and keep MSK, and Sk_{sig} secret locally. The corresponding steps in the system flowchart are ① and ②.

(2) Apply and Register phase

The apply and register phase's primary work is to complete the registration of SESP and DUs, including that DO distributes private attribute key for each DU. SESP needs to transfer a certain amount of system tokens to SEContract when applying for registration. If the amount of deposit is less than fines when doing evil, DU can choose not to use the search service. When DU applies for registration, it needs to provide a public key of ECC. After that, the DO generates a private attribute key for the DU according to his attributes set. The core algorithm is $\mathrm{KeyGen}(\mathrm{MSK}, \mathrm{PK}, \omega) \longrightarrow Sk_{\omega}$, which is run by DO locally. The algorithm inputs MSK, PK, the attribute set ω of DU, and DU's public key Pk_{com} and outputs the private attribute key Sk_{ω} of DU. Then, DO uses the public key provided by DU when applying for registration to encrypt Sk_{ω} and obtain the ciphertext of Sk_{ω}, $CSk_{\omega} = \varepsilon.\mathrm{Enc}_{Pk_{com}}(Sk_{\omega})$. DO uploads CSk_{ω} to the SEContracts, so that DU can securely obtain and decrypt it to get his private attribute key Sk_{ω}. The corresponding steps in the system flowchart are ③, ④, ⑤, and ⑥.

(3) Build index phase

The main work of the build index phase is that DO encrypts the sharing data and outsource it. Take IPFS as an example, we can use the returned address and key to identify the data. After that, DO extracts the keywords set from the sharing data, builds indices for all the sharing data with the same keyword set, and generates the auxiliary information. DO sends the indices to SESP and uploads the auxiliary information to SEContract. The corresponding steps in the system flowchart are ⑦ and ⑧. It consists of the following three subalgorithms:

(a) $\mathrm{EncryFile}(\{F_{\eta}\}_{1\le\eta\le d}) \longrightarrow (\{\mathrm{FID}_{\eta}\}_{1\le\eta\le d})$

This algorithm is run by DO. For each element in $\{F_{\eta}\}_{1\le\eta\le d}$ where d denotes the number of the sharing data, DO encrypts it by the key key_{η} and outsource it. Taking IPFS as an example, the returned address of F_{η} is $\mathrm{href}_{F_{\eta}}$, and the sharing data's identity is $\mathrm{FID}_{\eta} = \mathrm{IDGen}(\mathrm{key}_{\eta}, \mathrm{href}_{\eta})$. The algorithm's final output is the identity set $\mathrm{FID} = \{\mathrm{FID}_{\eta}\}_{1\le\eta\le d}$. IDGen is an encryption function module defined by DO, which is not the focus of this paper, so the flowchart does not present it.

(b) $\mathrm{IndexGen}(\mathrm{KW}, \mathrm{FID}) \longrightarrow I$

DO runs this algorithm, and the main work is to establish the corresponding indices based on sharing data and the relevant keywords. At first, DO need to extract the keywords KW_{η} for each F_{η} in $\{F_{\eta}\}_{1\le\eta\le d}$, then $\mathrm{KW} = \mathrm{KW}_1 \cup \mathrm{KW}_2 \cdots \cup \mathrm{KW}_d$. For $\forall \mathrm{KW}_{\tau} = \{\mathrm{kw}_1, \cdots, \mathrm{kw}_m\}$ and $\mathrm{KW}_{\tau} \in \mathrm{KW}$ $(\mathrm{KW}_{\tau} \neq \varnothing, 1 \le \tau \le n)$, if the corresponding data set is FID_{τ}, then use all the elements in FID_{τ} as leaf nodes to generate a Merkle Tree, and the root is $\mathrm{MerkleRoot}_{\tau}$. We defined that $I_{\tau} = (\mathrm{KW}_{\tau}, \mathrm{MerkleRoot}_{\tau}, \mathrm{FID}_{\tau})$, and the final indices of the keywords set KW matching the sharing data FID is $I = \{I_{\tau}\}_{1\le\tau\le n}$ in which n is the number of indices.

(c) $\mathrm{Encrypt}(I, \mathscr{P}, \mathrm{PK}) \longrightarrow (\mathrm{CI}, \mathrm{AI})$

DO runs this algorithm and the primary work is to encrypt the indices by a specific access policy to generate the ciphertext of indices CI and the auxiliary information AI. The inputs are the indices I, the access policy $\mathscr{P}$, and the public system parameters PK, while the outputs are CI and AI. DO encrypts the keywords of each I_{τ} in I to get the ciphertext of the keywords CKW_{τ}. DO signs CKW_{τ} and $\mathrm{MerkleRoot}_{\tau}$ by his private key to ensure the integrity of the index. DO uploads the ciphertext $\mathrm{CI} = \{\mathrm{CI}_{\tau}\}_{1\le\tau\le n}$ to SESP, and the corresponding auxiliary information AI is uploaded to SEContract. For example, with four files, the data structure of an index is shown in Figure 3.

(4) Token generation phase

The main work of the token generation phase is that DU uses his private attribute key to call the trapdoor function to generate a search token and commitment for the searching keywords. $\mathrm{TokenGen}(Sk_{\omega}, \mathrm{KW}_{\mathrm{search}}) \longrightarrow (\mathrm{TOK}, \mathrm{COMM})$ is the core algorithm whose inputs are the private attribute key Sk_{ω} of DU, and the keywords set $\mathrm{KW}_{\mathrm{search}}$ retrieved, and outputs are search token TOK and the commitment COMM. The commitment is to prove that DU did search for the keywords set provided by him when arbitrating. After that, DU uploads TOK and COMM to SEContract and pays the fee simultaneously. The corresponding steps in the system flowchart is ⑨.

(5) Search phase

The search phase's primary work is to use the search token to retrieve the ciphertext of indices uploaded by DO and return the successful matching results. The core algorithm of this phase is $\mathrm{TEST}(\mathrm{TOK}, \mathrm{CI}_{\tau}) \longrightarrow \{0, 1\}$, which is run by SESP locally. This algorithm's inputs are the search token TOK and the ciphertext of index CI_{τ}, and the output is 0 or 1. If the output is 1, then the match is booming, and the search result is $\mathrm{CI}_{\mathrm{result}}$. SESP needs to upload $\mathrm{CI}_{\mathrm{result}}$ to SEContract before the preagreed time. Otherwise, DU can claim back the charge fee. The corresponding steps in the system flowchart are ⑩, ⑪, ⑫, and ⑬.

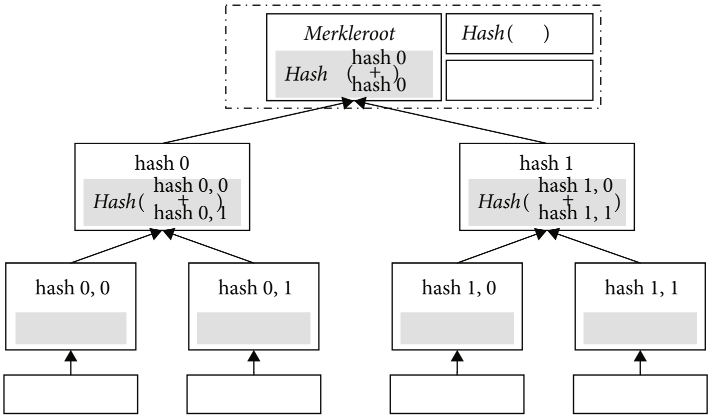


Figure 3: An example of the data structure for an index.

(6) Verification phase

The verification phase's primary work is to verify the search results returned by SESP according to the results and the auxiliary information uploaded by DO, including the verification of existence, integrity, and correctness. If the verification result is that the SESP has done evil, economic punishment will be imposed on SESP. This algorithm is executed by the blockchain, corresponding to step ⑭ in the flowchart. It can be subdivided into three subphases as follows:

(a) $\text{ExistenceVerify}(\text{AI}, \text{random}, \{H_2(W_j{}')\}_{1\le j\le m}, \text{COMM}) \longrightarrow \{0,1\}$

When the search result returned by SESP is *null*, the algorithm can verify the existence of the sharing data searched by DU. The algorithm's inputs are the auxiliary information AI, a random number related to the search token and $\{H_2(W_j{}')\}_{1\le j\le m}$, and the commitment corresponding to the search request. The output is 0 or 1.

(b) $\text{IntegrityVerify}(\text{CI}_{\text{result}}) \longrightarrow \{0,1\}$

This algorithm can verify the integrity of the search results returned by SESP and prevent SESP from returning only partial or even forged results. The input of this algorithm is the search result $\text{CI}_{\text{result}}$, and the output is 0 or 1.

(c) $\text{CorrectnessVerify}(\text{TOK}, \text{CI}_{\text{result}}) \longrightarrow \{0,1\}$

This algorithm can verify the correctness of the search results returned by SESP and prevent SESP from returning the wrong results. This algorithm's inputs are the search token TOK and the search result $\text{CI}_{\text{result}}$. The output is 0 or 1.

(7) Withdraw phase

The main work of this phase is that each participant withdraws their coins from smart contract. The SESP's coin includes the deposit at the time of registration and DU's payment for using the search service. The coin of DU is mainly compensation, which comes from the penalty of SESP. It should be noted that each fee needs a freeze period, during which DU can apply for arbitration on the blockchain. Only after the freezing period has passed can SESP withdraw this fee from the contract. The corresponding steps in the system flowchart are ⑮.

5. Implementation Details of Proposed Scheme

To achieve our goal, we constructed an attribute-based searchable encryption algorithm that supports multikeyword search and combined with the EOS blockchain platform to realize our fair and reliable scheme. This section will elaborate on the details of our smart contracts deployed on EOS and the concrete construction of BFR-SE.

5.1. Smart Contract Design.

In order to make the logic clearer, we divide the smart contract in the scheme into two parts, PMContract and SEContract. We use *_self* to represent the account of smart contract itself and *_self.asset* to represent the balance in the contract. Let require_auth be a function that represents which account's permission is needed to continue. We will describe the two smart contracts in detail in this section.

5.1.1. Participant Management Contract (PMContract).

The PMContract is composed of five interfaces: *SetSPK*, *Register*, *GetPK*, *SetSK*, and *GetSK*. We initialize PMContract as follows:

Let three-tuple $(\text{Account}_{\text{user}}, \text{Pk}_{\text{com}}, CSk_\omega)$ denote a DU and create a multi_index named *table_user*, in which $\text{Account}_{\text{user}}$ is an EOS account of DU, Pk_{com} is a public key of DU, and CSk_ω is the private attribute key of DU. Let $\text{Account}_{\text{user}}$ be the primary key of *table_user*, whose corresponding index is *account_idx*. Let PK denote the system public parameters.

(1) *SetSPK*: when PMContract receives action (PMContract, *SetSPK*, Auth, (*pk*)), this function will be triggered to execute. It can only be invoked by DO to set and update the public system parameters

(2) *Register*: when PMContract receives action (PMContract, Register, Auth, (A_{user}, Pk_{com})), this function will be triggered to execute. It is invoked by DU to apply for registration in the system. The detail of this function can be seen in Algorithm 1

(3) *GetPK*: when PMContract receives action (PMContract, GetPK, Auth, (A_{user})), this function will be triggered to execute

(4) *SetSK*: when PMContract receives action (PMContract, SetSK, Auth, (A_{user})), this function will be triggered to execute. It is used for DO distributing private attribute keys to DUs. The detail of this function can be seen in Algorithm 2

(5) *GetSK*: when PMContract receives action (PMContract, GetSK, Auth, (A_{user})), this function will be triggered to execute

5.1.2. Searchable Encryption Contract (SEContract). The SEContract consists of 13 functions: *SetPK*, *SetAI*, *Deposit*, *SearchRequest*, *SendResult*, *ExistenceVerify*, *IntegrityVerify*, *CorrectnessVerify*, *GetFeeSESP*, *IsVerifyRound*, *IsResultReady*, *Compensate*, and *CommitVerify*. We initialize SEContract as follows:

Let six-tuple (Account$_{\text{user}}$, SerialNum, TOK, COMM, Height, Coin) be a search request initiated from DU and create a multi_index named *search_table*, in which Account$_{\text{user}}$ is the account of DU, SerialNum is the serial number of the search request, TOK is the search token, COMM is the commitment of DU for the request, Height is the block height when the request is initiated, and Coin is the charge fee paid by the user for the service. Let Account$_{\text{user}}$ be the primary key of *search_table*, and the corresponding index is *search_idx*. Let three-tuple (Account$_{\text{user}}$, SerialNum,CI$_{\text{result}}$) be a result returned from SESP and create a multi_index named result_table, in which Account$_{\text{user}}$ and SerialNum are to match the search request in *search_table*, and CI$_{\text{result}}$ denotes the search result. Let Account$_{\text{user}}$ be the primary key of *result_table*, and the corresponding index is *result_idx*. Let Account$_{\text{sesp}}$ be the account of SESP and Deposit be the balance of SESP in the contract. Let d represent the fee that the user needs to pay for each search and the amount of penalty when SESP does evil. Let *round_height* represent the time required for search round and verification round, BF be the auxiliary information which is a Bloom filter, and PK$_{\text{Sig}}$ be the public key for the signature of DO.

(1) *SetPK*: when SEContract receives action (SEContract, SetPK, Auth, (pk)), this function will be triggered to execute

(2) *SetAI*: when SEContract receives action (SEContract, SetAI, Auth, (AI)), this function will be triggered to execute

```
Input: $A_{user}$, $Pk_{com}$
Output: void
require_auth($A_{user}$)
$u = account\_idx$.find($A_{user}$)
if ($u$ == $null$) then
    u.$Pk_{com}$= $Pk_{com}$
    account_idx.modify(u)
else
    u.Account$_{user}$= $A_{user}$
    u.$Pk_{com}$= $Pk_{com}$
    $account\_idx$.emplace($u$)
end
```

Algorithm 1: Register.

(3) *Deposit*: when SEContract receives action (SEContract, Deposit, Auth, (A_{sesp}, coin)), this function will be triggered to execute. The detail of this function can be seen in Algorithm 3

(4) *SearchRequest*: when SEContract receives action (SEContract,SearchRequest,Auth, (A_{user},Sn,TOK, COMM)), this function will be triggered to execute. The detail of this function can be seen in Algorithm 4

(5) *SendResult*: when SEContract receives action (SEContract, SendResult, Auth, (A_{user}, Sn, CI$_{\text{result}}$)), this function will be triggered to execute. It can only be invoked by SESP. The detail of this function can be seen in Algorithm 5

(6) *ExistenceVerify*: when SEContract receives action (SEContract, ExistenceVerify, Auth, (A_{user},random, $\{H_2(W_j{}')\}_{1\leq j\leq m}$)), this function will be triggered to execute. The detail of this function can be seen in Algorithm 6

(7) *IntegrityVerify*: when SEContract receives action (SEContract, IntegrityVerify, Auth, (A_{user})), this function will be triggered to execute. The detail of this function can be seen in Algorithm 7

12 **end**

(8) *CorrectnessVerify*: when SEContract receives action (SEContract, CorrectnessVerify, Auth, (A_{user})), this function will be triggered to execute. The detail of this function can be seen in Algorithm 8

(9) *GetFeeSESP*: when SEContract receives action (SEContract, GetFeeSESP, Auth, (A_{user})), this function will be triggered to execute. The detail of this function can be seen in Algorithm 9

```
Input: $A_{\text{user}}$,$CSk_\omega$
Output: bool
require_auth(_self)
u = account_idx.find($A_{\text{user}}$)
if u == null then
    return false
else
    u.$CSk_\omega$=$CSk_\omega$
    account_idx.modify($A_{\text{user}}$)
    return true
end
```

Algorithm 2: SetSK.

(10) *IsVerifyRound*: this is a private function and can only be called internally by the contract itself. The detail of this function can be seen in Algorithm 10

(11) *IsResultReady*: this is a private function and can only be called internally by the contract itself. The detail of this function can be seen in Algorithm 11

(12) *Compensate*: this is a private function and can only be called internally by the contract itself. The detail of this function can be seen in Algorithm 12

(13) *CommitVerify*: this is a private function and can only be called internally by the contract itself. The core of this function is the algorithm TEST in the search phase. For detailed implementation, see the concrete construction of BFR-SE in the next section

5.2. Concrete Construction of BFR-SE. This section shows the concrete construction of BFR-SE, including the algorithms to be executed at each phase and how each participant interacts with the EOS blockchain. Our initialization is as follows:

Let G_0 and G_1 be cyclic groups of order p, and g be a generator of G_0. Let $e : G_0 \times G_0 \longrightarrow G_1$ be a bilinear pairing, $S = \{1, \cdots, l\}$ be the set of all attributes, $\{h_1{}', \cdots, h_k{}'\}$ be k general and distinct hash functions. $H_1 : \{0, 1\}* \longrightarrow G$ and $H_2 : \{0, 1\}* \longrightarrow Z_p$ are also two hash functions.

(1) Setup(1^λ)

Firstly, DO picks $a, b, c \longleftarrow Z_p$ randomly. For each attribute $\{i \mid i \in S\}$, compute that $\{h_i = H_1(i) \mid i \in S\}$.

The public system parameters are PK:

$$\mathrm{PK} = \left\{g^a, g^b, g^c, \{h_i\}_{i\in S}\right\}. \tag{2}$$

The system master key is MSK:

$$\mathrm{MSK} = (a, b, c). \tag{3}$$

DO randomly selects a key pair of ECDSA which be denoted $(Sk_{\text{sig}}, Pk_{\text{sig}})$, then keeps MSK and Sk_{sig} secret and sends the following two transactions to the blockchain:

$$\begin{gathered}(\mathrm{Ref}_{\mathrm{block}}, t, \mathrm{Sig}_{\mathrm{DO}}(\mathrm{Chain\_ID}, \mathrm{Tx}),\\ \mathrm{Action}(\mathrm{PMContract}, \mathrm{SetSPK}, \mathrm{Auth}_{\mathrm{DO}}, (\mathrm{PK}))),\end{gathered} \tag{4}$$

$$\begin{gathered}(\mathrm{Ref}_{\mathrm{block}}, t, \mathrm{Sig}_{\mathrm{DO}}(\mathrm{Chain\_ID}, \mathrm{Tx}),\\ \mathrm{Action}\left(\mathrm{SEContract}, \mathrm{SetPK}, \mathrm{Auth}_{\mathrm{DO}}, \left(\mathrm{Pk}_{\mathrm{Sig}}\right)\right)),\end{gathered} \tag{5}$$

(2) KeyGen$(\mathrm{MSK}, \omega) \longrightarrow Sk_\omega$

At first, DO sends the following transaction to the blockchain to obtain the DU's public key:

$$\begin{gathered}(\mathrm{Ref}_{\mathrm{block}}, t, \mathrm{Sig}_{\mathrm{DO}}(\mathrm{Chain\_ID}, \mathrm{Tx}),\\ \mathrm{Action}(\mathrm{PMContract}, \mathrm{GetPK}, \mathrm{Auth}_{\mathrm{DO}}, (\mathrm{A}_{\mathrm{user}})))\end{gathered} \tag{6}$$

Let the attribute set of DU be ω and $\omega \in S$, then randomly pick $t \longleftarrow Z_p$ and compute $K_1 = g^{(ac-t)/b}$, $K_2 = g^t$. For each $i \in \omega$, it computes that $K_{3,i} = h_i^t$. The private attribute key of DU is Sk_ω:

$$Sk_\omega = \left(K_1, K_2, \left\{K_{3,i}\right\}_{i\in\omega}\right). \tag{7}$$

Then, encrypt it with the public key of DU:

$$CSk_\omega = \varepsilon.\mathrm{Enc}_{Pk_{com}}(Sk_\omega). \tag{8}$$

DO sends the following transaction to the blockchain:

$$\begin{gathered}(\mathrm{Ref}_{\mathrm{block}}, t, \mathrm{Sig}_{\mathrm{DO}}(\mathrm{Chain\_ID}, \mathrm{Tx}),\\ \mathrm{Action}(\mathrm{SEContract}, \mathrm{SetSK}, \mathrm{Auth}_{\mathrm{DO}}, (\mathrm{A}_{\mathrm{user}}, \mathrm{CSk}_\omega))),\end{gathered} \tag{9}$$

(3) Encrypt$(I, (A_{l\times k}, \rho), PK) \longrightarrow (\mathrm{CI}, \mathrm{AI})$

For $\forall I_\tau \in I$, $I_\tau = (\mathrm{KW}_\tau, \mathrm{MerkleRoot}_\tau, \mathrm{FID}_\tau)$, randomly picks $r, s \longleftarrow Z_p$ and computes $C_0 = g^{cr}$, $C_1 = g^{bs}$. Let $(A_{l\times k}, \rho)$ be an access structure. It randomly chooses $y_2, y_3, \cdots, y_k \in Z_p$ and sets $\vec{\mathbf{v}} = (s, v_2, \cdots, v_k)^T$. For each $i = 1$ to l, it calculates $\sigma_i = A_i \times \vec{\mathbf{v}}$. Then, randomly picks $r_1, \cdots, r_l \in Z_p$ and performs the following calculations for each attribute:

$$C_{2,i} = g^{a\sigma_i} h_i^{-r_i}, C_{3,i} = g^{r_i}. \tag{10}$$

Let m be the size of the keyword set KW_τ. For each W_j

```
Input: $A_{sesp}$, coin
Output: void
require_auth($A_{sesp}$)
if $Account_{sesp}$== null then
    $Account_{sesp}$=$A_{sesp}$
end
send action (eosio.token, transfer, Auth, ($A_{sesp}$,_self, coin))
Deposit = Deposit + coin
```

Algorithm 3: Deposit.

```
Input: $A_{user}$, Sn,TOK, COMM
Output: void
require_auth($A_{user}$)
s = search_idx.find($A_{user}$)
send action (eosio.token, transfer, Auth, ($A_{user}$,_self, d))
if s == null then
    s.$Account_{user}$ = $A_{user}$
    s.SerialNum = Sn
    s.TOK = TOK
    s.COMM = COMM
    s.Coin = d
    s.Height = getCurrentHeight()
    search_idx.emplace(s)
else if (getCurrentHeight() > (s.Height+2 × round_height)) then
    s.SerialNum = Sn
    s.TOK = TOK
    s.COMM = COMM
    s.Coin = d
    s.Height = getCurrentHeight()
    search_idx.modify(s)
end
```

Algorithm 4: SearchRequest.

```
Input: $A_{user}$, Sn, $CI_{result}$
Output: void
require_auth($A_{sesp}$)
rlt=result_idx.find($A_{user}$)
if rlt!=null then
    rlt.SerialNum=Sn
    rlt.$CI_{result}$=$CI_{result}$
    result_idx.modify(rlt)
else
    rlt.$A_{user}$=$A_{user}$
    rlt.SerialNum=Sn
    rlt.$CI_{result}$=$CI_{result}$
    result_idx.Emplace(rlt)
end
```

Algorithm 5: SendResult.

in KW_τ, $1 \le j \le m$, perform the following calculation:

$$C_{4,j} = g^{a(r+s)/m} g^{brH_2(W_j)}. \tag{11}$$

The ciphertext of KW_τ will be

$$CKW_\tau = \left\{ C_0, C_1, \left\{ C_{2,i}, C_{3,i} \right\}_{i\in(1,\cdots,l)}, \left\{ C_{4,j} \right\}_{j\in(1,\cdots,m)} \right\}. \tag{12}$$

DO signs the ciphertext of the index by his private key:

$$Sig_\tau = SigGen\left(Sk_{Sig}, MerkleRoot_\tau \| SHA256(CKW_\tau) \right). \tag{13}$$

Let CI_τ be the ciphertext corresponding to KW_τ:

$$CI_\tau = (CKW_\tau, MerkleRoot_\tau, Sig_\tau, FID_\tau). \tag{14}$$

Set $HKW_\tau = SHA256(H_2(W_1)\|, \cdots, \|H_2(W_m))$, $W_j \in KW_\tau, 1 \le j \le m$.

```
Input: $A_{user}$, random, $\{H_2(W_j')\}_{1\le j\le m}$
Output: void
bool IsVr = IsVerifyRound($A_{user}$)
bool IsRd = IsResultReady($A_{user}$)
if IsVr== true && IsRd==false then
    Compensate($A_{user}$)
else if IsVr== true && IsRd==true then
    if CommitVerify($A_{user}$,random, $\{H_2(W_j')\}_{1\le j\le m}$) == true then
        bool isExist = BFVerify(BF, SHA256($\{H_2(W_j')\}_{1\le j\le m}$))
        if isExist == false then
            Compensate($A_{user}$)
        end
    end
```

Algorithm 6: ExistenceVerify.

```
Input: $A_{user}$
Output: void
bool IsVr = IsVerifyRound($A_{user}$)
bool IsRd = IsResultReady($A_{user}$)
if IsVr== true && IsRd==false then
    Compensate($A_{user}$)
else if IsVr== true && IsRd==true then
    rlt = result_idx.find($A_{user}$)
    bool isMK = VerifyMerkleRoot(rlt.$\text{MerkleRoot}_\tau$, rlt.FID)
    bool isSig = SigVerify(rlt.$\text{MerkleRoot}_\tau$|| SHA256(rlt.$\text{CKW}_\tau$), rlt.$\text{Sig}_\tau$, $\text{PK}_{\text{Sig}}$))
    if isMK && isSig then
        throw
     else
        Compensate($A_{user}$)
     end
end
```

Algorithm 7: IntegrityVerify.

```
Input: $A_{user}$
Output: void
bool IsVr = IsVerifyRound($A_{user}$)
bool IsRd = IsResultReady($A_{user}$)
if IsVr== true && IsRd==false then
    Compensate($A_{user}$)
else if IsVr== true && IsRd==true then
    s = search_idx.find($A_{user}$)
    rlt = result_idx.find($A_{user}$)
    bool isCv = Test(s.TOK,rlt.$\text{CKW}_\tau$)
    if isCv == false then
        Compensate($A_{user}$)
    end
end
```

Algorithm 8: CorrectnessVerify.

Finally, the ciphertext of the indices and the auxiliary information will be as follows:

$$\text{CI} = \{\text{CI}_\tau\}_{1\le\tau\le n}$$
$$\text{AI} = \text{BFGen}\left(\left\{h_1', \cdots, h_k'\right\}, \{\text{HKW}_1, \cdots, \text{HKW}_n\}\right) \longrightarrow \text{BF}. \tag{15}$$

After that, DO uploads CI to SESP, and the auxiliary information AI is uploaded to the blockchain by sending the following transaction:

$$\begin{gathered}(\text{Ref}_{\text{block}}, t, \text{Sig}_{\text{DO}}(\text{Chain\_ID}, \text{Tx}),\\ \text{Action}(\text{SEContract}, \text{SetAI}, \text{Auth}_{\text{DO}}, (\text{AI})),\end{gathered} \tag{16}$$

(4) $\text{TokenGen}(Sk_\omega, \text{KW}_{\text{search}}) \longrightarrow (\text{TOK}, \text{COMM})$

```
Input: $A_{\text{user}}$
Output: void
s = search_idx.find($A_{\text{user}}$)
if (getCurrentHeight() > s.Height+2 × round_height)& & (s.Coin>0) then
    Deposit = Deposit + s.Coin
    s.Coin =0
    search_idx.modify(s)
end
```

Algorithm 9: GetFeeSESP.

Firstly, DU obtains his ciphertext of private attribute key CSk_ω by sending the following transaction:

$$\begin{gathered}(\text{Ref}_{\text{block}}, t, \text{Sig}_{\text{user}}(\text{Chain\_ID}, \text{Tx}), \\ \text{Action}(\text{SEContract}, \text{GetSK}, \text{Auth}_{\text{user}}, (\text{A}_{\text{user}})).\end{gathered} \tag{17}$$

DU decrypts it to get the private attribute key:

$$Sk_\omega = \varepsilon.\text{Dec}_{Sk_{com}}(CSk_\omega). \tag{18}$$

Then, DU randomly picks $\pi \longleftarrow Z_p$. Let m be the size of $\text{KW}_{\text{search}}$ and compute it as follows:

$$\begin{gathered}\text{tok}_1 = g^{a\pi} \prod_{j=1}^{m} g^{b\pi H_2\left(W_j'\right)}, \\ \text{tok}_2 = g^{c\pi}, \\ \text{tok}_3 = K_1^\pi = g^{(ac-t)\pi/b}, \\ \text{tok}_4 = K_2^\pi = g^{\pi t},\end{gathered} \tag{19}$$

For each $i \in \omega$, it computes $H_i = K_{3,i}^\pi = h_i^{t\pi}$.Set $\text{tok}_5 = \{K_{3,i}^\pi\}_{i\in\omega} = \{h_i^{\pi t}\}_{i\in\omega}$, and the search token will be TOK:

$$\text{TOK} = \{\text{tok}_1, \text{tok}_2, \text{tok}_3, \text{tok}_4, \text{tok}_5\}. \tag{20}$$

It calculates the commitment of the search request as follows:

$$\begin{gathered}\text{HKW} = \text{SHA256}\left(H_2\left(W_1{}'\right)\Big\|, \cdots, \Big\|H_2\left(W_m{}'\right)\right), \\ \text{COMM} = \text{SHA256}(\pi\|\text{HKW}).\end{gathered} \tag{21}$$

Finally, DU picks $Sn \longleftarrow Z_q$ randomly as the serial number of the search request and sends the following transaction:

$$\begin{gathered}(\text{Ref}_{\text{block}}, t, \text{Sig}_{\text{user}}(\text{Chain\_ID}, \text{Tx}), \\ \text{Action}(\text{SEContract}, \text{SearchRequest}, \text{Auth}_{\text{user}}, (A_{\text{user}}, \text{Sn}, \text{TOK}, \text{COMM})),\end{gathered} \tag{22}$$

(5) $\text{TEST}(\text{TOK}, \text{CI}_\tau) \longrightarrow \{0, 1\}$

After SESP receives the TOK from DU, it will compare each row of CI. Firstly, it is judged whether the number of keywords m in CI_τ is the same as that in TOK. If different, compare the next row.

Assuming that the attribute set of DU satisfies the access policy, set μ_i be the recovery coefficient of the ith row in $A_{l\times k}$ and calculate as follows:

$$E = \prod_{i\in\omega} (e(C_{2,i}, \text{tok}_4) e(C_{3,i}, H_i))^{\mu_i}. \tag{23}$$

Then, determine whether the following two formulas (4) and (5) are equal. If it is, it returns 1. Otherwise, it returns 0.

$$\begin{gathered}e(C_0, \text{tok}_1) e(\text{tok}_3, C_1) E, \\ e\left(\prod_{j=1}^{m} C_{4,j}, \text{tok}_2\right).\end{gathered} \tag{24}$$

If the result is found successfully, then $\text{CI}_{\text{result}} = \text{CI}_\tau$; otherwise, $\text{CI}_{\text{result}} = \text{null}$. SESP sends the following transaction to the blockchain.

$$\begin{gathered}(\text{Ref}_{\text{block}}, t, \text{Sig}_{\text{SESP}}(\text{Chain\_ID}, \text{Tx}), \\ \text{Action}(\text{SEContract}, \text{Send Result}, \text{Auth}_{\text{SESP}}, (A_{\text{user}}, \text{Sn}, \text{CI}_{\text{result}})).\end{gathered} \tag{25}$$

(6) Verification

If DU believes that there are problems with the search results returned by SESP during the verification round, DU can initiate a request to the blockchain within this period and request the blockchain to make a judgment.

(a) Existence

When the result is null, DU can send the following transaction to the blockchain:

$$\begin{gathered}(\text{Ref}_{\text{block}}, t, \text{Sig}_{\text{user}}(\text{Chain\_ID}, \text{Tx}), \\ \text{Action}\left(\text{SEContract}, \text{ExistenceVerify}, \text{Auth}_{\text{user}}, \left(A_{\text{user}}, \pi, \left\{H_2\left(W_j{}'\right)\right\}_{1\le j\le m}{}'\right)\right).\end{gathered} \tag{26}$$

After the contract receives the transaction, it will first verify the DU's previous commitment to prevent DU from

```
Input: $A_{user}$
Output: bool
$s = search\_idx$.find($A_{user}$)
if ($getCurrentHeight() > (s.Height + round\_height)$) &&
    ($getCurrentHeight() < (s.Height + 2 \times round\_height)$) then
    return true
else
    return false
end
```

Algorithm 10: IsVerifyRound.

```
Input: $A_{user}$
Output: bool
s = search_idx.find($A_{user}$)
rlt = result_idx.find($A_{user}$)
if s.SerialNum == rlt.SerialNum && rlt.$CI_{Result}$! = null then
    return true
else
    return false
end
```

Algorithm 11: IsResultReady.

```
Input: $A_{user}$
Output: void
$s$=$search\_idx$.find($A_{user}$)
send action (eosio.token, transfer, Auth, (_self, $A_{user}$, $s.Coin+d$))
$s.Coin$=0
$search\_idx$.modify($A_{user}$)
$Deposit = Deposit - d$
```

Algorithm 12: Compensate.

submitting keywords that are different from that in the search request. It will be calculated as follows:

$$\begin{aligned} \mathrm{HKW}' &= \mathrm{SHA256}\left(H_2\left(W_1{}'\right)\|, \cdots, \|H_2\left(W_m{}'\right)\right) \\ \mathrm{COMM}' &= \mathrm{SHA256}\left(\pi\|\mathrm{HKW}'\right) \end{aligned} \tag{27}$$

Blockchain determines whether COMM$'$ and COMM in the contract are equal. If they are the same, the calculation will continue as follows:

$$g^{a\pi} g^{b\pi \sum_{j=1}^{m'} H_2\left(W_j\right)} = \mathrm{tok}_1 \tag{28}$$

If the above equation is tenable, this DU is honest. After that, compute the following formula to verify the existence of the search result.

$$\mathrm{BFVerify}\left(\mathrm{BF}, \mathrm{HKW}', \left\{h_1{}', \cdots, h_k{}'\right\}\right) \longrightarrow (0, 1). \tag{29}$$

(b) Integrity

When DU believes that the result returned by SESP is not complete or corrupted, he can send the following transaction to the blockchain.

$$\begin{gathered} \left(\mathrm{Ref}_{\mathrm{block}}, t, \mathrm{Sig}_{\mathrm{user}}(\mathrm{Chain\_ID}, \mathrm{Tx}),\right. \\ \left.\mathrm{Action}\left(\mathrm{SEContract}, \mathrm{IntegrityVerify}, \mathrm{Auth}_{\mathrm{user}}, \left(A_{\mathrm{user}}\right)\right)\right). \end{gathered} \tag{30}$$

(c) Correctness

When DU believes that the result returned by SESP does not include the keywords searched for, he can send the following transaction:

$$\begin{gathered}(\mathrm{Ref}_{\mathrm{block}}, t, \mathrm{Sig}_{\mathrm{user}}(\mathrm{Chain\_ID}, \mathrm{Tx}),\\ \mathrm{Action}(\mathrm{SEContract}, \mathrm{CorrectnessVerify}, \mathrm{Auth}_{\mathrm{user}}, (A_{\mathrm{user}})).\end{gathered} \tag{31}$$

6. Security and Performance Analysis of Proposed Scheme

6.1. Security and Privacy Analysis of BFR-SE

6.1.1. Security and Privacy of ABSE. The ABSE model we constructed in BFR-SE is inspired by the attribute-based search model VABKS of Zheng et al. [21]. Furthermore, we have expanded it to support the multikeyword search. VABKS is proved to be secure, and the complete proof process can refer to the security analysis in [21]. The security of VABKS relies on the decisional linear assumption.

We focus on the fairness and reliability of searchable encryption, and the security of ABSE is not the main work of this paper. So, we will mainly analyze the correctness of ABSE and briefly discuss its security and privacy.

(1) Correctness. Let μ_i be the recovery coefficient of the ith row in $A_{l\times k}$ and calculate as follows:

$$E=\prod_{i\in\omega}(e(C_{2,i},\mathrm{tok}_4)e(C_{3,i},H_i))^{\mu_i}=\prod_{i\in\omega}e(g,g)^{\sigma_i\pi t\mu_i}=e(g,g)^{\pi t\sum_{i\in\omega}\mu_i\sigma_i}. \tag{32}$$

If the attribute set ω of DU satisfies the access policy ($A_{l\times k}, \rho$), it will get s by calculating $\sum_{i\in\omega}\mu_i\sigma_i$, and $E=e(g,g)^{\pi ts}$.

Then, calculate as follows:

$$\begin{gathered}e\left(\prod_{j=1}^{m}C_{4,j},\mathrm{tok}_2\right)=e(g,g)^{arc\pi+asc\pi+brc\pi\sum_{j=1}^{m}H_2\left(W_j\right)},\\ e(C_0,\mathrm{tok}_1)=e(g,g)^{cra\pi+crb\pi\sum_{j=1}^{m}H_2\left(W_j'\right)},\\ e(\mathrm{tok}_3,C_1)=e(g,g)^{ac\pi s-t\pi s}.\end{gathered} \tag{33}$$

If the keywords set KW' in the search token is the same as the keywords set KW in the index, it will have

$$e(C_0,\mathrm{tok}_1)e(\mathrm{tok}_3,C_1)E==e\left(\prod_{j=1}^{m}C_{4,j},\mathrm{tok}_2\right). \tag{34}$$

(2) Security and Privacy. From a security point of view, all attribute-based cryptographic algorithms need to resist collusion attacks. In the ABSE used in BFR-SE, we pick a random number t for each DU at the key generation phase, and the attribute-based part of the private key $\{K_{3,i}\}_{i\in\omega}$ is related to it. So, different DUs cannot combine their respective attributes to launch a collision attack.

From the perspective of privacy, DO encrypts his indices and stores them on SESP without revealing any information. Moreover, DO has fine-grained access control on the search function. For DU, a random number π is used every time generating a search token, making the search token generated will not be the same even if DU searches for the same keywords multiple times. The adversary cannot analyze DUs' privacy by collecting the traces of search requests.

6.1.2. Fairness and Reliability of BFR-SE

(1) Fairness. In this paper, we proposed a pay-per-use searchable encryption scheme. We believe that both SESP and DU are not always credible, and the dishonest behavior of either party may cause economic disputes. In our scheme, the participant with substantial computing power needs to pay a certain amount of deposit before becoming SESP. We divide each search of DU into two rounds: search round and verification round. When a DU initiates a search request, he needs to transfer the fee to smart contract.

DUs can initiate a request for arbitration in the verification round when SESP does not return any result, return partial results, or return incorrect results. The blockchain will arbitrate the search results. If it determines that SESP has acted dishonestly, SESP will be subject to a financial penalty, and the fine will compensate DU. Although DU may expose a little bit of their information when applying for arbitration, they will get financial compensation.

For SESP, if he can provide the correct results before the pre-agreed time, he can take away his profits after the freezing period. When DU initiates a search request, he promises the set of keywords retrieved and stores it on-chain, ensuring that DU will not submit a different set of keywords during the verification round to defraud the SESP deposit. Moreover, a particular fine can be introduced to DU so that DU cannot apply for arbitration without any certainty.

In summary, BFR-SE is fair to both SESP and DU.

(2) Reliability. In our scheme, we draw on the idea of verifiable searchable encryption in which DU can verify the results return by SESP from three aspects, including existence, integrity, and reliability. Therefore, the reliability of verifiable searchable encryption is also available in our scheme. Unlike the verifiable searchable encryption, we utilize the blockchain to introduce a reward and punishment mechanism for the scheme. When DU finds any problem with the results returned by SESP, they apply for arbitration during the verification round. If the SESP is indeed dishonest, the blockchain will punish SESP financially and compensate DU.

Table 2: Functional comparison between BFR-SE and other related searchable encryption schemes.

	BFR-SE	Ref. [20]	Ref. [21]	Ref. [29]	Ref. [30]
Fairness	Yes	No	No	Yes	Yes
Reliability	Strong	Weak	Weak	No	No
Privacy protection	Strong	Strong	Strong	Weak	Weak
Multikeyword search	Yes	Yes	No	No	Yes
Suitable for multiusers	Yes	No	Yes	Yes	Yes
Fine-grained access control	Yes	No	Yes	No	No
Practicability	Yes	Yes	Yes	No	No

Therefore, BFR-SE is more reliable than previous schemes.

6.1.3. Verifiability of Search Results

(1) Existence. BFR-SE uses Bloom filter to verify the existence of the search result. DO stores all the information corresponding to the keyword set in the indices into the Bloom filter. If the result returned by SESP is null, which means that there is no matching data for the search request, DU could verify it using the keywords searched for and Bloom Filter. If the verification result is 1, the keywords searched for having a high probability of existence. Although there is a specific false-positive rate, the research in Ref. [37] shows that the calculation method of this false-positive rate is as follows:

$$\left(1-\left(1-\frac{1}{l_{BF}}\right)^{kn}\right)^{k}\approx\left(1-e^{-kn/l_{BF}}\right)^{k}. \tag{35}$$

It can be seen that by setting the values of l_{BF} and k, the false-positive rate can be reduced. For example, when $k=(\ln 2)l_{\mathrm{BF}}/n$, it can get a minimum false-positive rate of $(0.6185)^{l_{BF}/n}$. Therefore, DUs can verify the existence of search results in BFR-SE.

(2) Integrity. For each row of indices in our scheme, DO uses his private key to sign the keywords and the MerkleRoot obtained from the data-related information as leaf nodes. Each row of the indices uploaded by DO contains the signature. DUs can use the public key of DO to verify whether the keywords and MerkleRoot are damaged. For the returned search results, DU can verify the data's integrity according to whether the MerkleRoot can be constructed. Once the data-related information as leaf nodes are destroyed, or the SESP only returns partial results, the MerkleRoot cannot be constructed. Therefore, DUs can verify the integrity of search results in BFR-SE.

(3) Correctness. If the existence and integrity are verified, then DU can get the ciphertext of the keyword set of the search results. DU only needs to use this ciphertext and his search token as inputs and repeatedly execute the TEST function to verify the search results. Therefore, DUs can verify the correctness of search results in BFR-SE.

6.2. Security and Privacy Analysis of BFR-SE

6.2.1. Functional Comparison. We compared BFR-SE with previous verifiable searchable encryption and blockchain-based searchable encryption schemes from the following aspects: fairness, reliability, privacy protection, whether it supports multikeyword search, whether it is suitable for multi-user situation, whether it supports fine-grained access control for the search function, and practicability.

From the comparison in Table 2, comparing verifiable searchable encryption and previous blockchain-based searchable encryption schemes, the following conclusions can be drawn:

(1) The former and earlier related studies did not consider the fairness of searchable encryption. With the emergence of blockchain, these blockchain-based schemes all meet the requirements of fairness

(2) The former supports DUs to verify search results and has a certain degree of reliability, but because there is no subsequent sufficient punishment, the reliability is weak. However, the recent searchable encryption schemes based on blockchain have not been considered reliable

(3) The former stores the indices and DU's search records on SESP, which can better protect DU's privacy on the premise that SESP is credible. The latter stores the indices and search records on blockchain and uses a deterministic encryption algorithm. Since the information on-chain is public, even if the keywords are encrypted, when the search records are large enough, it is not impossible to analyze some privacy of DU

(4) The performance of the former depends on the capabilities of SESP and has high practicability. The latter is mostly based on low-performance blockchain platforms such as Bitcoin and Ethereum, and the design is not particularly perfect, so there are still problems in performance and security. In real enterprise applications, it does not have practicality

Compared with these two types of schemes, BFR-SE designs a relatively perfect reward and punishment mechanism with blockchain. If SESP is dishonest, it will pay the price. Therefore, our scheme has both fairness and reliability. The indices are still stored on SESP in our scheme, and the blockchain only plays the role of arbitration when there are disputes, which makes BFR-SE more efficient than the previous blockchain-based schemes. Our constructed ABSE is not a deterministic encryption algorithm. The random number makes the search token different, even for the same keyword set. Therefore, BFR-SE has strong privacy protection capabilities. We have extended the work of Ref. [21] to support a multikeyword search. The combination of attribute-based searchable encryption and blockchain makes BFR-SE meet multiuser scenarios in a distributed environment quickly and enables DO to have fine-grained access control on his sharing data. BFR-SE uses EOS blockchain, which is the current high-performance public chain. Moreover, it considers more practical scenarios, such as SESP and DU may be dishonest. So, our scheme has better practicality.

6.2.2. Storage Analysis. BFR-SE is a fair and reliable searchable encryption scheme based on the EOS blockchain. Since the storage resource on the blockchain is very valuable, and the acquisition of RAM in the EOS blockchain requires the user to mortgage the system token, it is necessary to analyze the size of the data stored on-chain.

In the beginning, we define some symbols. We set $|G_0|$, $|G_1|$ to represent the bit length of an element in group G_0 and G_1, respectively. Let $|Z_p|$ be the bit length of an element in field Z_p, $|Sk_{\text{sig}}|$ be the bit length of the signature, $|S|$ be the number of all attributes, $|U|$ be the number of DUs, |TOK| be the bit length of search token, |COMM| be the bit length of commitment, $|\text{CI}_{\text{result}}|$ be the bit length of the result returned by SESP. Let m be the size of the keyword set.

According to the experiment simulation in our scheme, we set $|G_0| = |G_1| = 1024$ bits, $|Z_p| = 128$ bits, $|Sk_{\text{sig}}| = 576$ bits, $|Sk_{\text{com}}| = |Sk_{\text{sig}}| = 256$ bits $|Pk_{\text{com}}| = |Pk_{\text{sig}}| = 272$ bits. The length of *Account*, *SerialNum*, *blockheight*, *Deposit*, and *Coin* are all 64 bits. The implementation of our Bloom filter refers to the C++ code on Github, which address is https://github.com/bbondy/bloom-filter-cpp.git. We set the length of the bloom filter to 20 KB and there are 5000 rows of ciphertext indices, so $l_{\text{BF}}/n = (20 \times 8 \times 1024)/5000 = 32.768$ and the number of hash functions in Bloom Filter will be $k = (\ln 2) l_{\text{BF}}/n = 22$. Then, the false-positive will be 1.45×10^{-7}. In our scheme, three operations that will interact with blockchain to store data in the smart contract, which are as follows:

(1) Initialization

DO uploads the public system parameters and his public key for the signature to smart contract. The storage cost is as follows:

$$3|G| + |G||S| + |Pk_{\text{Sig}}|. \tag{36}$$

(2) Registration

The information on-chain mainly includes registration-related information uploaded by DU and SESP. The DU registration includes the information submitted by DU and the private key distributed by DO. The SESP registration includes the account and the deposit. The specific storage overhead is as follows:

$$(|\text{account}| + |Pk_{\text{com}}| + 2|G| + |S||G|)|U| + (|\text{account}| + |\text{Deposit}|). \tag{37}$$

(3) Search

The information on-chain mainly includes auxiliary information uploaded by DO, search requests initiated by DUs, and search results returned by SESP. The specific storage overhead is as follows:

$$\begin{aligned} l_{\text{BF}} &+ (|\text{account}| + |\text{SerialNum}| + |\text{TOK}| + |\text{COMM}| + |\text{height}| + |\text{coin}|)|\text{U}| \\ &+ (|\text{account}| + |\text{SerialNum}| + |\text{CI}_{\text{result}}|)|U|. \end{aligned} \tag{38}$$

The storage overhead of BFR-SE varies with the number of attributes is shown in Figure 4.

For simplicity, the figure only shows that the storage overhead varies with the number of attributes when there are only 10 DUs and 50 keywords. From the figure, we can see that the storage overhead is mainly spent in the search phase, while the initialization and registration phase are negligible. As the number of DUs and keywords grows, storage overhead will also increase linearly. However, since we only store some information necessary for verification on-chain, compared with other blockchain-based schemes [24, 26–30] that store all index information on-chain, the storage overhead is reduced a lot.

Users can obtain the RAM in EOS by collateralizing system tokens, and the current price is 42EOS/MB. DO can purchase RAM according to the scale of his system. Unlike Ethereum transactions that need to consume ETH as gas, the tokens mortgaged when acquiring RAM in EOS can still be redeemed at the original price. Above all, BFR-SE is feasible and practical.

6.2.3. Performance Analysis. Before analyzing the performance, we define two primary operations' computational cost: P for bilinear pairing and E for power exponent operation. Here we ignore the computational overhead of operations such as hash functions because they are very efficient than the above two. In our proposed scheme, the computational overhead of the primary operations is shown in Table 3.

There are many studies on the analysis mentioned above [38–43], so we will not repeat them too much. Like storage resource, the computing resource on-chain is also very valuable. If the interaction with the blockchain is too frequent or the computational overhead is too large, it will have a terrible impact on system performance. So, we mainly focus on the execution time of BFR-SE on-chain.

We used 6 nodes to build an EOS private chain in a laboratory environment. The 6 nodes we chose were all

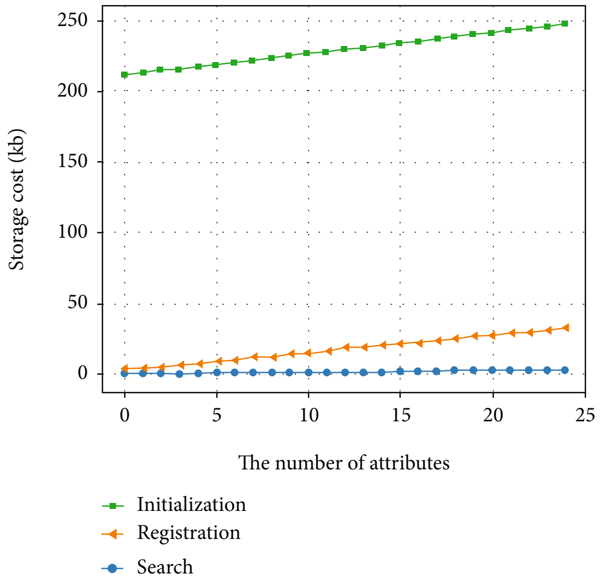


Figure 4: The storage overhead of BFR-SE varies with the number of attributes.

Table 3: The computational overhead of main operations in BFR-SE.

Operation	Computational overhead
Initialization	$(3+\|S\|)E$
Key generation	$(2+\|S\|)E$
Keyword encryption	$(2+3\|S\|+m)E$
Search token generation	$(4+\|S\|+m)E$
Search	$4P$

MacBook Pro (2017) with Intel (R) Core (TM) i5 CPU that clocks at 3.1 GHz and has 8 GB of RAM. The version of the EOS blockchain we chose is v2.0.7. The computational overhead of other blockchain-based schemes [24, 26, 27, 30] is all second level, which is obviously not better than ours, and will not be analyzed anymore. We compared BFR-SE with the scheme in Ref. [25], as shown in Figure 5.

In BFR-SE, most of the interactions with the blockchain are to upload data to smart contract, such as initialization and registration. The computational overhead of this part can be ignored. The main computational overhead of BFR-SE occurs when the blockchain arbitrates. It can be seen from the figure that as the number of indices to increase, the computational overhead on-chain of BFR-SE has always remained at a stable level, about 40 ms, while the scheme in Ref. [25] will grow. It is because, in our scheme, all the time-consuming operations are executed off-chain. The EOS block producers' configuration in the Mainnet is much higher than that of the MacBook used in our simulation environment. When our contracts are deployed on the Mainnet, the performance will be even more outstanding. The EOS blockchain generates a block in 0.5 seconds, and the transaction will be confirmed soon after execution. Therefore, compared with Ref. [25], BFR-SE has a better performance.

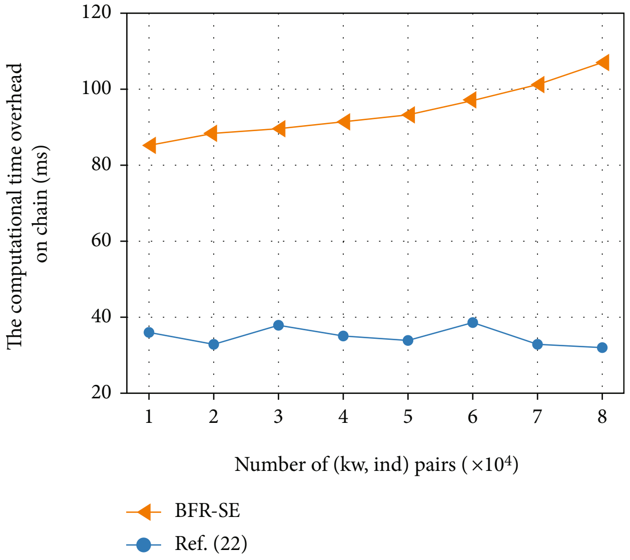


Figure 5: The computational overhead of BFR-SE and Ref. [25] varies with the number of indices.

7. Conclusion

To achieve a fair and reliable searchable encryption scheme, we constructed an attribute-based searchable encryption ABSE that supports multiple keywords search and designed an exclusive reward and punishment mechanism by using blockchain. In our scheme, DO sends the ciphertext of indices to SESP and uploads the auxiliary information to the blockchain. SESP must return the correct search results before a preagreed block height, and the charge fee paid from DU will be frozen for a period during which DU could initiate an arbitration request to the blockchain if he disagrees with the results. As the cornerstone of trust, blockchain will punish the dishonest party economically, ensuring the scheme's absolute fairness and reliability [44–53]. Besides, ABSE can be used by DO to have fine-grained access control on the search function. Experiments and analyses show that our scheme is feasible and has better performance. However, our scheme still has many shortcomings. For example, our scheme uses an index structure, and the signature guarantees the integrity of the indices, but this significantly reduces the flexibility of the scheme, especially when adding or updating the index of sharing data. Simultaneously, due to an attribute-based encryption algorithm, topics such as the revocation or update of permission are also one of the directions that need to be studied in the future. We will continue to refine our approach in conjunction with some other research [37, 54–57].

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Fundamental of China under Grants 61272519, 61170297, 61472258, and 61802094, China Postdoctoral Science Foundation under Grant 2021M690305, and National Natural Science Foundation of Zhejiang Province under Grant LY20F020012.

References

[1] Y. Xu, X. Yan, Y. Wu, Y. Hu, W. Liang, and J. Zhang, "Hierarchical bidirectional RNN for safety-enhanced B5G heterogeneous networks," *IEEE Transactions on Network Science and Engineering*, 2021.

[2] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing.*, vol. 15, no. 4, pp. 577–590, 2018.

[3] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[4] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *The 39th IEEE International Conference on Distributed Computing Systems*, pp. 144–153, 2019.

[5] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[6] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.

[7] X. Zhou, X. Yang, J. Ma, and K. Wang, "Energy efficient smart routing based on link correlation mining for wireless edge computing in IoT," *IEEE Internet of Things Journal*, 2021.

[8] X. Yan, J. Zhang, H. Elahi, M. Jiang, and H. Gao, "A personalized search query generating method for safety-enhanced vehicle-to-people networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5296–5307, 2021.

[9] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceeding 2000 IEEE Symposium on Security and Privacy. S&P*, pp. 44–55, 2000.

[10] B. Dan, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques*, Interlaken, Switzerland, May 2-6, 2004.

[11] M. Abdalla, M. Bellare, D. Catalano et al., "Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions," in *CRYPTO'05: Proceedings of the 25th annual international conference on Advances in Cryptology*, pp. 205–222, Berlin, Heidelberg, 2005.

[12] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222–233, 2014.

[13] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Transactions on Parallel & Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2016.

[14] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multikeyword fuzzy search over encrypted data in the cloud," in *IEEE INFOCOM 2014- IEEE Conference on Computer Communications*, pp. 2112–2120, 2014.

[15] S. Tahir, S. Ruj, Y. Rahulamathavan, M. Rajarajan, and C. Glackin, "A new secure and lightweight searchable encryption scheme over encrypted cloud data," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 4, pp. 530–544, 2019.

[16] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, Decentralized Business Review, 2008.

[17] Q. Chai and G. Gong, "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," in *2012 IEEE International Conference on Communications (ICC)*, pp. 917–922, 2012.

[18] K. Kurosawa and Y. Ohtaki, "Uc-secure searchable symmetric encryption," in *FC 2012: Financial Cryptography and Data Security*, vol. 7397, pp. 285–298, 2012.

[19] X. Zhu, Q. Liu, and G. Wang, "Verifiable dynamic fuzzy search over encrypted data in cloud computing," in *The 15th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2015)*, vol. 9530, pp. 655–666, 2015.

[20] M. Azraoui, K. Elkhiyaoui, M. Onen, and R. Molva, "Publicly verifiable conjunctive keyword search in outsourced databases," in *2015 IEEE Conference on Communications and Network Security (CNS)*, pp. 619–627, 2015.

[21] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: verifiable attribute-based keyword search over outsourced encrypted data," in *IEEE INFOCOM 2014- IEEE Conference on Computer Communications*, pp. 522–530, 2014.

[22] M. H. Ameri, M. R. Asaar, J. Mohajeri, and M. Salmasizadeh, *A generic construction for verifiable attribute-based keyword search schemes*, IACR Cryptology ePrint Archive, 2015.

[23] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1187–1198, 2016.

[24] H. Li, F. Zhang, J. He, and H. Tian, "A searchable symmetric encryption scheme using blockchain," 2017, https://arxiv.org/abs/1711.01030.

[25] H. Li, H. Tian, F. Zhang, and J. He, "Blockchain-based searchable symmetric encryption scheme," *Computers & Electrical Engineering*, vol. 73, pp. 32–45, 2019.

[26] S. Hu, C. Cai, Q. Wang, C. Wang, X. Luo, and K. Ren, "Searching an encrypted cloud meets blockchain: a decentralized, reliable and fair realization," in *IEEE INFOCOM 2018- IEEE Conference on computer Communications*, pp. 792–800, 2018.

[27] C. Cai, J. Weng, X. Yuan, and C. Wang, "Enabling reliable keyword search in encrypted decentralized storage with fairness," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 1, pp. 131–144, 2021.

[28] Q. Tang, "Towards blockchain-enabled searchable encryption," in *Information and Communications Security*, pp. 482–500, Springer International Publishing, Cham, 2020.

[29] L. Chen, W. K. Lee, C. C. Chang, K. K. R. Choo, and N. Zhang, "Blockchain based searchable encryption for electronic health

record sharing," *Future Generation Computer Systems*, vol. 95, pp. 420–429, 2019.

[30] S. Jiang, J. Cao, J. A. McCann et al., "Privacy-preserving and efficient multi-keyword search over encrypted data on blockchain," in *2019 IEEE International Conference on Blockchain (Blockchain)*, pp. 405–410, 2019.

[31] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[32] V. Buterin, *Ethereum: a next-generation smart contract and decentralized application platform*, white paper, 2013.

[33] N. Szabo, "Smart contracts," 1994, http://szabo.best.vwh.net/smart.contracts.idea.html.

[34] D. Larimer, "Eos.io technical white paper," 2017, https://steemit.com/eos/@eosio/eos-io-technical-white-paper.

[35] H. Gao, Z. Ma, S. Luo, and Z. Wang, "BFR-MPC: a blockchain-based fair and robust multi-party computation scheme," *IEEE Access*, vol. 7, pp. 110439–110450, 2019.

[36] H. Gao, Z. Ma, S. Luo, Y. Xu, and Z. Wu, "BSSPD: a blockchain-based security sharing scheme for personal data with fine- grained access control," *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, Article ID 6658920, 20 pages, 2021.

[37] J. Zhang, Y. Yan, Z. Cheng, and W. Wang, "Lightweight attention pyramid network for object detection and instance segmentation," *Applied Sciences*, vol. 10, no. 3, pp. 883–899, 2020.

[38] X. Zhou, X. Xu, W. Liang et al., "Intelligent small object detection based on digital twinning for smart manufacturing in industrial CPS," *IEEE Transactions on Industrial Informatics*, 2022.

[39] Y. Xu, Z. Liu, C. Zhang, J. Ren, Y. Zhang, and X. Shen, "Blockchain-based trustworthy energy dispatching approach for high renewable energy penetrated power systems," *IEEE Internet of Things Journal*, 2021.

[40] X. Zhou, Y. Li, and W. Liang, "CNN-RNN based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, 2021.

[41] X. Yan, Y. Xu, B. Cui, S. Zhang, T. Guo, and C. Li, "Learning URL embedding for malicious website detection," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6673–6681, 2020.

[42] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, "Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5790–5798, 2021.

[43] X. Yan, B. Cui, Y. Xu, P. Shi, and Z. Wang, "A method of information protection for collaborative deep learning under GAN model attack," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 871–881, 2021.

[44] Y. Xu, C. Zhang, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, "Blockchain-enabled accountability mechanism against information leakage in vertical industry services," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1202–1213, 2021.

[45] C. Zhang, Y. Xu, Y. Hu, J. Wu, J. Ren, and Y. Zhang, "A blockchain-based multi-cloud storage data auditing scheme to locate faults," *IEEE Transactions on Cloud Computing*, 2021.

[46] Y. Xu, C. Zhang, G. Wang, Z. Qin, and Q. Zeng, "A blockchain-enabled deduplicatable data auditing mechanism for network storage services," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1421–1432, 2021.

[47] Y. Xu, Q. Zeng, G. Wang, C. Zhang, J. Ren, and Y. Zhang, "An efficient privacy-enhanced attribute-based access control mechanism," *Concurrency & Computation: Practice & Experience*, vol. 32, no. 5, pp. 1–10, 2020.

[48] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, "Deep-learning-enhanced multitarget detection for end-edge-cloud surveillance in smart IoT," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12588–12596, 2021.

[49] C. Zhang, Z. Ni, Y. Xu, E. Luo, L. Chen, and Y. Zhang, "A trustworthy industrial data management scheme based on redactable blockchain," *Journal of Parallel and Distributed Computing*, vol. 152, pp. 167–176, 2021.

[50] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, "Blockchain empowered arbitrable data auditing scheme for network storage as a service," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 289–300, 2020.

[51] J. Zhang, M. Z. A. Bhuiyan, X. Yang et al., "AntiConcealer: reliable detection of adversary concealed behaviors in EdgeAI assisted IoT," *IEEE Internet of Things Journal*, 2021.

[52] J. Zhang, M. Z. A. Bhuiyan, Y. Xu, A. K. Singh, D. F. Hsu, and E. Luo, "Trustworthy target tracking with collaborative deep reinforcement learning in EdgeAI-aided IoT," *IEEE Transactions on Industrial Informatics*, 2022.

[53] G. Yu, X. Zha, X. Wang et al., "Enabling attribute revocation for fine-grained access control in blockchain-IoT systems," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1213–1230, 2020.

[54] X. Yan, Y. Xu, X. Xing, B. Cui, Z. Guo, and T. Guo, "Trustworthy network anomaly detection based on an adaptive learning rate and momentum in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6182–6192, 2020.

[55] Y. Wang, T. Li, H. Qin et al., "A brief survey on secure multi-party computing in the presence of rational parties," *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 6, pp. 807–824, 2015.

[56] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2019.

[57] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proceedings of the 30th Annual International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology, ser. EUROCRYPT'11*, pp. 568–588, Berlin, Heidelberg, 2011.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 4329219, 13 pages
https://doi.org/10.1155/2021/4329219

Research Article

Anomaly Detection Collaborating Adaptive CEEMDAN Feature Exploitation with Intelligent Optimizing Classification for IIoT Sparse Data

Jianming Zhao,[1,2,3,4] Peng Zeng,[1,2,3,4] Ming Wan,[5] Xinlu Xu,[5] Jinfang Li,[5] and Qimei Jiang[6]

[1] *State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China*
[2] *Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang 110016, China*
[3] *Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China*
[4] *University of Chinese Academy of Sciences, Beijing 100049, China*
[5] *School of Information, Liaoning University, Shenyang 110036, China*
[6] *AVIC Changhe Aircraft Industry (Group) Corporation Ltd., Jingdezhen 333002, China*

Correspondence should be addressed to Ming Wan; wanming@lnu.edu.cn

Received 25 July 2021; Revised 4 September 2021; Accepted 6 September 2021; Published 7 October 2021

Academic Editor: Yan Huo

IIoT (Industrial Internet of Things) has gained considerable attention and has been increasingly applied due to its ubiquitous sensing and communication. However, the sparse characteristic of sensing data in distributed IIoT networks may bring out tremendous challenges to implement the security protection measures. Based on the design of centralized data gathering and forwarding, this paper proposes a novel anomaly detection approach for IIoT sparse data, which can successfully collaborate the adaptive CEEMDAN (Complete Ensemble Empirical Mode Decomposition with Adaptive Noise) feature exploitation with one intelligent optimizing classification. Furthermore, in the adaptive CEEMDAN feature exploitation, the CEEMDAN energy entropy based on adaptive IMF (Intrinsic Mode Function) selection is designed to extract the sensing features from IIoT sparse data; in the intelligent optimizing classification, one effective OCSVM (One-Class Support Vector Machine) classifier optimized by the IABC (Improved Artificial Bee Colony) swarm intelligence algorithm is introduced to detect various abnormal sensing features. The experimental results show that, not only does the CEEMDAN energy entropy based on adaptive IMF selection accurately describe the change of industrial production by analyzing the probability distribution and energy distribution of sparse sensing data, but also the proposed IABC-OCSVM classifier has higher detection efficiency compared with the OCSVM classifiers optimized by other swarm intelligence algorithms.

1. Introduction

IIoT (Industrial Internet of Things), which can effectively implement real-time simulation and remote control during the whole production or manufacturing cycle, has been regarded as an important driving force in the industrial intelligent revolution [1]. Furthermore, IIoT can successfully establish one seamless connection between OT (Operational Technology) and IT (Information Technology), and the application of various IIoT devices (such as sensors, collectors, or controllers) can cover most aspects of industrial production by using some advanced technologies [2, 3], including sensing technology, wireless interconnection and communication technology, and intelligent analysis technology. Under the integration of distributed monitoring and centralized management, IIoT can accomplish the data processing of various industrial activities in a more efficient way. Consequently, it can not only improve the production quality and efficiency enormously, but it can also reduce the product cost and resource consumption significantly.

Actually, the original definition of IIoT is characterized by the high interconnectivity and large-scale distributed network, and various IIoT devices can be directly or indirectly exposed to the public Internet. However, information security problems in no matter what type of cyberphysical systems or social networks emerge rapidly and extensively, and the corresponding security incidents also occur repeatedly [4, 5]. As a consequence, IIoT is facing more and more severe challenges of information security [6–9], and it may suffer from greater risks than traditional IoT. In particular, one integrated IIoT system always consists of thousands of sensor nodes, which ensure interconnection and interoperability by using some specific wireless communication protocols. Once one or several sensor nodes have been maliciously infiltrated and controlled by some sophisticated adversaries, the corresponding disruptive activities may spread at a rapid rate due to the through-hull connection of all sensor nodes and have tremendous implications on the whole system [10]. According to the basic flow and interaction of sensing data, IIoT network architecture can be briefly divided into three layers: data acquisition layer, data transmission layer, and data processing layer, and each layer can experience various degrees of security threats due to distinct technology solutions. For instance, in the data acquisition layer, some intrinsic system flaws of IIoT devices may be considered as the most direct invasion targets to inject malicious codes [11]; in the data transmission layer, the public wireless communication protocols and distributed network structure may become some of the weak points, which can be stealthily exploited to perform data-stealing or data-tampering attacks, such as Sybil attacks or arbitrage attacks [12, 13]; in the data processing layer, various local or remote servers are always exposed to the public network without some extra protection measures, and these servers can be potentially targeted by malicious adversaries who can easily excavate more attack entries and paths [14, 15].

In order to guarantee the stability and reliability of IIoT systems, both academia and industry have carried out many theoretical researches and practical applications on IIoT security protection measures: for the data privacy challenge, the work in [16] discusses and summarizes the main issues in the traditional IIoT architecture and designs the detailed data interaction process based on the blockchain architecture to enhance security and privacy in smart factories; for the data authenticity challenge, the work in [17] proposes a robust certificateless signature scheme for data crowdsensing in the cloud-assisted IIoT, which can be proven to effectively deal with four types of signature forgery attacks; for the data confidentiality challenge, the work in [18] presents a secure industrial data access control scheme for cloud-assisted IIoT, and it uses the ciphertext policy-attribute-based encryption scheme to provide fine-grained data protection; for the malicious data transmission challenge, the work in [12] introduces a secured and intelligent communication scheme for PES (Pervasive Edge Computing) in an IIoT-enabled infrastructure, and it proposes a lightweight Sybil attack detection protocol to protect low-powered IIoT devices; for the data congestion challenge, by using an average consensus-based algorithm, the work in [19] puts forward an optimal scheduling framework to resist a DoS attack for IIoT-based smart microgrids. In the above protection measures, some additional security functions or schemes are designed to improve the security of original IIoT systems. Although they can reflect an enhanced level of security capability due to the fine theoretical and experimental analysis, their applicability and feasibility in real-world IIoT systems await verification by future explorations. The main causes involve the following two aspects: on the one hand, most IIoT devices only have low power and limited system resources, and the security add-ons may decrease their work performance by performing the higher or lower computational costs of security operations [20]; on the other hand, IIoT is usually designed to serve industrial control systems, whose requirements on high availability and reliability may be not completely satisfied because of the inefficient adaption between the original system design and some added security services. Differently, anomaly detection in IIoT systems can be widely regarded as a feasible and effective measure to identify unexpected industrial activities [8, 21–23], because it can scarcely affect industrial availabilities and real-time requirements by using the bypass monitoring. However, the sensing data in distributed IIoT networks has some special characteristics of sparsity, statefulness, and correlation. In practice, the sparsity of sensing data may bring out tremendous challenges to implement the global anomaly detection, because the extracted spatial features seem to be unfavourable for a full-scale anomaly detection model without establishing the intrinsic relationship between different sparse sensing data. In order to solve the above problem, one ideal method is to collect and analyze all sparse sensing data in a local wireless sensor network, which is mainly applied to complete one technological process in the whole industrial production or manufacturing. Additionally, based on the relatively short-range communication characteristic, most IIoT systems always utilize the data collector to gather and forward the sensing data from distributed IIoT sensors, and this design can contribute to developing an experienced machine-learning anomaly detection model, which can thoroughly explore the statefulness and correlation characteristics of sparse sensing data. From this point of view, this paper proposes a novel anomaly detection approach for IIoT sparse data, and this approach successfully collaborates the adaptive CEEMDAN (Complete Ensemble Empirical Mode Decomposition with Adaptive Noise) feature exploitation with one intelligent optimizing classification. Moreover, the CEEMDAN energy entropy based on adaptive IMF (Intrinsic Mode Function) selection is designed to extract the sensing features from IIoT sparse data, and one effective OCSVM (One-Class Support Vector Machine) classifier optimized by the IABC (Improved Artificial Bee Colony) swarm intelligence algorithm is introduced to detect various abnormal sensing features. Additionally, we use some real-world data captured from one local oilfield IIoT system in the northeastern part of China to evaluate our approach, and the experimental results show that, for one thing, compared with the traditional CEEMDAN singular spectrum entropy and EEMD singular value decomposition, the CEEMDAN energy entropy based on adaptive IMF selection can accurately describe the change of sparse sensing data and is more sensitive to the size of abnormal data; for another, compared with the OCSVM classifiers optimized by other swarm intelligence algorithms, the proposed IABC-OCSVM classifier has higher detection efficiency.

2. Adaptive CEEMDAN Feature Exploitation

2.1. Preparation. Collect all IIoT sensing data in time interval T ($T = \sum_{i=1}^{m} t_i$, $\forall i \in [1, m]$), and extract the corresponding data sequence $D_{\mathrm{i}} = d_1^i d_2^i d_3^i \cdots d_n^i, \forall i \in [1, m]$ from the IIoT sensing data in each time interval t_i ($i \in [1, m]$), where d_n^i represents the nth data value in the data sequence D_i. After that, all data sequences D_i ($i \in [1, m]$) form a data sequence set $D = \{D_1, D_2, D_3, \cdots, D_m\}$, where m is the number of data sequences in the set D.

Due to the different number of IIoT sensing data in each time interval t_i, the dimensions of all data sequences D_i ($i \in [1, m]$) are distinct from one another. In order to reconstruct new data samples with the same dimension, an adaptive CEEMDAN feature exploitation method is properly proposed. Furthermore, this method first uses the CEEMDAN decomposition to perform the multiscale analysis on each data sequence, and then adaptively selects the effective IMF components as the feature factors [24]. After that, the corresponding energy entropies are calculated as the final feature values to reconstruct all data samples $Y_i = (y_1^i, y_2^i, y_3^i, \cdots, y_f^i)$ ($i \in [1, m]$), which have the same dimension. Here, y_j^i represents the jth feature variable in the ith data sample Y_i, and f is the dimension number of Y_i.

2.2. Adaptive IMF Selection. As mentioned above, in order to construct the data samples Y_i ($i \in [1, m]$) with the same dimension, it is necessary to determine the feature factors and calculate the corresponding feature values which can be further utilized to obtain the feature variable y_j^i. In terms of feature factor selection, although the IMF components can be used as the feature factors for some traditional anomaly detection models, there is still a considerable issue that the fixed parameter values cannot accurately describe the intrinsic characteristics of original data. To address this issue, the proposed feature exploitation method can sufficiently analyze the contribution of a single IMF component and the global reconstruction error, and adaptively adjust the number of effective IMF components according to the intrinsic characteristics. The specific selection process is listed as follows:

Step 1. We calculate the root mean square error (RMSE), correlation coefficient, and energy difference between the original data and the reconstructed data, and design the adjustment coefficient β to appropriately adjust the number of IMF components for the adaptive selection of effective IMF components.

Suppose x and x' represent the original data and the reconstructed data, respectively. The numerical difference between x and x' can be measured by the RMSE, which is defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(x_k - x_k'\right)^2}. \tag{1}$$

The correlation between x and x' can be measured by the correlation coefficient, which is defined as

$$r = \frac{\sum_{k=1}^{n}(x_k - \bar{x})\left(x_k' - \overline{x'}\right)}{\sqrt{\sum_{k=1}^{n}(x_k - \bar{x})^2}\sqrt{\sum_{k=1}^{n}\left(x_k' - \overline{x'}\right)}}. \tag{2}$$

Additionally, the energy difference between x and x' can be calculated by

$$\mathrm{Diff}\left(E(x), E\left(x'\right)\right) = \frac{1}{\left|E(x) - E\left(x'\right)\right|}. \tag{3}$$

Here, E represents the energy value calculation of the original data or the reconstructed data.

Based on the above parameters, we can define the final adjustment coefficient β as follows:

$$\beta = 1 - \frac{\mathrm{RMSE}}{r + \mathrm{diff}\left(E(x), E\left(x'\right)\right)}. \tag{4}$$

Step 2. We calculate the cumulative variance contribution and cumulative energy proportion of each IMF component and dynamically adjust their threshold parameters. After that, we further select the effective IMF components which are less than two threshold parameters. Two threshold parameters of IMF components can be calculated by

$$\begin{cases} T_\lambda = 1 - \dfrac{\sum_{j=1}^{m}\lambda_j}{\sum_{j=1}^{J}\lambda_j}, \\ T_E = 1 - \dfrac{\sum_{j=1}^{m}E_j}{\sum_{j=1}^{J}E_j}, \end{cases} \quad m \in [1, J], \tag{5}$$

Here, λ_j is the variance of the jth IMF component, and E_j is the energy of the jth IMF component.

In terms of feature value calculation, the following two points need to be emphasized: the first is the probability distribution of data sequence, and the second is the energy distribution of data sequence. In practice, the technological processes in industrial production can be mapped to industrial communication behaviours by analyzing industrial communications data [25]. That is, when industrial communication behaviours show different states or stages, the corresponding probability distribution and energy distribution of data sequences dynamically change. As a result, the probability distribution and energy distribution of each IMF component obtained by the CEEMDAN decomposition can also change when performing the multiscale analysis on the data sequences. In order to successfully track this change, this paper introduces the information entropy based on the energy distribution of IMF components, and takes the energy entropies of effective IMF components as the final feature values.

2.3. Feature Calculation Based on CEEMDAN Energy Entropy. As shown in Figure 1, the specific steps of feature calculation based on CEEMDAN energy entropy are listed as follows:

Step 1 (data preprocessing). As mentioned earlier, we obtain each data sequence $D_i = d_1^i d_2^i d_3^i \cdots d_n^i$, $\forall i \in [1, m]$ from the original IIoT sensing data, and we form the data sequence set $D = \{D_1, D_2, D_3, \cdots, D_m\}$.

Step 2 (IMF component calculation). Each data sequence D_i ($\forall i \in [1, m]$) is decomposed by the CEEMDAN decomposition to obtain J IMF components.

First of all, we suppose the original data $x = D_i$, and we carry out I different experiments on $x + \varepsilon_0 w_v$ by using the CEEMDAN decomposition. Additionally, the EMD decomposition in each experiment continues running until the first EMD modal component is obtained. From these I experiments, the first average IMF component can be further calculated by

$$\mathrm{imf}_1' = \frac{1}{I}\sum_{v=1}^{I} \mathrm{imf}_{v1}. \tag{6}$$

Here, imf_{v1} represents the first IMF component of the vth experiment.

Also, the first unique remainder can be obtained by

$$r_1 = x - \mathrm{imf}_1'. \tag{7}$$

Secondly, according to the above method, we further decompose $r_j + \varepsilon_j E_j(w_v)$, $v = 1, 2, \cdots, I$ for each j ($j = 1, 2, \cdots, J$), and we calculate the $(j+1)$-th IMF component by

$$\mathrm{imf}_{j+1}' = \frac{1}{I}\sum_{v=1}^{I} E_1\left(r_j + \varepsilon_j E_j(w_v)\right). \tag{8}$$

Also, the jth unique remainder can be obtained by

$$r_j = r_{j-1} - \mathrm{imf}_j', \quad j \in [2, J]. \tag{9}$$

Here, imf_j' is the jth IMF component obtained by the CEEMDAN decomposition, $E_j(\cdot)$ is the jth EMD modal component obtained by the EMD decomposition, ε_{j-1} is the SNR adjustment coefficient when adding the noise to solve imf_j', and w_v is an added zero mean white noise source for v experiments.

Finally, we repeat the above calculation process until no remainder can be decomposed, and we obtain all J IMF components $\mathrm{IMF} = \{\mathrm{imf}_1', \mathrm{imf}_2', \cdots, \mathrm{imf}_J'\}$. Also, the final remainder can be calculated by

$$R = x - \sum_{j=1}^{J} \mathrm{imf}_j'. \tag{10}$$

To sum up, the original data x can be finally decomposed into

$$x = \sum_{j=1}^{J} \mathrm{imf}_j' + R. \tag{11}$$

Step 3. According to the adaptive IMF selection process, we need to calculate the RMSE, correlation coefficient, and energy difference between the original data x and the reconstructed data x' which is reconstructed by the IMF components, and we also calculate the cumulative variance contribution and cumulative energy proportion of each IMF component. Through the adaptive IMF selection, we can obtain f effective IMF components.

Step 4. By further calculating the energy E_j ($\forall j \in [1, f]$) of each effective IMF component, we can construct the corresponding energy vector $V_E = (E_1, E_2, \cdots, E_f)$.

Step 5. For the energy vector V_E, the calculated energy entropy $H(E_j)$ ($\forall j \in [1, f]$) of each effective IMF component can be regarded as one feature value. Also, we can get the energy entropy vector $V_H = (H_1, H_2, \cdots, H_f)$. The energy entropy of each effective IMF component can be calculated by

$$H\left(E_j\right) = -P\left(E_j\right) \log P\left(E_j\right). \tag{12}$$

Here, E_j represents the energy value of the jth IMF component; $P(E_j) = E_j / \sum_{j=1}^{J} E_j$ is the energy proportion of the jth IMF component in the total energy.

Step 6. We set the data sample $Y_i = V_H = (H_1, H_2, \cdots, H_f)$ ($\forall i \in [1, m]$), and we form the final data sample set $Y = \{Y_1, Y_2, \cdots, Y_m\}$.

3. IABC-OCSVM Anomaly Detection Classifier

3.1. OCSVM Classifier. OCSVM [26, 27], which has a relatively fine classification effect and a generalization capability for small sample data, belongs to one improved version of traditional SVM (Support Vector Machine). Differently, OCSVM exploits the aggregation of original data in the high-dimensional feature space to find one optimal separating hyperplane, which keeps the maximum distance from the coordinate origin. In one sense, OCSVM only needs one class of samples to train a suitable classifier.

Actually, OCSVM is briefly designed to solve the following quadratic programming problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\omega\|^2 + \frac{1}{vl}\sum_{i=1}^{l} \xi_i - \rho \\ \text{s.t.} \quad & \Phi(x_i)\omega \geq \rho - \xi_i, \quad \xi_i \geq 0,\ i = 1 \cdots l. \end{aligned} \tag{13}$$

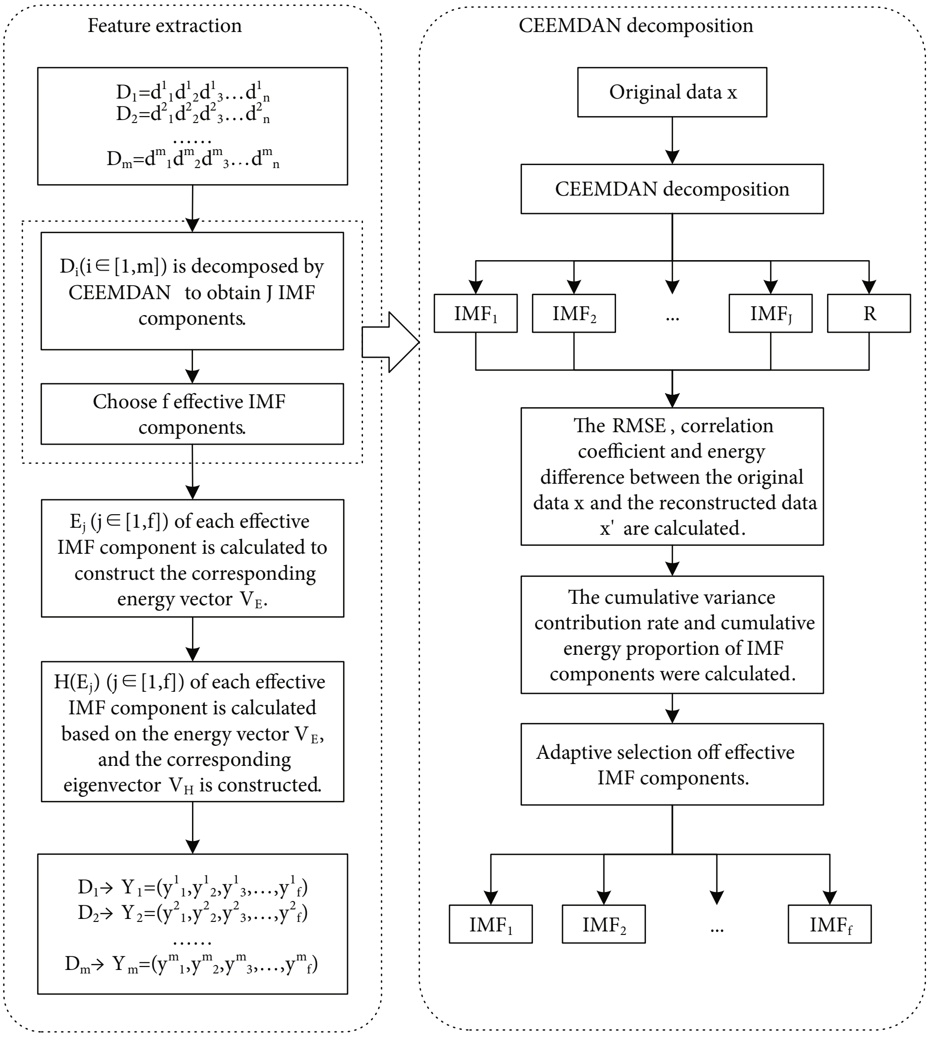

Feature extraction
$D_1=d^1{}_1d^1{}_2d^1{}_3\ldots d^1{}_n$
$D_2=d^2{}_1d^2{}_2d^2{}_3\ldots d^2{}_n$
......
$D_m=d^m{}_1d^m{}_2d^m{}_3\ldots d^m{}_n$
$D_i(i\in[1,m])$ is decomposed by CEEMDAN to obtain J IMF components.
Choose f effective IMF components.
E_j ($j\in[1,f]$) of each effective IMF component is calculated to construct the corresponding energy vector V_E.
$H(E_j)$ ($j\in[1,f]$) of each effective IMF component is calculated based on the energy vector V_E, and the corresponding eigenvector V_H is constructed.
$D_1\rightarrow Y_1=(y^1{}_1,y^1{}_2,y^1{}_3,\ldots,y^1{}_f)$
$D_2\rightarrow Y_2=(y^2{}_1,y^2{}_2,y^2{}_3,\ldots,y^2{}_f)$
......
$D_m\rightarrow Y_m=(y^m{}_1,y^m{}_2,y^m{}_3,\ldots,y^m{}_f)$
CEEMDAN decomposition
Original data x
CEEMDAN decomposition
IMF_1
IMF_2
...
IMF_J
R
The RMSE , correlation coefficient and energy difference between the original data x and the reconstructed data x' are calculated.
The cumulative variance contribution rate and cumulative energy proportion of IMF components were calculated.
Adaptive selection off effective IMF components.
IMF_1
IMF_2
...
IMF_f

Figure 1: Feature exploitation process of CEEMDAN energy entropy based on adaptive IMF selection.

Here, x_i ($\forall i\in[1,l]$) represents one training sample in the training sample set X, and l is the number of training samples; $\Phi: X\longrightarrow H$ represents the mapping function from the original data space to the high-dimensional feature space; ω and ρ represent the normal vector and compensation of the hyperplane in the high-dimensional feature space, respectively; $\nu\in(0,1]$ represents the trade-off parameter, which is used to control the proportion of support vectors in the training samples; ξ_i represents the relaxation variable, which indicates the misclassified degree of some training samples.

By introducing the Lagrange function to solve the quadratic programming problem, we can further construct the dual model by using the kernel function and obtain the following decision function:

$$f(x)=\operatorname{sgn}\left(\sum_{i=1}^{l}\alpha_i k\left(x_i,x_j\right)-\rho\right). \tag{14}$$

Here, $\rho=\sum_{i=1}^{l}\alpha_i k(x_i,x_j)$, and RBF (Radial Basis Function) is selected as the kernel function:

$$k\left(x_i,x_j\right)=\left\langle\Phi\left(x_i\right),\Phi\left(x_j\right)\right\rangle=\exp\left(\frac{-\left\|x_i-x_j\right\|^2}{2\sigma^2}\right). \tag{15}$$

From the above functions, we can see that the OCSVM's trade-off parameter ν and the RBF's parameter σ are two critical factors affecting the classification performance, and the optimization of these parameters is an important phase to obtain an excellent OCSVM classifier [28].

3.2. IABC Parameter Optimization Based on Multivariate Gaussian Mutation. In order to strengthen OCSVM's classification performance, this paper proposes a novel IABC-OCSVM anomaly detection model, which uses one improved ABC swarm intelligence algorithm to optimize the above parameters. More specifically, the ABC swarm intelligence algorithm is a typical multiobjective optimization method which imitates the searching behaviours of different bees, and its minimum searching model includes two basic elements: bee colony and honey source [29, 30]. Through the local optimization behaviour of individual bees in the searching process, the division and cooperation of three different bee colonies (the leader, the follower, and the scouter) can highlight the global optimization in the colonies. In order to homogenize the distribution of the honey source and improve the searching efficiency, this paper introduces the multivariate Gaussian mutation into the traditional ABC algorithm to dynamically guide the searching processes of different bee colonies, mainly including the following: (1) in the searching process of scouter bees, which is also the initial process of the honey source, since the initial honey source is mutated by the multivariate Gaussian mutation; (2) in the searching process of leader bees, wherein the OCSVM's classification accuracy of current global optimization is used to dynamically guide the searching process; and (3) in the searching process of follower bees, where the local optimum of leader searching is applied to carry out the variant search of the neighbouring honey source. Figure 2 describes the parameter optimization and anomaly detection process of the IABC-OCSVM model.

In the initialization process of the honey source, the initial honey source is mutated by the multivariate Gaussian mutation:

$$\begin{cases} x_{i,j} = x_j^{\min} + \text{rand}\,(0,1)\left(x_j^{\max} - x_j^{\min}\right), \\ p = \dfrac{1}{\sqrt{|\sum|(2\pi)^d}} \exp\left(-\dfrac{1}{2}(x-\mu)^T \overset{-1}{\sum}(x-\mu)\right). \end{cases} \tag{16}$$

Here, the expression of $x_{i,j}$ ($\forall i \in [1, N]$, $\forall j \in [1, D]$) is the initialization formula of the ith honey source, and N and D are the number and dimension of honey sources, respectively; in our algorithm, D is set to 2 due to the OCSVM's parameter v and RBF's parameter σ; $x_j^{\max}$ and $x_j^{\min}$ are the maximum and minimum in each dimension of the honey source; the expression of p is the multivariate Gaussian mutation formula, and $x = \{x_{1,j}, x_{2,j}, \cdots x_{N,j}\}$ and $p = \{p_{1,j}, p_{2,j}, \cdots p_{N,j}\}$ represent all honey sources before and after Gaussian mutation, respectively; μ and $\sum$ are the mean and covariance matrix of x, and $\sum^{-1}$ and d are the inverse and dimension of $\sum$.

In the searching process of leader bees, based on the Gbest searching strategy, the dynamic searching process is carried out through the guide of global optimization, and the OCSVM's classification accuracy in the current searching process is introduced to realize the adaptive search. The search of a neighbouring honey source can be expressed by

$$v_{i,j} = f_{p_i} p_{i,j} + \phi_{i,j}\left(f_{p_i} p_{i,j} - f_{p_k} p_{k,j}\right)\frac{1}{\text{iter}} + \psi_{i,j}\left(f_{p_g} p_{g,j} - f_{p_i} p_{i,j}\right)\frac{1}{\text{iter}}. \tag{17}$$

Here, $v_{i,j}$ represents a new honey source; $p_{i,j}$ is the honey source generated after the multivariate Gaussian mutation, and $p_i = \{p_{i,j}\}$ ($\forall j \in [1, D]$); $p_{k,j}$ ($k \neq i$, $\forall k \in [1, N]$) is a neighbouring honey source randomly selected from all honey sources, and is different from the current honey source $p_{i,j}$; $p_{g,j}$ represents the global optimal solution; f_{p_*} represents the OCSVM's classification accuracy corresponding to the honey source p_*; $\phi_{i,j}$ is one random number in the range $[-1, 1]$; $\psi_{i,j}$ is one random number in the range $[0, 1]$; iter is the goal-setting number of iterations.

In the searching process of follower bees, according to the local optimum in the searching process of leader bees, the mutation operation can be performed on the neighbouring honey source, and the OCSVM's classification accuracy in the current searching process is introduced to realize the variant search. The search of a neighbouring honey source can be expressed by

$$v_{i,j} = f_{p_l} p_{l,j} + \phi_{i,j}\left(f_{p_i} p_{i,j} - f_{p_k} p_{k,j}\right)\frac{1}{\text{iter}}. \tag{18}$$

Here, $p_{l,j}$ represents the optimal solution in the searching process of leader bees.

In the whole searching process, each honey source represents a feasible solution, and the yield of a honey source is consistent with the fitness of a feasible solution, which is calculated by

$$\text{Fit}_i = \begin{cases} \dfrac{1}{1 + f_{p_i}}, & f_{p_i} \geq 0, \\ 1 + \text{abs}\left(f_{p_i}\right), & f_{p_i} < 0. \end{cases} \tag{19}$$

Here, f_{p_i} represents the OCSVM's classification accuracy corresponding to the honey source p_i.

4. Experimental Evaluation and Discussion

4.1. Experimental Data and Preparation. In order to verify the effectiveness and advantage of the proposed approach, we use some real-world data captured from one local oilfield IIoT system in the northeastern part of China to perform some experimental evaluations, and the basic system architecture can be briefly stated as follows: all IIoT sensors are physically deployed in the wellheads and perform the real-time data acquisition of the pumping well working status, mainly including the pressure, the motor speed, the flow, and some electrical parameters. By using the WIA-PA protocol [31], the IIoT sensors in one wellhead send these sensing data to one RTU (Remote Terminal Unit) which can be regarded as the data collector in our approach, and the RTU forwards these sensing data to the upper monitoring center by using the Modbus/TCP protocol. After capturing the Modbus/TCP packets in one RTU for 9 hours, we totally obtain 109,672 IIoT sensing data, and form 225 data sequences by the initial preparation.

4.2. Experimental Comparison and Analysis on Different Feature Exploitations. For the obtained data sequence set, in each experiment, we randomly select 200 data sequences as the normal data sequences and construct 100 abnormal data sequences by injecting or falsifying some malicious data which cannot conform to the regular production pattern.

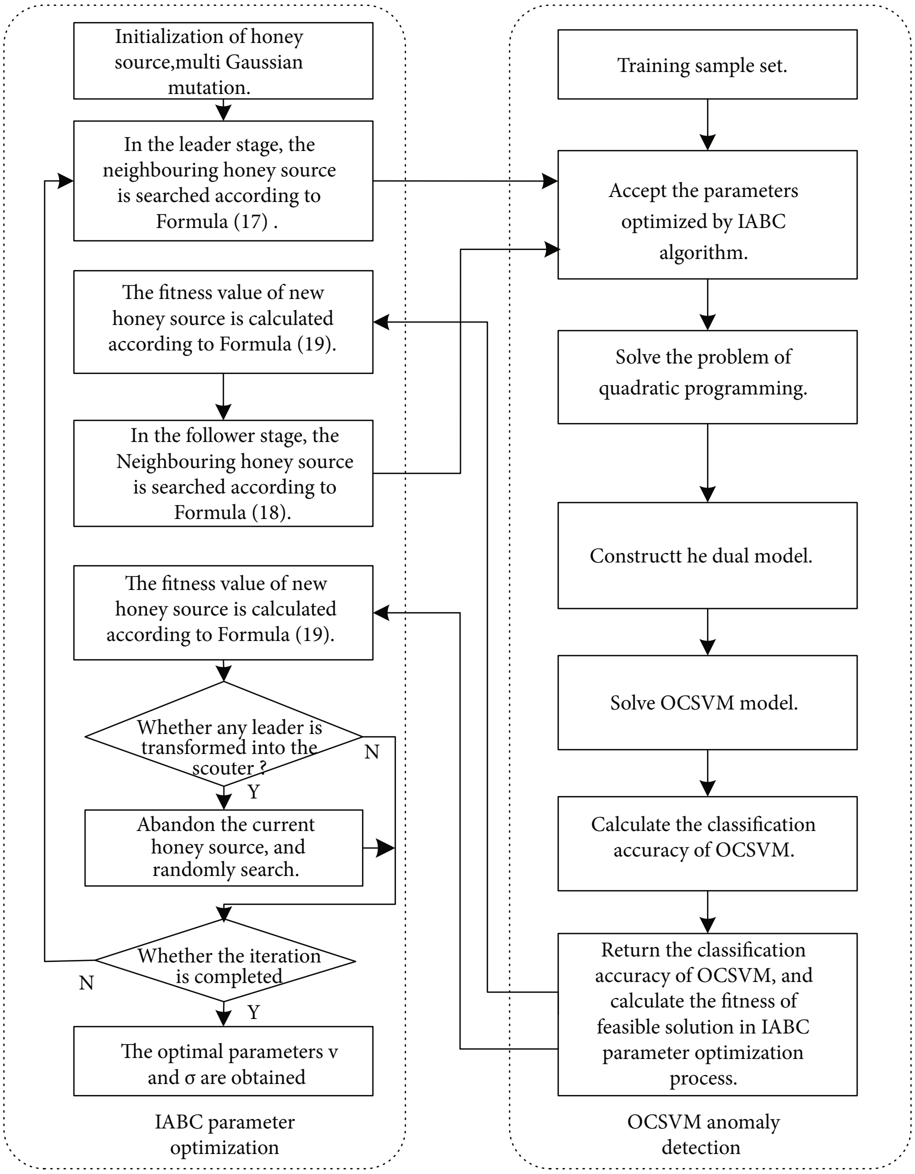
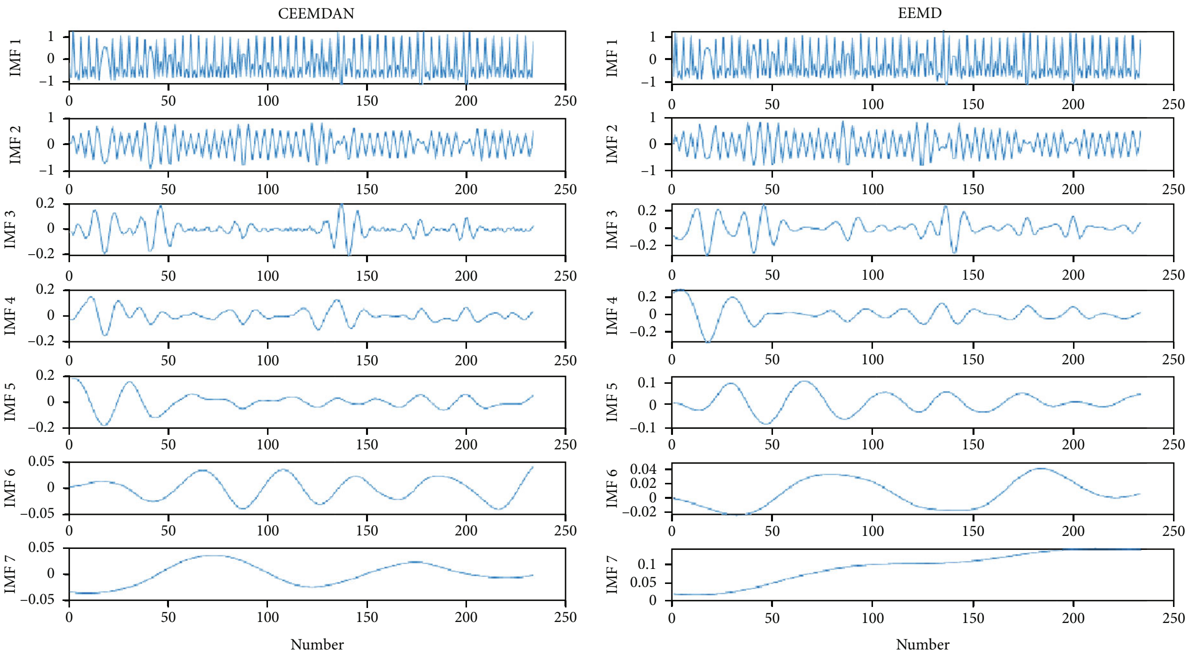

Initialization of honey source,multi Gaussian mutation.
In the leader stage, the neighbouring honey source is searched according to Formula (17) .
The fitness value of new honey source is calculated according to Formula (19).
In the follower stage, the Neighbouring honey source is searched according to Formula (18).
The fitness value of new honey source is calculated according to Formula (19).
Whether any leader is transformed into the scouter ?
N
Y
Abandon the current honey source, and randomly search.
Whether the iteration is completed
N
Y
The optimal parameters v and σ are obtained
IABC parameter optimization
Training sample set.
Accept the parameters optimized by IABC algorithm.
Solve the problem of quadratic programming.
Constructt he dual model.
Solve OCSVM model.
Calculate the classification accuracy of OCSVM.
Return the classification accuracy of OCSVM, and calculate the fitness of feasible solution in IABC parameter optimization process.
OCSVM anomaly detection

Figure 2: Parameter optimization and anomaly detection of IABC-OCSVM model.

After the proposed adaptive CEEMDAN feature exploitation, we record all normal and abnormal data samples as “+1” and “-1” data samples, respectively. Moreover, all normal data samples are used to train the IABC-OCSVM anomaly detection classifier, and the test sample set consists of randomly selected 100 “+1” data samples and 100 “-1” data samples. Additionally, because the number of malicious data in each data sequence can directly reflect different attack powers, we design 5 incremental attack powers when constructing 100 abnormal data sequences. From attack power 1 to 5, the number of malicious data in each data sequence is set from 6 to 10. In order to verify the main advantage of adaptive CEEMDAN feature exploitation in the multiscale analysis of data sequences, we introduce the classification accuracy as one significant evaluation indicator to perform two distinct groups of experiments: the first group of experiments compare the CEEMDAN decomposition with the EEMD decomposition whose IMF components are depicted in Figure 3, and the training and test classification accuracies of their extracted features are shown in Table 1; the second group of experiments compare different test classification accuracies of CEEMDAN energy entropy, CEEMDAN singular spectrum entropy, and EEMD singular value decomposition, and the experimental results are shown in Table 2.

As seen in Table 1, when the average training classification accuracies of EEMD and CEEMDAN decompositions reach 92.30% and 95.10%, their average test classification accuracies are 86.50% and 89.00%, respectively. From the above compared results, it can be concluded that both the training classification accuracy and the test classification accuracy of CEEMDAN decomposition are larger than the ones of EEMD decomposition. That is to say, the CEEMDAN decomposition can effectively discover more intrinsic



(a)

(b)

Figure 3: Compared results of CEEMDAN and EEMD decompositions.

Table 1: Training and test classification accuracies of CEEMDAN and EEMD decompositions under different attack powers.

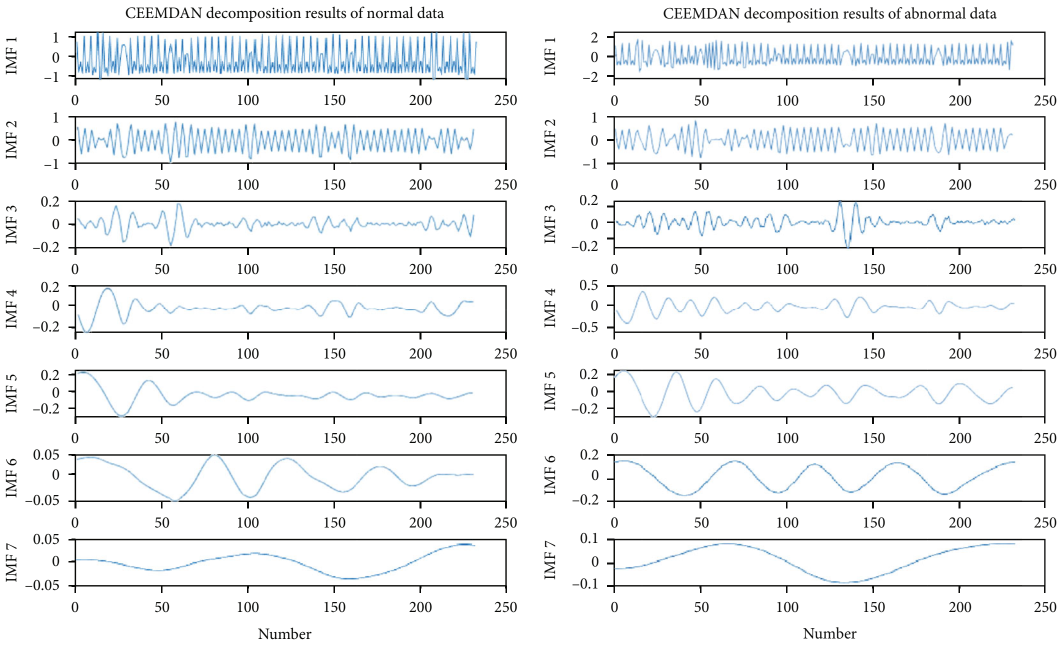
Attack power	CEEMDAN		EEMD	
	Training accuracy	Test accuracy	Training accuracy	Test accuracy
1	96.0%	86.5%	92.5%	81.5%
2	93.5%	88.0%	91.0%	87.5%
3	91.0%	90.0%	92.0%	88.5%
4	96.5%	87.0%	94.5%	87.0%
5	98.5%	93.5%	91.5%	88.0%
Average	95.10%	89.00%	92.30%	86.50%

Table 2: Test classification accuracies of three different feature exploitation methods.

Attack power	CEEMDAN energy entropy	CEEMDAN singular spectrum entropy	EEMD singular value decomposition
1	86.5%	82.0%	80.5%
2	88.0%	85.0%	82.5%
3	90.0%	83.5%	83.5%
4	87.0%	85.5%	81.0%
5	93.5%	88.5%	83.5%
Average	89.00%	84.90%	82.20%

characteristics of original data, and the corresponding extracted features can contribute to improving the classification accuracy of the OCSVM classifier.

From Table 2, we can see that, for the feature exploitation methods based on CEEMDAN singular spectrum entropy and EEMD singular value decomposition, their average classification accuracies are 84.90% and 82.20%, respectively. Obviously, these accuracies are less than that of the proposed feature exploitation method, which can reach 89.00%. Through the comprehensive comparison of these two tables, we can conclude that, on the one hand, the proposed feature exploitation method has distinct advantages in the improvement of classification accuracy, on the other hand, these results indirectly show that the



(a)

(b)

Figure 4: CEEMDAN decomposition results of normal and abnormal data samples.

proposed method can more accurately describe the change of industrial communication behaviour. Additionally, as the attack power increases, that is, the number of malicious data in each data sequence increases, the classification accuracy generally shows an upward trend. In other words, the proposed feature exploitation method is more sensitive to the number of malicious data, which can help to improve the anomaly detection performance.

Figure 4 compares the CEEMDAN decomposition results of normal and abnormal data samples. When some abnormal communication behaviours occur in industrial production, not only the energy information and probability information in all data sequences change accordingly, but also the implicit information in each data sequence differs from others under different scales. Figure 5 depicts the energy proportion and variance contribution rate of different IMF components after the CEEMDAN decomposition. Totally, the energy and variance of each IMF component can appear surprisingly distinct from each other. Based on this result, when selecting the appropriate feature parameters, we can focus on the IMF components which have larger contribution rates and remove the IMF components with insufficient information.

4.3. Experimental Comparison and Analysis on Different Parameter Optimizations. In order to further illustrate the influence of parameter optimization on the OCSVM's classification performance, we, respectively, use the traditional ABC algorithm and PSO (Particle Swarm Optimization) algorithm to optimize the OCSVM classifier and compare their classification accuracies by performing some experiments under 5 attack powers. Moreover, the fitness curves in two parameter optimization processes are shown in Figure 6, and the training and test classification accuracies of two classifiers are compared in Table 3. Obviously, the above experimental results can directly reflect that two parameter optimization algorithms have different effects on the OCSVM's classification performance. In terms of classification accuracy, the average training and test classification accuracies of the ABC-OCSVM classifier are 95.10% and 89.00%, respectively. Differently, the average training and test classification accuracies of the PSO-OCSVM classifier are 98.00% and 83.20%, respectively. Although the average training classification accuracy of the ABC-OCSVM classifier is slightly lower than that of the PSO-OCSVM classifier, the average test classification accuracy of the ABC-OCSVM classifier can present a trend of higher resolution. That is, the ABC-OCSVM classifier can have a smaller span change from training accuracy to test accuracy, and obtain a relatively higher classification accuracy in practice. Also, the above compared results have proven that different combinations of OCSVM's trade-off parameter ν and RBF's parameter σ can have a pronounced impact on the OCSVM's classification accuracy, and one fine parameter optimization algorithm can help to improve the detection performance of OCSVM's classifier.

In order to obtain better parameters and further improve the anomaly detection efficiency, we propose an IABC-OCSVM anomaly detection classifier optimized by the improved ABC algorithm. To evaluate this classifier, we perform some compared experiments to analyze the training classification accuracy, test classification accuracy, and test time between the traditional ABC-OCSVM classifier and the IABC-OCSVM classifier, and Table 4 shows the experimental results under 5 attack powers. As shown in Table 4, under a similar average test time, the average training and

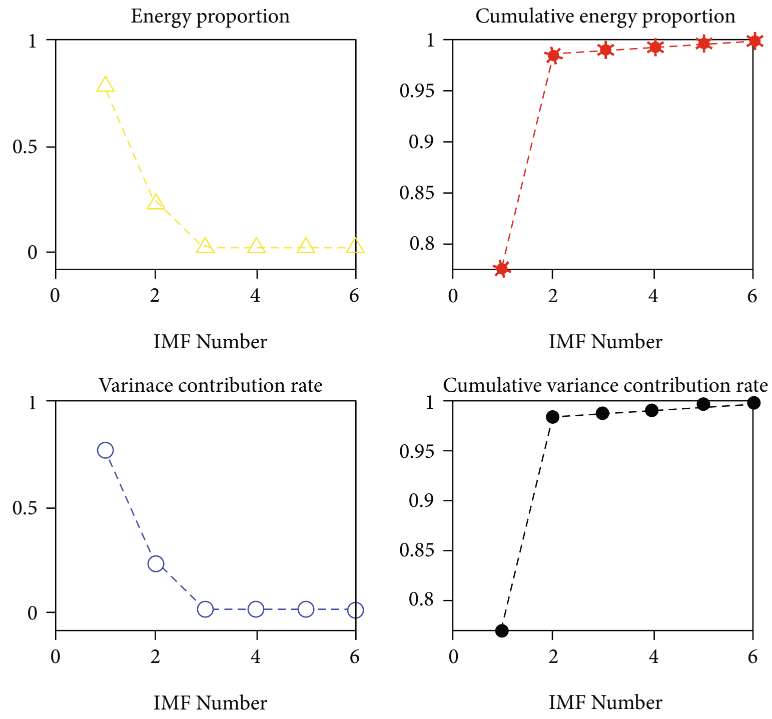


Figure 5: Energy proportion and variance contribution rate of different IMF components.

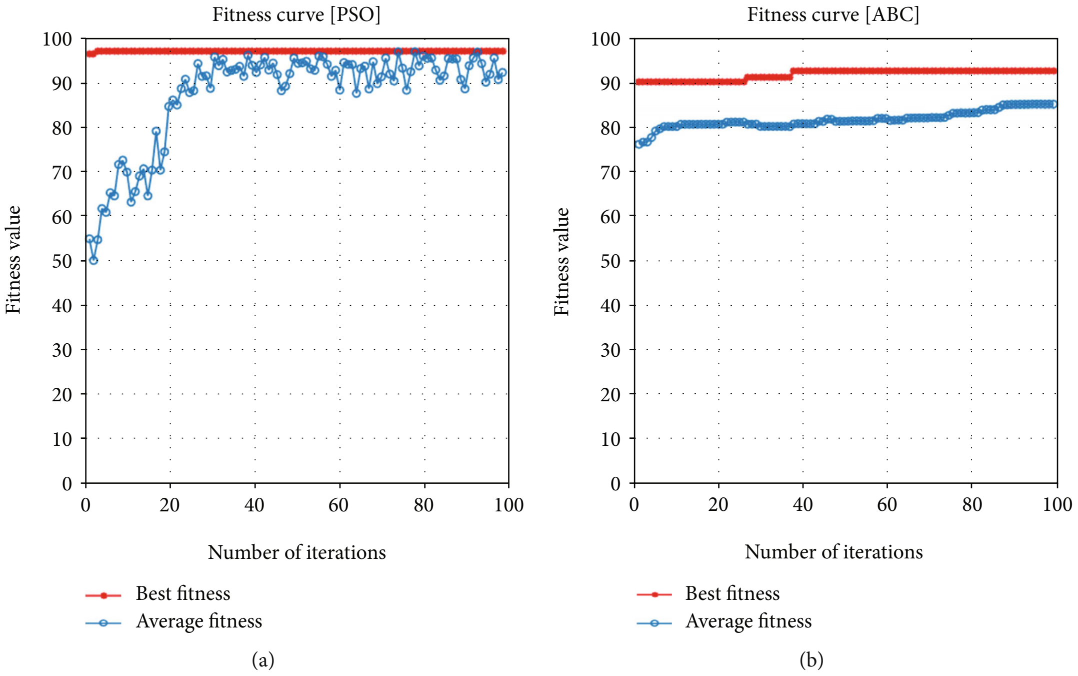


Figure 6: Fitness curves in the traditional ABC and PSO parameter optimization processes.

test classification accuracies of the IABC-OCSVM classifier can reach 94.50% and 89.80%, respectively. Although the average training classification accuracy of the IABC-OCSVM classifier is slightly lower than that of the ABC-OCSVM classifier, its average test classification accuracy is higher than the one of the ABC-OCSVM classifier. Especially, for the test samples with a stronger attack power, the test classification accuracy of the IABC-OCSVM classifier is significantly higher than the one of the ABC-OCSVM classifier. For example, under attack power 5, the test classification accuracy of the IABC-OCSVM classifier can reach 95.50%, which grows by two percentage points. More narrowly, Figure 7 gives the classification results of training samples and test samples under attack power 5. Furthermore, Figure 7(a) shows 3 training

TABLE 3: Training and test classification accuracies of traditional ABC-OCSVM and PSO-OCSVM anomaly detection classifiers.

Attack power	ABC-OCSVM		PSO-OCSVM	
	Training accuracy	Test accuracy	Training accuracy	Test accuracy
1	96.0%	86.5%	98.0%	80.5%
2	93.5%	88.0%	96.5%	83.0%
3	91.0%	90.0%	98.0%	85.0%
4	96.5%	87.0%	100.0%	84.0%
5	98.5%	93.5%	98.0%	83.5%
Average	95.10%	89.00%	98.00%	83.20%

TABLE 4: Detection efficiency comparisons between traditional ABC-OCSVM and PSO-OCSVM anomaly detection classifiers.

Attack power	ABC-OCSVM			IABC-OCSVM		
	Training accuracy	Test accuracy	Test time	Training accuracy	Test accuracy	Test time
1	96.0%	86.5%	0.0079 s	96.0%	86.5%	0.0076 s
2	93.5%	88.0%	0.0079 s	92.0%	88.5%	0.0080 s
3	91.0%	90.0%	0.0090 s	91.0%	90.5%	0.0081 s
4	96.5%	87.0%	0.0076 s	95.0%	88.0%	0.0079 s
5	98.5%	93.5%	0.0086 s	98.5%	95.5%	0.0089 s
Average	95.10%	89.00%	0.0082 s	94.50%	89.80%	0.0081 s

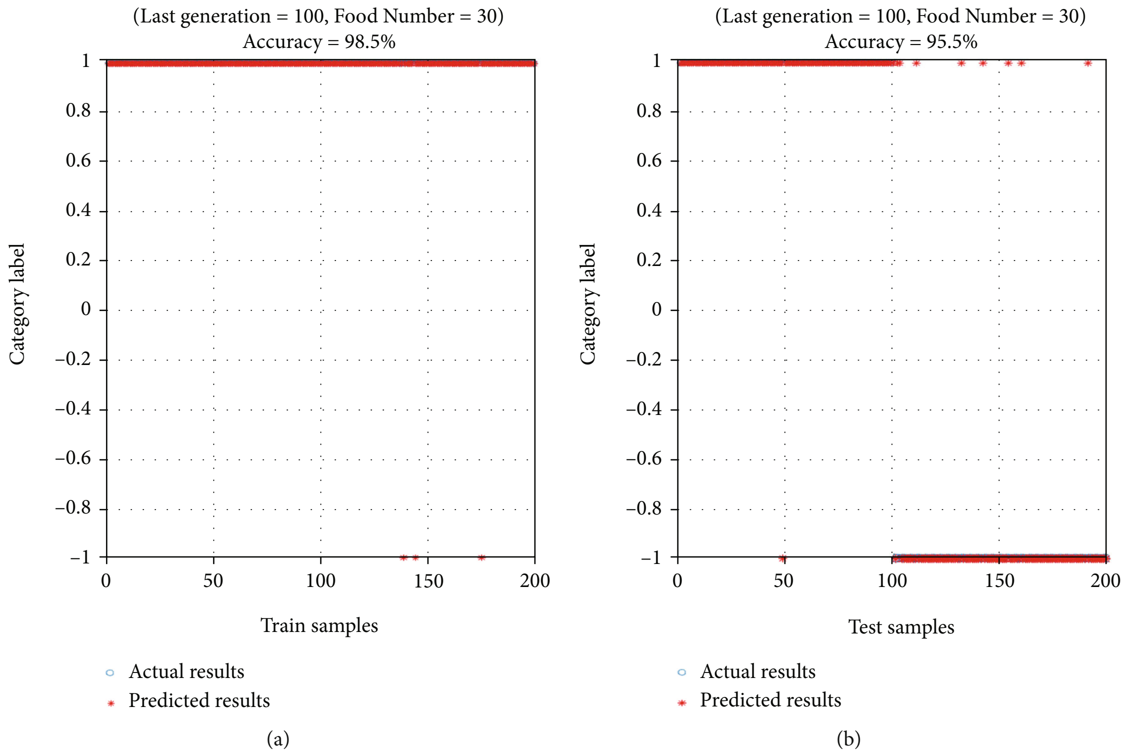


FIGURE 7: Classification results of training samples and test samples under attack power 5.

samples that are wrongly classified in all 200 training samples, and Figure 7(b) shows 9 test samples that are wrongly classified in all 200 test samples. Additionally, the average test time of the IABC-OCSVM classifier is only 0.0081 s, which still reaches the millisecond level and has a strong real-time classification capability. From the comprehensive evaluation of classification accuracy and detection time, the proposed IABC-OCSVM classifier has a higher detection efficiency.

5. Conclusions

The sparsity of IIoT sensing data may bring out tremendous challenges to implement the global anomaly detection, and the collection and analysis of all sparse sensing data in a local wireless sensor network can provide a feasible opportunity to develop an experienced machine-learning anomaly detection model by exploring their statefulness and correlation characteristics. From this point of view, this paper proposes a novel IABC-OCSVM anomaly detection approach for IIoT sparse data, which can successfully collaborate the adaptive CEEMDAN feature exploitation with the intelligent optimizing OCSVM classifier. Firstly, the multiscale analysis of IIoT data sequences is carried out through the CEEMDAN decomposition, and the effective IMF components can be adaptively selected to calculate the corresponding energy entropies and construct the final data samples. Secondly, this approach designs one improved ABC algorithm based on a multivariate Gaussian mutation to optimize the important parameters of a traditional OCSVM classifier, which can unambiguously match with the adaptive CEEMDAN feature exploitation method. Finally, many experiments are performed to evaluate the proposed approach: on the one hand, by comparing different feature exploitation methods, we prove that the proposed feature exploitation method can more accurately describe the change of industrial communication behaviour, and have distinct advantages to improve the classification accuracy; on the other hand, by comparing different parameter optimization algorithms, we prove that the proposed IABC-OCSVM classifier can have higher detection efficiency.

Data Availability

In this manuscript, the analyzed data are some real-world data captured from one local oilfield IIoT system northeast of China, and some contents and specific parameters are not completely open to the public due to the commercialized secrets. If other researchers want to use these data, please contact the corresponding author or the first author. The requests for data will be considered by them after a confidentiality agreement.

Conflicts of Interest

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

Acknowledgments

This work is supported by the Defense Industrial Technology Development Program (Grant No. JCKY2020205B022) and the Scientific Research Project of Liaoning Educational Department (Grant No. LJKZ0082).

References

[1] S. Vitturi, C. Zunino, and T. Sauter, “Industrial communication systems and their future challenges: next-generation ethernet, IIoT, and 5G,” *Proceedings of the IEEE*, vol. 107, no. 6, pp. 944–961, 2019.

[2] S. Mantravadi, R. Schnyder, C. Moller, and T. D. Brunoe, “Securing IT/OT links for low power IIoT devices: design considerations for industry 4.0,” *IEEE Access*, vol. 8, pp. 200305–200321, 2020.

[3] G. Rathee, M. Balasaraswathi, K. P. Chandran, S. D. Gupta, and C. S. Boopathi, “A secure IoT sensors communication in industry 4.0 using blockchain technology,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 533–545, 2021.

[4] Z. Cai, Z. He, X. Guan, and Y. Li, “Collective data-sanitization for preventing sensitive information inference attacks in social networks,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[5] Z. Cai and X. Zheng, “A private and efficient mechanism for data uploading in smart cyber-physical systems,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[6] K. Tange, M. de Donno, X. Fafoutis, and N. Dragoni, “A systematic survey of industrial internet of things security: requirements and fog computing opportunities,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2489–2520, 2020.

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

[8] M. Wan, J. Li, Y. Liu, J. Zhao, and J. Wang, “Characteristic insights on industrial cyber security and popular defense mechanisms,” *China Communications*, vol. 18, no. 1, pp. 130–150, 2021.

[9] X. Zheng and Z. Cai, “Privacy-preserved data sharing towards multiple parties in industrial IoTs,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[10] A. C. Panchal, V. M. Khadse, and P. N. Mahalle, “Security issues in IIoT: a comprehensive survey of attacks on IIoT and its countermeasures,” in *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)*, Lonavala, India, November 2018.

[11] J. Yakubu, A. M. Abdulhamid, H. A. Christopher, H. Chiroma, and M. Abdullahi, “Security challenges in fog-computing environment: a systematic appraisal of current developments,” *Journal of Reliable Intelligent Environments*, vol. 5, no. 4, pp. 209–233, 2019.

[12] F. Khan, M. A. Jan, A. Rehman, S. Mastorakis, M. Alazab, and P. Watters, “A secured and intelligent communication scheme for IIoT-enabled pervasive edge computing,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5128–5137, 2021.

[13] Z. Cai and Z. He, “Trading private range counting over big IoT data,” in *The 39th IEEE International Conference on Distributed Computing Systems (ICDCS 2019)*, Dallas, USA, 2019.

[14] A. Jurcut, T. Niculcea, P. Ranaweera, and N. A. le-Khac, “Security considerations for internet of things: a survey,” *SN Computer Science*, vol. 1, no. 4, pp. 1–19, 2020.

[15] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, “Generative adversarial networks,” *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.

[16] J. Wan, J. Li, M. Imran, D. Li, and Fazal-e-Amin, “A blockchain-based solution for enhancing security and privacy

in smart factory," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3652–3660, 2019.

[17] Y. Zhang, R. H. Deng, D. Zheng, J. Li, P. Wu, and J. Cao, "Efficient and robust certificateless signature for data crowdsensing in cloud-assisted industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5099–5108, 2019.

[18] S. Qi, Y. Lu, W. Wei, and X. Chen, "Efficient data access control with fine-grained data protection in cloud-assisted IIoT," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2886–2899, 2020.

[19] S. Z. Tajalli, M. Mardaneh, E. Taherian-Fard et al., "DoS-resilient distributed optimal scheduling in a fog supporting IIoT-based smart microgrid," *IEEE Transactions on Industry Applications*, vol. 56, no. 3, pp. 2968–2977, 2020.

[20] J. M. Mcginthy and A. J. Michaels, "Secure industrial internet of things critical infrastructure node design," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8021–8037, 2019.

[21] D. Liu, H. Zhen, D. Kong et al., "Sensors anomaly detection of industrial internet of things based on isolated forest algorithm and data compression," *Scientific Programming*, vol. 2021, Article ID 6699313, 9 pages, 2021.

[22] D. Wu, Z. Jiang, X. Xie, X. Wei, W. Yu, and R. Li, "LSTM learning with Bayesian and Gaussian processing for anomaly detection in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5244–5253, 2020.

[23] P. Zhan, S. Wang, J. Wang et al., "Temporal anomaly detection on IIoT-enabled manufacturing," *Journal of Intelligent Manufacturing*, vol. 32, no. 6, pp. 1669–1678, 2021.

[24] S. Tian, X. Bian, Z. Tang, K. Yang, and L. Li, "Fault diagnosis of gas pressure regulators based on CEEMDAN and feature clustering," *IEEE Access*, vol. 7, pp. 132492–132502, 2019.

[25] M. Wan, W. Shang, and P. Zeng, "Double behavior characteristics for one-class classification anomaly detection in networked control systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3011–3023, 2017.

[26] Z. Ghafoori, S. M. Erfani, S. Rajasegarar, J. C. Bezdek, S. Karunasekera, and C. Leckie, "Efficient unsupervised parameter estimation for one-class support vector machines," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 5057–5070, 2018.

[27] A. A. Shorman, H. Faris, and I. Aljarah, "Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 7, pp. 2809–2825, 2020.

[28] Y. Xiao, H. Wang, and W. Xu, "Parameter selection of Gaussian kernel for one-class SVM," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 941–953, 2015.

[29] L. Cui, G. Li, X. Wang et al., "A ranking-based adaptive artificial bee colony algorithm for global numerical optimization," *Information Sciences*, vol. 417, pp. 169–185, 2017.

[30] L. Zhang, S. Wang, K. Zhang et al., "Cooperative artificial bee colony algorithm with multiple populations for interval multiobjective optimization problems," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 5, pp. 1052–1065, 2019.

[31] L. Sun, W. Liang, K. Wang, S. Zhang, and Q. Miao, "WIA-PA protocol conformance testing method based on Petri net model for device life cycle," *Information and Control*, vol. 44, no. 6, pp. 703–710, 2015.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8586016, 8 pages
https://doi.org/10.1155/2021/8586016

Research Article

Masked Face Detection Algorithm in the Dense Crowd Based on Federated Learning

Rui Zhu,[1] **Kangning Yin**,[2] **Hang Xiong**,[1] **Hailian Tang**,[2] **and Guangqiang Yin**[2]

[1] *School of Information and Communication Engineering, University of Electronic Science and Technology of China, Sichuan 611731, China*
[2] *School of Information and Software Engineering, University of Electronic Science and Technology of China, Sichuan 611731, China*

Correspondence should be addressed to Guangqiang Yin; yingq@uestc.edu.cn

Received 20 July 2021; Accepted 15 September 2021; Published 4 October 2021

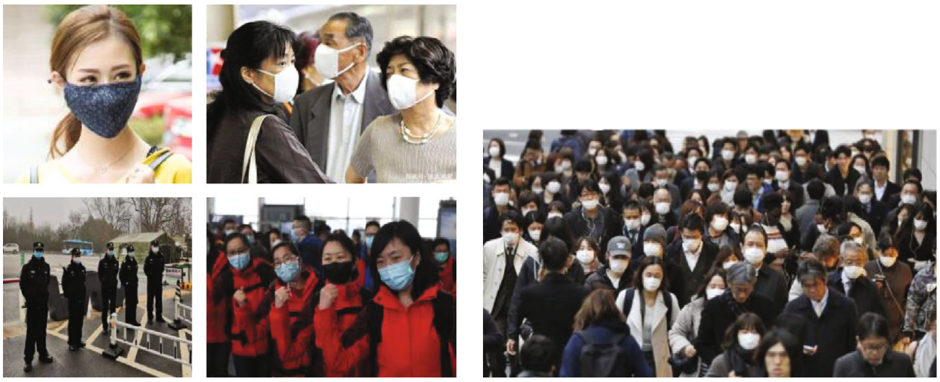
Academic Editor: Yan Huang

Wearing masks is an effective and simple method to prevent the spread of the COVID-19 pandemic in public places, such as train stations, classrooms, and streets. It is of positive significance to urge people to wear masks with computer vision technology. However, the existing detection methods are mainly for simple scenes, and facial missing detection is prone to occur in dense crowds with different scales and occlusions. Moreover, the data obtained by surveillance cameras in public places are difficult to be collected for centralized training, due to the privacy of individuals. In order to solve these problems, a cascaded network is proposed: the first level is the Dilation RetinaNet Face Location (DRFL) Network, which contains Enhanced Receptive Field Context (ERFC) module with the dilation convolution, aiming to reduce network parameters and locate faces of different scales. In order to adapt to embedded camera devices, the second level is the SRNet20 network, which is created by Neural Architecture Search (NAS). Due to privacy protection, it is difficult for surveillance video to share in practice, so our SRNet20 network is trained in federated learning. Meanwhile, we have made a masked face dataset containing about 20,000 images. Finally, the experiments highlight that the detection mAP of the face location is 90.6% on the Wider Face dataset, and the classification mAP of the masked face classification is 98.5% on the dataset we made, which means our cascaded network can detect masked faces in dense crowd scenes well.

1. Introduction

COVID-19 spreads rapidly among the population and has a serious impact on society, economy, and people's normal lives. The weekly epidemiological update of the World Health Organization (WHO) [1] presented that the cumulative number of cases reported globally is now over 186 million, and the number of deaths exceeds 4 million. Fortunately, wearing masks is an effective and simple method to prevent the spread of COVID-19 [2], and almost everyone is obligated to wear a face mask in public places. Relying solely on manpower for inspections inevitably has disadvantages, such as high work intensity, low efficiency, and timeliness, but using detection algorithms to complete this task can save many human resources. Using computer vision technology to detect whether people wear masks and to give corresponding reminders can achieve the purpose of noncontact detection, preventing the spread of the virus and ensuring people's safety.

Moreover, most of the existing algorithms train the model by collecting the data together, but the reality is that videos captured by the cameras in public places will not be easily obtained because of personal privacy [3]. The surveillance video data of public place belong to different departments, which make the data form an isolated island and difficult to be concentrated together for model training. As a new distributed machine learning method, federated learning, with the help of the storage and computing capacity of the device itself, can cobuild the model without data out of the local, so as to protect data privacy and effectively solve the problem of data island [4].



(a) Examples in public dataset

(b) Expected dataset

Figure 1: Comparing images.

Therefore, the task is decomposed into two subnetworks. The first network is used for the general face location, and the second is used for the masked face classification. The main contributions of our paper are listed below:

(1) The DRFL network is proposed and trained on the Wilder Face dataset to locate faces in dense crowds

(2) The SRNet20 network is designed with NAS and trained by methods of federated learning to classify masked faces

(3) A masked face dataset is created and contains 18,000 images in the train set and 1,751 images in the test set. In order to facilitate other researchers, this dataset is also published on the GitHub: https://github.com/woshizr/masked-Face

2. Related Work

2.1. Face Detection Algorithms. Face detection is closely related to general object detection. In recent years, object detection algorithms have developed rapidly, which are mainly divided into two categories: single stage object detection algorithms, represented by YOLO [5] and RetinaNet [6], divide the image into regions and predict bounding boxes and probabilities for each region simultaneously. Therefore, this kind of algorithm is faster. The two-stage object detection algorithms, represented by RCNN [7] and FPN [8], generate a large number of proposal regions, which then classify the proposals into foreground classes or background. Therefore, the accuracy of this kind of algorithm is higher. Based on the object detection algorithms, a large number of face detection algorithms and masked face detection algorithms have been developed: MTCNN [9] uses 3 cascaded networks to achieve face detection; Face RCNN [10] is based on Faster RCNN [11] for face detection; SSH [12] enhances the feature extraction of convolutional layers with different depths to achieve multiscale face detection; PyramidBox [13] uses the context information of the face to improve the detection of occluded faces; Didi company proposes a mask wearing detection algorithm based on DFS [14], the algorithm detects the face region first, expands the face area based on the face features, and then uses the attention mechanism to find the mask area, and finally detects whether the face is wearing a mask; AIZOO proposes a lightweight mask wearing detection algorithm [15] based on SSD and improves the network structure; RetinaMask [16] detects the face with mask by adding attention mechanism in context module.

Many efforts have also been made in society to help with masked face detection. In [17], three kinds of masked face datasets are proposed, including masked face detection dataset (MFDD), real-world masked face recognition dataset (RMFRD), and simulated masked face recognition dataset (SMFRD). Among them, RMFRD is currently the world's largest real-world masked face dataset, which provides the correct masked face dataset (CMFD) and the incorrectly masked face dataset (IMFD), and some sample images are shown in Figure 1(a); however, the dataset in dense scene is often shown as Figure 1(b). Therefore, the performance of the algorithm in Figure 1(b) can better illustrate the advantages and disadvantages of the algorithm.

2.2. Federated Learning. The development of artificial intelligence technology has encountered two main challenges: one is that data exists in the form of data islands in most industries; the other is that training models require a lot of data, and improper collection of data will make it difficult to protect the privacy and security of data. In the traditional centralized machine learning method, the data collected from different devices need to be uploaded to the cloud [18], and the central server in the cloud uses the data to train the model, as shown in Figure 2. Data are directly exposed in the cloud, which is difficult to protect user privacy [19].

To solve the above problems, in 2016, Google proposed federated learning [20], a machine learning framework based on user privacy protection. Their main idea is to build machine learning models based on data distributed on multiple devices and prevent user privacy from being leaked. Federated learning allows the device to use local data to train the model, after the training, the local device does not need to send sensitive data to the cloud, but only needs to upload the model parameters [21]. The central server of federated learning then aggregates the collected model parameters,

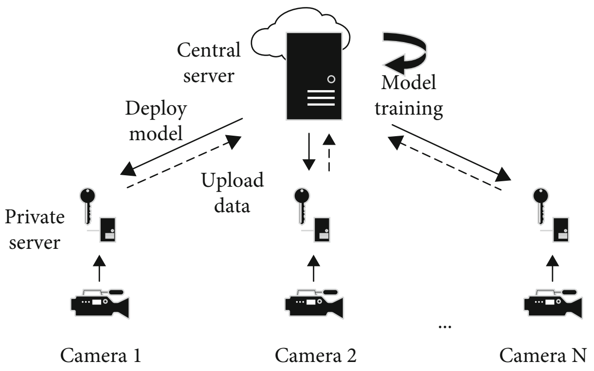


Figure 2: Centralized training.

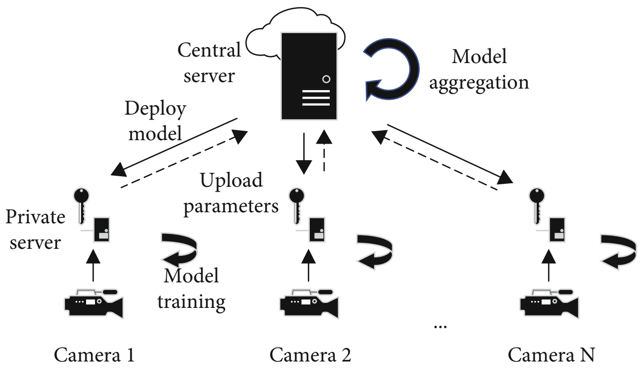


Figure 3: Federated training.

and this process continues until the joint training models reach the expected accuracy, as shown in Figure 3.

The data of the provider are kept locally, and the leakage of data privacy is suppressed from the source. Of course, federated learning also involves many aspects; in this paper, it mainly involves the use of multiparty data sources for federated training.

3. Models and Improvements

The whole algorithm is divided into two cascaded subnetworks: a general face location network and a face classification network with masks. The algorithm process is shown in Figure 4.

All face boxes are found in the input image through the face location network, and then whether each face box is wearing a mask is determined through the classification network. Especially, a federated training method is used to keep the data locally, and only the model parameters are transferred between clients, when training the classification network.

3.1. Dilation RetinaNet Face Location Network. The DRFL network is inspired by RetinaNet. In order to solve the problems of occlusion and multiscale faces in the masked face detection task in dense crowd, the backbone of the DRFL network uses ResNet50 [22] as the feature extraction network. C3, C4, and C5 represent the low-level feature, middle-level feature, and high-level feature extracted for the image. P3, P4, and P5 are feature fusion in the FPN network through upsampling and residual connection. The fused features are used to enhance feature extraction, increase the scope of the receptive field, and enhance the robustness of small-scale face detection through independent Enhanced Receptive Field Context (ERFC) module. The DRFL network structure is shown in Figure 5.

The entire feature extraction network combines top-down and bottom-up feature fusion strategies to improve the multiscale prediction network. Finally, a multitask loss function is used to fully consider the central point distance between the face and the detection frame, overlap rate, and key point information, thereby improving the accuracy of face detection.

3.2. Enhanced Receptive Field Context Module. The ERFC module with special dilation convolution is used to extract the feature output by the FPN. The advantage of using dilation convolution is that it can increase the receptive field while avoiding the loss of information caused by the pooling operation. Each convolution output contains a larger range of information and captures multiscale context information. As shown in Figure 6, (a) corresponds to 3×3 convolution with dilation rate 1, which is the same as ordinary convolution operation, (b) corresponds to 3×3 convolution with dilation rate 2, and the receptive field has increased to 5×5.

The specific operation of ERFC module is to first compute the input features by the 3×3 convolution, and then one of them is to enhance the extraction of context information through the parallel 3×3 convolution with dilation rate 1 and 3×3 convolution with dilation rate 2, in order to improve the detection robustness of occluded faces. At the same time, the local parameters are reduced by 16.7% without changing the receptive field and detection accuracy. Finally, all the outputs are concatenated as the output of the entire ERFC module and transmitted to the next network as shown in Figure 6(c).

3.3. Masked Face Classification Network. The significance of NAS is to solve the parameter adjustment problem of deep learning models, which is a cross-research that combines optimization and machine learning. Before deep learning, the traditional machine learning models might also encounter the problem of parameter adjustment. Because the structure of the shallow model is relatively simple, most studies unify the structure of the model as a super parameter to search, such as the number of hidden neurons in the three-layer neural network. The methods for optimizing these hyperparameters are mainly black box optimization methods, such as evolutionary optimization, Bayesian optimization, and reinforcement learning.

However, in deep learning, with the expansion of the model scale, the number of super parameters also increases, which brings new challenges to the optimization problem. The search space of NAS directly affects the difficulty of optimization. A simple search strategy [23] in neural network search is to multiply each branch by a weight during training and to send the result to the next level. After

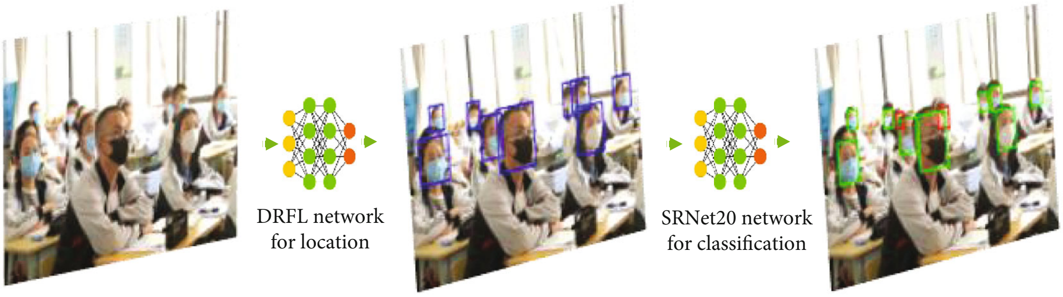


Figure 4: Two cascaded subnetworks.

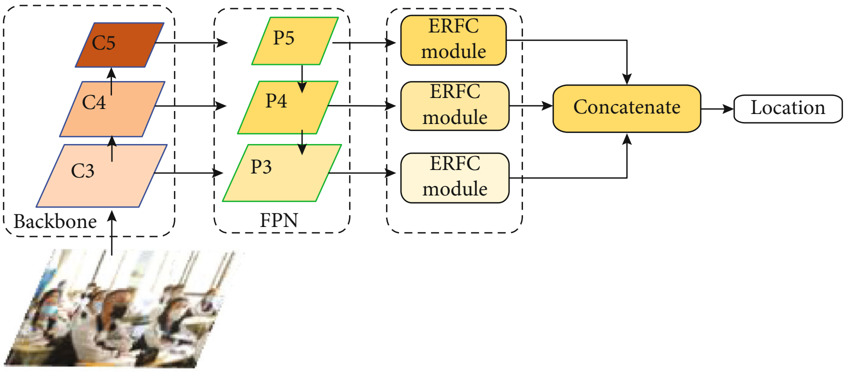


Figure 5: The DRFL network architecture.

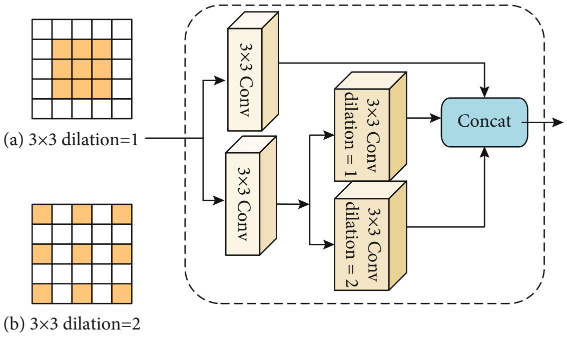


Figure 6: The ERFC module with different dilation rates.

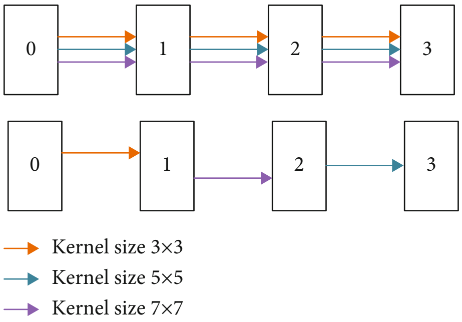


Figure 7: The search strategy in this paper.

training, the branch with the largest weight is retained. The working principle of the search is shown in Figure 7.

Specifically, in this paper, we designed the SRNet20 network based on ResNet18 network, the convolution kernel of 3×3 is replaced by the parallel structure of 3×3, 5×5, 7×7, and the NAS method is used to find the most suitable branch of the task. Then, in the experimental part, we train the searched classification network on our own dataset and compare the results with the original ResNet results on the dataset.

3.4. Model Training Method of Federated Learning. In this paper, the dataset is divided into 10 disjoint parts, representing 10 independent clients, which simulates the real situation of training the classification network. Client C_i has a local private dataset D_i, and model M_0 is published from the central server.

The steps in the training stage are as follows:

(1) Client C_i receives model M_0 from the central server

(2) Client C_i trains the model based on the local dataset D_i and obtains a new model M_i

(3) Client C_i calculates the model parameter difference $M_{\Delta i}$, where $M_{\Delta i} = M_i - M_0$, and uploads the parameter difference $M_{\Delta i}$ to the central server

(4) The central server aggregates the parameter differences uploaded by users, updates the model M_0,

(a) Unmasked face

(b) Masked face

Figure 8: Some images in our dataset.

and resends it to clients participating in federated learning

After a round of update is completed, we check whether the accuracy of the local model meets the requirements. If it meets the requirements, stop training; otherwise, prepare for the next round of training.

4. Experiments and Results

4.1. Dataset. First, the general face location network uses the public Wider Face [24] dataset. It is a benchmark dataset in the field of face detection. It contains 32,203 images and a total of 393,703 annotated faces, of which 158,989 annotated faces are in the training set and 39,496 are in the validation set. Each subset contains 3 levels of detection difficulty: easy, medium, and hard. These different faces have a wide range of changes in terms of scale, posture, illumination, expression, and occlusion. Using this dataset to train the DRFL network will have better detection and location capabilities for faces of different scales.

Second, the masked face classification network is trained on self-made dataset. The training set contains 18,000 images, including 9,000 faces with masks and 9,000 faces without masks. The test set contains 1,751 images, including 656 faces wearing masks and 1,095 faces without masks. The dataset contains face data of different ages, genders, and orientations, which can prevent the network from overfitting the data of a single pose and improve the generalization ability of the network. Some images are shown in Figure 8.

4.2. Loss Function. Based on the loss function of RetinaFace [25], the feature pyramid is adopted to realize the fusion of multiscale information, which plays an important role in the detection of small faces. Its multitask loss function for any training anchor i is shown in the following equation.

$$L = L_{cls}(p_i, p_i^*) + \lambda_1 p_i^* L_{\text{box}}(k_i, k_i^*) + \lambda_2 p_i^* L_{pts}(q_i, q_i^*). \quad (1)$$

There are three parts of the loss function:

(1) Face classification loss $L_{cls}(p_i, p_i^*)$, where p_i is the predicted probability of anchor i which has a face and p_i^* is 1 for the positive anchor and 0 for the negative anchor. L_{cls} is the softmax loss for binary classes

(2) Face box regression loss $L_{\text{box}}(k_i, k_i^*)$, where $k_i = \{k_x, k_y, k_w, k_h\}_i$ and $k^*{}_i = \{\mathrm{k}_x^*, \mathrm{k}_y^*, \mathrm{k}_w^*, k_h^*\}_i$ represent the coordinates of the predicted box and groundtruth box in the positive anchor. $L_{\text{box}}(k_i, k_i^*) = R(k_i - k_i^*)$, where R is smooth L1 defined in [26]

(3) Facial landmark regression loss L_{pts}, where $q_i = \{q_{x_1}, q_{y_1}, \cdots, q_{x_5}, q_{y_5}\}_i$ and $q_i^* = \{q_{x_1}^*, q_{y_1}^*, \cdots, q_{x_5}^*, q_{y_5}^*\}_i$ represent the predicted five facial landmarks and groundtruth associated with the positive anchor. The loss is similar to the box centre regression. The loss-balancing parameters $\lambda_1$and λ_2 are set to 0.25 and 0.1

In the face classification network, we use CrossEntropy loss shown in the following equation.

$$L_{CE} = -\sum_{i=1}^{n} p(x_i) \log (q(x_i)). \quad (2)$$

The $p(x_i)$ represents the real label of x_i, and $q(x_i)$ represents the possibility of x_i measured through the network.

4.3. Setup for Experiments

4.3.1. Data Augmentation. When training the deep learning network, the specific operation randomly cropped the image in the mini-batch to 0.8-1.0 times the size of the original image, and at the same time perform a horizontal flip with a 50% probability, and finally use the resize operation to adjust to a uniform size. Before entering the network, normalize each channel of the image.

The images are randomly cropped and randomly flipped to achieve data augmentation, which improves the accuracy and robustness of the model to a certain extent.

4.3.2. Anchors. The DRFL network uses different anchor boxes in different feature pyramid layers from P3 to P5. In the lower feature layer, small-scale anchor points are tiled to capture small facial features. The high feature layer corresponds to a large area in the original image, so large facial features are captured in the high-level feature layer. The sizes of anchors are shown in Table 1.

4.3.3. Optimization Strategy. In the experiment, the optimization strategy for training the network is to use Adam for the first 10 epochs and SGD for the subsequent epochs. At the 20th epoch, the learning rate decays to 0.1 times, and at 40 epochs, it decays to 0.01 times.

4.4. Tests and Results. In order to test the performance of this network, there are the following three experiments. Experiment 1 tests mAP of the DRFL network. Experiment 2 tests the ERFC module with dilation convolution and without dilation convolution. Experiment 3 compares mAP of original ResNet with SRNet (ResNet after NAS) and verifies the feasibility of federated learning.

(1) *Experiment 1.* Train the DRFL to realize face location, and test the results on the Wider Face validation set. The comparison with other algorithms is shown in Table 2

The results show that our network has advantages in the easy part and the medium part of this validation set. The performance of our algorithm is similar to other algorithms and basically meets the actual needs.

(2) *Experiment 2.* In order to verify the effectiveness of the dilation convolution in the ERFC module, using one 3 × 3 convolution kernel with dilation rate 2 to replace two 3 × 3 convolution kernels with dilation rate 1, we train an unreplaced DRFL network on the same dataset as the baseline and compare it with the replaced network. The test results on the Wider Face validation set are shown in Table 3

The results show that the ERFC module using dilation convolution hardly affects performance while reducing 16.7% parameters, and it is suitable for deploying on embedded cameras.

(3) *Experiment 3.* First, select the appropriate classification model. The convolution kernels of SRNet20, which is created by NAS, are shown in Table 4

Comparing the mAP of the searched network and the original network on the face classification dataset is shown in Table 5.

Comparing with the masked face classification accuracy, the SRNet20 is 8.8% higher than ResNet18, and the SRNet50 is 5.4% higher than the original ResNet50, which proves the effectiveness of the NAS for classification network.

Second, in order to verify the feasibility of federated learning, we simulate a total of 10 clients, and n represents the number of clients who really participate in the training. The model is SRNet20 network, and the number of clients and accuracy are shown in Figure 9.

Table 1: Anchor size in DRFL network.

Feature pyramid	Anchor
P3 (80 × 80 × 64)	16, 20.16, 25.40
P4 (40 × 40 × 64)	64, 80.3, 101.59
P5 (20 × 20 × 64)	256, 322.54, 406.37

Table 2: Accuracy on the Wider Face validation set.

Method		Difficulty	
	Easy	Medium	Hard
MTCNN	84.8%	82.5%	59.8%
Face R-CNN	93.7%	92.1%	83.1%
SSH	93.1%	92.1%	84.5%
DRFL (ours)	94.7%	93.0%	84.2%

Table 3: Results on the Wider Face validation set.

Model		Difficulty	
	Easy	Medium	Hard
DRFL (without dilation)	94.7%	93.1%	84.4%
DRFL (with dilation)	94.7%	93.0%	84.2%

Table 4: Kernel sizes of layers.

Model	Layer 1	Layer 2	Layer 3
SRNet 20	3 × 3, 5 × 5, 7 × 7	7 × 7, 3 × 3, 5 × 5	7 × 7, 7 × 7, 7 × 7

Table 5: mAP of the original network and the searched network.

Model	mAP
ResNet18 (pretraining)	90.0%
ResNet50 (pretraining)	93.0%
SRNet20	98.8%
SRNet50	98.4%

The result shows that the model quickly overfits when the number of participating clients is small. As the number of participating clients increases, the accuracy gradually rises. After sufficient training, the results of federated training are shown in Table 6.

Finally, masked face detection in the dense crowd is completed by cascade network. The mAP of the face location is 90.6%, and the mAP of the masked face classification is 98.5%. We input the test images into the cascade network, and the results are shown in Figure 10. The red box represents the person without the mask, and the green box represents the person with the mask. The near faces can be correctly detected even with slight occlusion, but the blurred faces in the distance are still missed, and this is also the direction for future improvements.

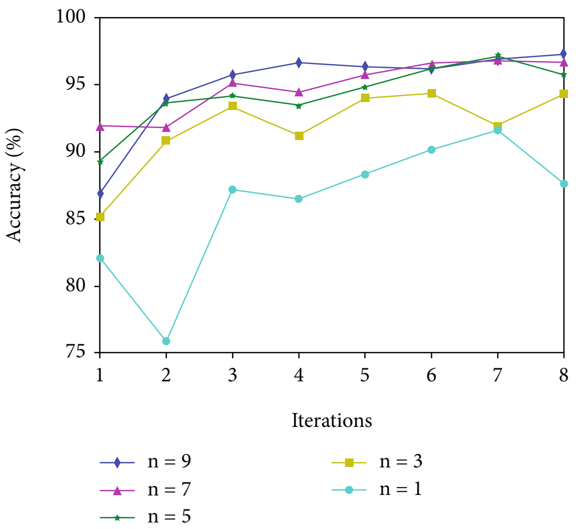


Figure 9: Influence of different number of clients on accuracy.

Table 6: The mAP of centralized training and federated learning.

Method	mAP
SRNet20 (centralized training)	98.8%
SRNet20 (federated learning)	98.5%

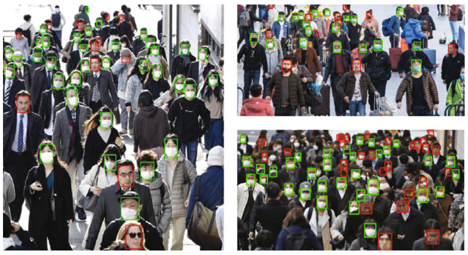


Figure 10: Results of our algorithm.

5. Conclusions

In this paper, we create the DRFL network to implement multiscale face location and create SRNet20 network by NAS to classify masked faces. For privacy protection, we introduce federated learning to provide a joint training solution for multiparty data sources in the real world. By cascading the two networks, the purpose of masked face detection in dense crowds is achieved. From the effect of the test images, our DRFL network has good performance. But for long-distance faces that are blurred or severely occluded, the detection effect needs to be further improved. In the future, we can increase the dataset or adjust the network structure to enhance the network detection robustness. Or we may use a lightweight backbone network to achieve real-time detection in dense crowd scene and apply it to actual life scenarios.

Data Availability

Data is available at https://github.com/woshizr/masked-Face.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by Centre for Public Security Information and Equipment Integration Technology of UESTC (University of Electronic Science and Technology of China). Thanks are due to the computing platform provided by the laboratory.

References

[1] World Health Organization, *COVID-19 weekly epidemiological update, 13 July 2021*, WHO, 2021.

[2] N. H. L. Leung, D. K. W. Chu, E. Y. C. Shiu et al., "Respiratory virus shedding in exhaled breath and efficacy of face masks," *Nature Medicine*, vol. 26, no. 5, pp. 676–680, 2020.

[3] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[4] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks: a survey towards private and secure applications," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, 2016.

[6] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, Venice, Italy, 2017.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, 2014.

[8] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125, Honolulu, HI, USA, 2017.

[9] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[10] H. Wang, Z. Li, X. Ji, and Y. Wang, "Face r-cnn," 2017, https://arxiv.org/abs/1706.01061.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[12] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, "Ssh: single stage headless face detector," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4875–4884, Venice, Italy, 2017.

[13] X. Tang, D. K. Du, Z. He, and J. Liu, "Pyramidbox: a context-assisted single shot face detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 797–813, Munich, 2018.

[14] Y. WANG, X. ZHANG, J. YE, H. SHEN, Z. LIN, and W. TIAN, "Mask-wearing recognition in the wild," *SCIENTIA SINICA Informationis*, vol. 50, no. 7, pp. 1110–1120, 2020.

[15] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, pp. 21–37, Springer, Cham, 2016.

[16] M. Jiang, X. Fan, and H. Yan, "Retinamask: a face mask detector," 2020, https://arxiv.org/abs/2005.03950.

[17] Z. Wang, G. Wang, B. Huang et al., "Masked face recognition dataset and application," 2020, https://arxiv.org/abs/2003.09093.

[18] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering (TNSE)*, vol. 7, no. 2, pp. 766–775, 2020.

[19] Z. Xu and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 38, no. 5, pp. 968–979, 2020.

[20] B. Mcmahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[21] Z. Cai and Z. He, "Trading Private Range Counting over Big IoT Data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, 2019.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

[23] H. Liu, K. Simonyan, and Y. Yang, "Darts: differentiable architecture search," 2018, https://arxiv.org/abs/1806.09055.

[24] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: a face detection benchmark," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5525–5533, Las Vegas, NV, USA, 2016.

[25] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: single-stage dense face localisation in the wild," 2019, https://arxiv.org/abs/1905.00641.

[26] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, Santiago, Chile, 2015.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 7018486, 15 pages
https://doi.org/10.1155/2021/7018486

Research Article

Deep Learning-Based Service Scheduling Mechanism for GreenRSUs in the IoVs

Jitong Li,[1,2] Chao Wang ,[1] Daehee Seo,[3] Xiaoman Cheng,[1] Yunhua He,[1] Limin Sun,[4] and Ke Xiao[1]

[1]*School of Information Science and Technology, North China University of Technology, 100144, China*
[2]*School of Computing, Beijing University of Posts and Telecommunications, 100876, China*
[3]*College of Convergence Engineering, Sangmyung University, 03016, Republic of Korea*
[4]*The Institute of Information Engineering, Chinese Academy of Science, 100093, China*

Correspondence should be addressed to Chao Wang; wangchao.andy@gmail.com

Received 24 June 2021; Revised 6 August 2021; Accepted 6 September 2021; Published 30 September 2021

Academic Editor: Yan Huo

Green roadside units (RSUs), also called renewable energy-powered RSUs, are utilized recently rather than the traditional electric-powered RSUs with high power consumption and the large infrastructure deployment cost in the Internet of vehicles (IoVs). However, the power of the green RSUs is limited and unstable, which is affected by the battery size and charging environment. Therefore, a big challenge to deploy green RSUs in the IoVs is to schedule their service process properly, in order to extend the service efficiency of RSUs. In this paper, a deep learning-based communication scheduling mechanism is proposed regarding the service scheduling problem. In particular, a three-part scheduling algorithm consisting of RSU clustering, deep learning-based traffic prediction, and a vehicle access scheduling algorithm is presented to maximize the service number of vehicles and minimize the energy cost. An extensive simulation is done, and the simulation results indicate that our algorithm can serve more vehicles with less energy consumption compared with other scheduling mechanisms under different scenarios.

1. Introduction

The Internet of vehicles (IoVs) is composed of lots of vehicles and RSUs. These vehicles are equipped with on-board units (OBUs) including a global positioning system (GPS) and the RSUs are connected to the centralized network or service providers [1]. In the IoVs, each vehicle can communicate with each other at the ad hoc style with the DSRC technique or Pc5 interface of LTE-V. This kind of communication is referred as vehicle-to-vehicle (V2V) communication. Besides, these vehicles can also communicate with RSUs with the interface of LTE-V [2] and this kind of communication is commonly referred to vehicle-to-infrastructure (V2I) communication. Based on the two styles of communication, most of services in the IoVs can be carried out, such as location services [3], blockchain services [4], social services [5], and crowd sensing services [6].

In the IoVs, the RSUs play important roles. They not only extend the service range of the IoVs but also provide high-speed message forwarding services for vehicles with high bandwidth links to the service providers [7]. However, the forwarding services require high power consumption [8]. In addition, the RSUs are powered by physical infrastructures linked to the electrical grid, which costs too much. Especially in some rural areas, the cost of deploying electrical facilities for RSUs may be much higher. At last, traditional electric-powered RSUs cause carbon dioxide emissions indirectly. Therefore, the requirement of renewable power, such as being polar and windy, to be integrated into RSUs instead of the traditional powered functions is proposed by several papers [9].

However, there are some critical problems to be solved while deploying the renewable energy-powered RSUs. First, the power on the RSU is limited due to the battery size and charging environment. In addition, the power consumption of communication is influenced by the communication time, communication frequency, communication distance, and channel quality. Without a proper communication scheduling strategy, the power may be exhausted quickly, which will have a serious impact on the normal development of services in the IoVs. Thus, this paper focuses on how to maximize the performance of renewable energy-powered RSUs in order to serve as many vehicles as possible with less energy consumption.

Motivated by this, a deep learning-based communication scheduling mechanism is proposed in this paper, composed of three parts. First, a clustering method named t-SNE is utilized to classify all the RSUs to several clusters, each of which has a special evolving rule on traffic flows. Second, we use a deep learning-based algorithm to predict the future traffic flow for each RSU cluster, with the history knowledge of the traffic status under each RSU. At last, once the prediction results are obtained, a service ability scheduling algorithm and a priority-based RSU access algorithm are proposed to schedule the vehicle access order. The contributions of this paper are outlined as follows:

(1) A clustering method t-SNE is utilized, based on which the future traffic flow for each RSU can be a prediction with a deep learning algorithm

(2) Based on the future traffic prediction, we propose a service ability scheduling algorithm and a priority-based RSU access algorithm to maximize the RSU service efficiency

(3) An extensive simulation is deployed to demonstrate that the proposed algorithm can achieve higher efficiency with less energy cost compared with other scheduling mechanisms under different scenarios

The rest of the paper is organized as follows. Section Related Work summarizes the related work. Section System Model presents the system model and problem formulation. Section Deep Learning-Based Communication Scheduling Algorithm details the deep learning-based communication scheduling mechanism. And section Simulations presents the simulation results. Lastly, section Discussion and Conclusion draws the conclusion.

2. Related Work

It is necessary to reduce energy consumption as much as possible for RSUs in the IoVs, and several papers have done some work referring to the grid-powered RSUs. In [10], the authors focus on the hardware design of RSUs and propose a centralized control module to schedule the energy cost on RSUs. Their idea is based on the artificial neural network and is composed of three algorithms to achieve the purpose of the green scheduler. In [11], the authors incorporate a sleep mechanism on grid-powered roadside units to further reduce power consumption. However, it is not fit for the renewable energy-powered RSUs. The off-grid wind-powered RSUs are considered in [12]. Varied from conventional reliability analysis, the reliability in this paper is redefined in a highway environment. Then, the energy models are developed and the minimum battery size can be determined when a certain reliability standard and quality of service are achieved.

Due to the high cost of grid-powered RSUs, the renewable energy-powered RSUs are proposed. As the energy on this kind of RSUs is limited, how to minimize the energy consumption is also a wide concern. An energy-efficient scheduling framework is proposed in [13]. The task scheduling and energy consumption are considered jointly in the proposed heuristic algorithm. To minimize energy consumption and satisfy task latency constrains, an imitation learning-enabled online task scheduling algorithm is proposed in [14]. In [15], it is assumed that any vehicle's position in the network is strongly deterministic. Then, three energy-efficient online traffic scheduling algorithms are introduced to minimize the long-term power consumption subject to the communication requests associated with the passing vehicles.

The scheduling problem in renewable energy-powered RSUs is also a hot research topic. A joint scheduling and power control scheme is proposed in [16]; it is formulated as a mixed-integer nonlinear programming (MINLP) problem. Paper [17] investigates the problem of scheduling the downlink communication for renewable energy-powered RSUs toward vehicles, with the objective of maximizing the number of served vehicles. In [18], the authors propose low-complexity algorithms for downlink traffic scheduling in green vehicular roadside infrastructure. However, the algorithms require some priori information, which may not be always available in the IoVs. In [19], a reinforcement learning technique for optimizing downlink scheduling is proposed in an energy-limited vehicular network. Its objective is to equip RSUs with the required artificial intelligence to realize an optimal scheduling policy that will guarantee the operation of the vehicular network during the discharge cycle while fulfilling the largest number of service requests. However, works [17, 19] just simulate the scenario in the simulator without testing the algorithms with vehicle trajectories of reality.

Moreover, some works have been done from the perspective of services to save energy of green RSUs. In a green communication scenario, the safety and QoS are both a concern in [20]. A deep reinforcement learning model named deep Q-network is proposed, which learns an energy-efficient scheduling policy. With the policy, the battery of an RSU is extended and the safety of environment is promoted while the quality of service (QoS) levels is met. Aiming at the problem of power deficiency in solar-powered roadside units (SRSUs), the challenge of QoS loss is addressed in [21], in which a two-phase approach is proposed. With the purpose of energy consumption and time delay guarantee, a distributed packet scheduling optimization strategy is proposed in [22] for the renewable energy-powered RSUs. Based on the proposed strategy, the system energy consumption can be minimized and the delay of the system can also be reduced, in which an optimization model is established based on the Lyapunov theory. To solve

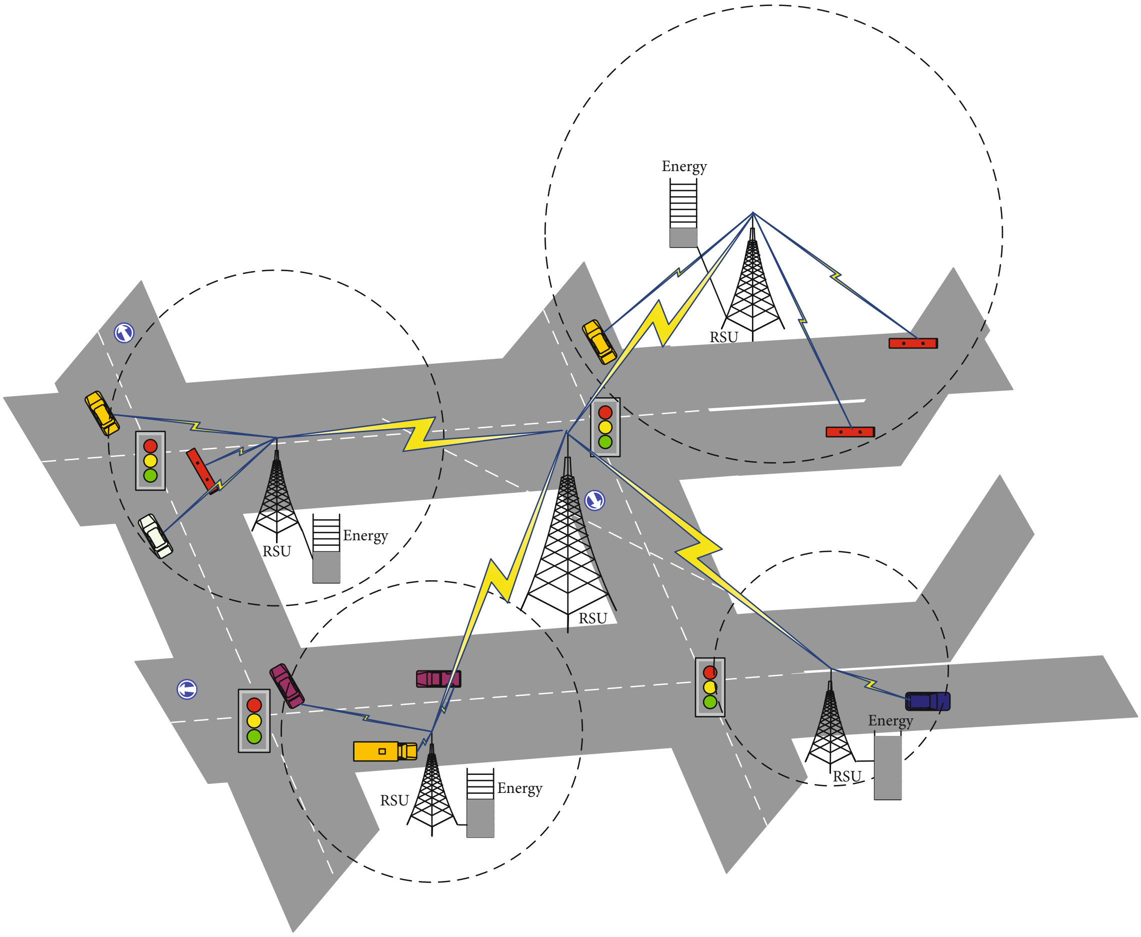


Figure 1: Solar-powered RSUs in the scenario of IoVs.

the problems that existed in battery-enabled RSUs and electric vehicles (EVs), an intelligent energy-harvesting framework is constructed in [23], where RSUs and EVs are integrated. Based on a three-stage Stackelberg game developed by the authors, the utilities of RSUs and EVs are maximized.

3. System Model

3.1. Scenario Description. We consider an Internet of vehicles scenario, shown as Figure 1. In the scenario, all the vehicles are equipped with a DSRC or LTE-V communication module, with which each vehicle can interact with the ITS service center. Besides, any vehicle accessing to the centralized network must connect to an RSU with its communication module. Due to consideration of energy saving, more and more RSUs are powered by renewable energy such as solar energy. Thus, the energy on RSUs is limited and each RSU can only serve finite number of vehicles. Thus, in this paper, we focus on the communication scheduling mechanism of RSUs to maximize the number of service vehicles.

3.2. Problem Formulation. Based on the above scenario, we construct the system model as an undirected graph $G(R, E, V)$, shown in Figure 2. Figure 2 is a formalized definition of the described scenario. RSUs in Figure 1 are mapped to the vertexes in Figure 2, and the communication links among RSUs are described as the edges between vertexes. Graph G denotes the Internet of vehicles scenario and the details of the network are described as follows:

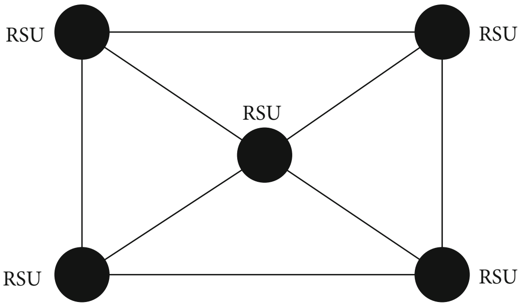


Figure 2: System model.

(1) R is the set of RSUs, denoted as $R = \{r_1, r_2, \cdots, r_n, \cdots, r_M\}$. In the network, there are$M$ RSUs and everything can be accessed by vehicles passing by. With the assumption that each RSU is powered by renewable energy, the communication bandwidth of each

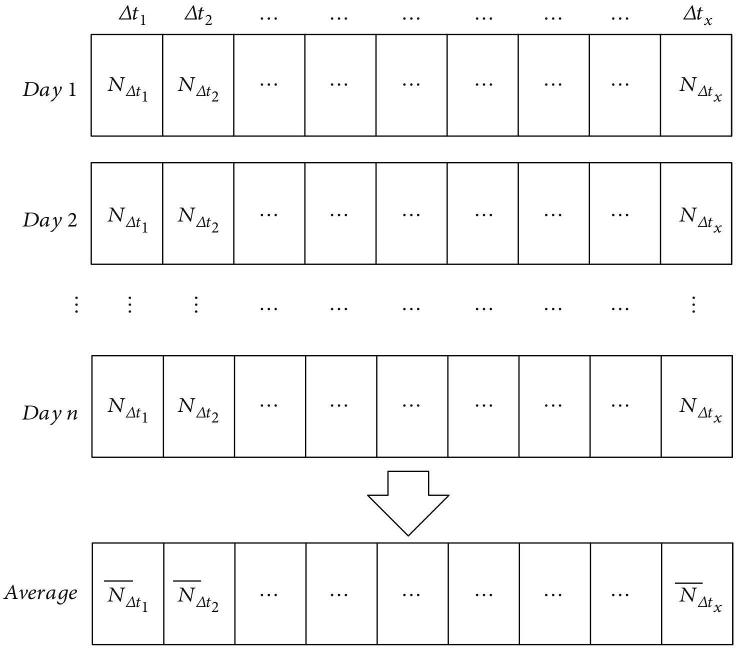


FIGURE 3: Data preprocessing.

RSU can be tuned with time. In this paper, S_{r_n} is the service ability of RSU $r_n (r_n \in R)$ and its value is determined by the energy charged, with $\max_{r_n}$ being the maximum value of S_{r_n} determined by the battery size. Here, we assume that S_{r_n} is an integer and it denotes the number of vehicles that r_n can server. For any RSU r_n, each communication with a vehicle will cost some service ability of r_n, which is assumed to be 1 in this paper. If the time period is considered, then S_{r_n} can be rewritten as $S_{r_n}^{\Delta t}$, which means the service ability of r_n at the time period Δt. Besides, in order to maintain the service ability at a time period, some energy still will be costly. In other words, if there is no vehicle accessing, the RSU still spends service ability on maintaining the communication service. Here, another assumption is made that if no vehicle uses the allocated service ability in the time period, the service ability will be wasted to maintain the service, and the amount of service ability is the same as the communication cost to vehicles

(2) An edge is denoted as $E_{ij} (E_{ij} \in E)$ that connects two RSUs r_i and r_j. It means that r_i and r_j are adjacent RSUs and a vehicle can move from the coverage range of r_i to that of r_j

(3) V is the set of vehicles, denoted as $V = \{v_1, v_2, \cdots, v_i, \cdots, v_N\}$. For any vehicle $v_i \in V$, it moves from the coverage of one RSU to another RSU, and at any time, it can only communicate with no more than one RSU. The trajectory of vehicle v_i is L_{v_i}, which is a location sequence, such as $(r_1^{\Delta t_1}(v_i), r_2^{\Delta t_2}(v_i), r_3^{\Delta t_3}(v_i), \cdots, r_n^{\Delta t_x}(v_i))$. Here, $r_n^{\Delta t_x}(v_i)$ means that vehicle v_i at time period Δt_x is covered by RSU r_n. Besides, we define another term $S_{r_n}^{v_i}(\Delta t_x)$. If $S_{r_n}^{v_i}(\Delta t_x)$ is 1, RSU r_n serves vehicle v_i at time period Δt_x successfully; otherwise, $S_{r_n}^{v_i}(\Delta t_x) = 0$

In this scenario, we focus on the communication scheduling problem on each RSU to optimize the network. The objective is to find a proper combination $S_{r_n}^{v_i}(\Delta t_x)$ for each RSU and vehicle for each time period that maximizes the service time. The problem is formulated as follows.

$$\text{Maximize throughput} = \frac{\sum_{v_i \in V} \sum_{\Delta t_x \in T} \sum_{r_n \in R} S_{r_n}^{v_i}(\Delta t_x)}{\text{total communication}}, \quad (1)$$

$$\text{Maximize service rate} = \frac{|\text{serviced vehicles}|}{|V|}, \quad (2)$$

subject to

$$\text{Service vehicles} = \left\{ v \mid v \in V, S_{r_n}^{v}(\Delta t_x) = 1, \exists \Delta t_x \in T, r_n \in R \right\}, \quad (3)$$

$$\sum_{\Delta t_x \in T} \sum_{v_i \in V} S_{r_n}^{v_i}(\Delta t_x) \le \min\left(B_{r_n}^{\Delta t_x}, \max_{r_i}\right), \quad \text{for } \forall r_n \in R, \quad (4)$$

$$r_n^{\Delta t_x}(v_i) \in L_i, \quad \forall S_{r_n}^{v_i}(\Delta t_x) = 1. \quad (5)$$

Formula (1) means that given a proper service scheduling set for each RSU and vehicle, with the limited energy in RSUs, maximize the successful communication rate between vehicles and RSUs. Formula (2) is used to maximize the ratio between the number of vehicles that can communicate with RSUs and the total number of vehicles. Constraint

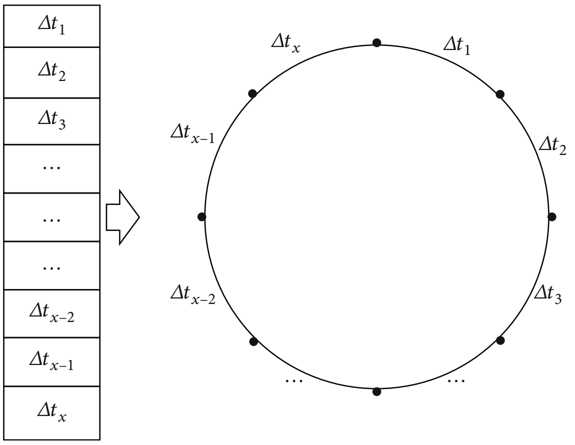


Figure 4: Method of sliding window.

```
Input:
$S$,Service ability kept currently
$d$,Length of features
$M$,Traffic prediction model
$N = \{N_{\Delta_{t_1}}, N_{\Delta_{t_2}}, \cdots, N_{\Delta_{t_{n-1}}}\}$,Traffic status sequence
Output:
$S_{\Delta t_n}$, Energy Distribution for next time slot
$N_{pre} = \{N_{\Delta t_n}, N_{\Delta t_{n+1}}, \cdots, N_{\Delta t_{last}}\}$
for $N_{\Delta t_i} \in N_{pre}$ do
    $N_{\Delta t_i} = TrafficPre(M, N_{\Delta t_{i-d}}, N_{\Delta t_{i-d+1}}, \cdots, N_{\Delta t_{i+1}})$
endfor
$S_{\Delta t_n} = N\Delta t_n / \sum_{N_{\Delta t_i} \in N_{pre}} N_{\Delta t_i} \cdot S$
Return $S_{\Delta t_n}$
```

Algorithm 1: Service Ability Distribution Algorithm.

(3) ensures that for each RSU, the energy cost for its communication with vehicles should be no more than the energy it is charged. Constraint (4) ensures if an RSU can communicate with a vehicle; the vehicle should be in the RSU's coverage range.

3.3. Main Idea. In order to maximize the service number of vehicles and the throughput, all the vehicles in the network should be connected to an RSU at any time, and then, the service number is maximum intuitively. Thus, we should figure out the number of vehicles in the network, based on which the total energy required to satisfy the communication between all vehicles and RSUs can be calculated.

As the energy cost of each RSU is relative to the number of vehicles covered by each RSU, the first step is to predict the number of vehicles passed by each RSU. Based on this, the required energy of each RSU can be obtained. However, the amounts of energy in each RSU are not the same, which are determined by surrounding environments. When the energy of an RSU cannot afford the energy cost on the vehicles passed by, a scheduling algorithm should be designed. A reasonable communication scheduling algorithm should be designed to save energy or reduce energy waste, especially when the energy of an RSU cannot afford the energy cost on communications. The main idea of the algorithm is to tune energy cost over all time periods and then to select more vehicles to access.

```
Input:
$S$,Service ability allocated for the current time slot
$V = \{v_1, v_2, \cdots, v_n\}$,Vehicle set in the current time slot
$D = \{D_1, D_2, \cdots, D_n\}$,Destination of each vehicle
$F = \{F_1, F_2, \cdots, F_n\}$,Source of each vehicle
Output:
$L$,Service order list
$P = \varnothing$,Initial priority set
for $v_i \in V$ do
    $p_{v_i} = Priority(v_i, D_i, F_i)$,Compute the service priority
    add $p_{v_i}$ to set $P$
endfor
repeat
    Add the maximum value in $P$ to $L$
until $|L| > S$
Return $L$
```

Algorithm 2: Priority-based RSU access algorithm.

4. Deep Learning-Based Communication Scheduling Algorithm

4.1. Deep Learning-Based Traffic Prediction Mechanism. As Section 3 shows, the energy on each RSU is limited and each time of communication between an RSU and a vehicle costs energy. In this subsection, our concern is on how to predict the future traffic according to the current traffic status for each RSU. Here, we use the LSTM model to predict future traffic. Compared with the RNN model, LSTM can tackle a longer sequence better to avoid the affection of short memory. In order to make the traffic data fit the LSTM model, the traffic data should be preprocessed first. However, due to the sparsity of the training data of each RSU and the similarity of traffic-evolving rules on many RSUs, we cluster RSUs to several classes using the clustering method. At last, we use the LSTM model to train the traffic-evolving data in different classes to get a prediction model for each cluster of RSUs. Next, we will introduce the deep learning-based traffic prediction mechanism in the following parts.

4.1.1. Data Preprocessing Method. Usually, the traffic flow changes with the time and reveals similar evolving rules day by day. Thus, we focus on every day as an independent object to study the rules of traffic flow.

First, a day can be divided into many equal time slots Δt, such as an hour or a quarter.

For each RSU, the amount of traffic flow in each time slot forms a value sequence for a day, shown as Figure 3, where $N_{\Delta t_x}$ means the total number of vehicles at time slot Δt_x covered by an RSU. However, there may be noise in the data for some days, which is not the common phenomenon. For example, a traffic accident may cause a fast rising of the traffic flow under an RSU, which happens very

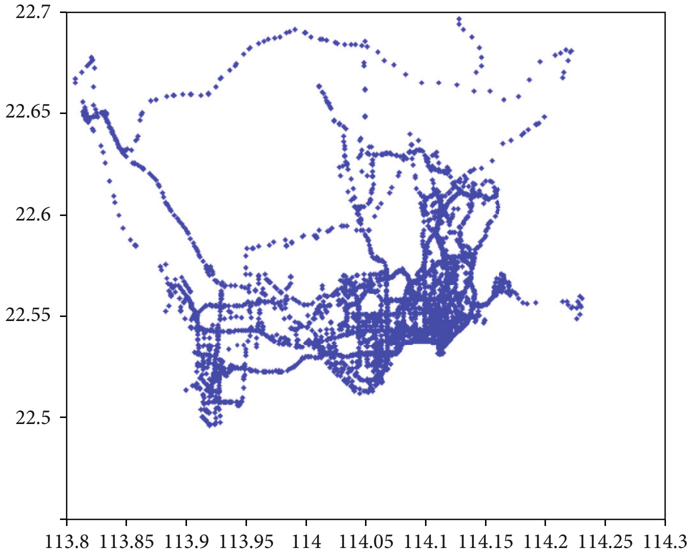


Figure 5: Trajectory snapshot.

Table 1: Simulation scenario description.

Area size	50 km × 50 km
Duration	8 days
Number of RSUs	1536
Coverage of RSUs	1 km × 1 km
Number of trajectories	975391

occasionally. Therefore, we get the average traffic of the same time slot on each day, shown as the bottom of Figure 3, which can reveal the original traffic status for each slot.

4.1.2. RSU Clustering Method. There are three advantages for RSU clustering. First, the amount of RSUs in a city is usually linearly related to the size of the city and there may be thousands of RSUs in a city. If we train a model for each RSU, it will cost too much computing resource, which is also not necessary. In addition, it is found that traffic flows on many RSUs reveal some similar evolving patterns, which can be tackled together. At last, it can solve the data sparsity problem as there may not be enough data to be trained for some RSUs in practice. When some RSUs are clustered into one class, the data on this class of RSUs can be shared together, so that the training set for this cluster of RSUs is extended.

After data preprocessing, the evolving rules for a cluster of RSUs are denoted as a sequence $(\bar{N}_{\Delta t_1}, \bar{N}_{\Delta t_2}, \bar{N}_{\Delta t_3}, \cdots, \bar{N}_{\Delta t_x})$. Here, this sequence is used as the feature vector for model training. As mentioned before, the time slot Δt can be chosen as a quarter, an hour, or longer. In this paper, it is needed to catch the evolving rule from a microscopic time perspective but the length of the sequence cannot be large considering the computing consumption. Besides, there should be obvious up or down trending for the sequence value; rather than that, most of the values in the sequence are closed to zero. Thus, Δt is set to a quarter, so that the sequence length, namely, the number of dimensions of the feature vector, is $60/15 \times 24 = 96$.

So far, each RSU has a special feature vector, which will reveal individual characteristic. Next, all these feature vectors are used as the input source to cluster all the RSUs. But, it is hard to get a good result if we directly cluster such high-dimensional feature vectors with general clustering methods [24]. In order to solve the problem of high-dimensional data clustering, many approaches have been proposed. Clustering with a subset and fuzzy clustering are typical representatives of these methods [25, 26]. Moreover, the deep learning-based clustering algorithm is also used to cluster high-dimension data, in which a deep learning strategy is used to learn low-dimensional representation of high-dimensional data. Then, the ordinary clustering algorithm can be used after the low-dimensional representation is obtained. Although a better clustering effect can be obtained using these clustering methods, they cluster high-dimensional data in an abstract way and there are still challenges in data visualization and clustering rationality.

T-distributed stochastic neighbour embedding (t-SNE) is a new dimension reduction and visualization technique for high-dimension data [27], which is used to reduce the dimensionality of each RSU's traffic flow so that all RSUs can be clustered well. T-SNE is developed from SNE that has a "crowding problem." Different from SNE, the Euclidean distance is converted to joint probability to express the similarity between data points in t-SNE (Euclidean distance is converted to conditional probability in SNE). As shown in equations (6) and (7), p_{ij} and q_{ij} are the joint probability of the original high-dimensional data and low-dimensional data after dimensionality reduction, respectively.

$$p_{ij} = \frac{\exp\left(-\left\|x_i - x_j\right\|^2 / 2\sigma^2\right)}{\sum_{k \neq l} \exp\left(-\left\|x_k - x_l\right\|^2 / 2\sigma^2\right)}, \tag{6}$$

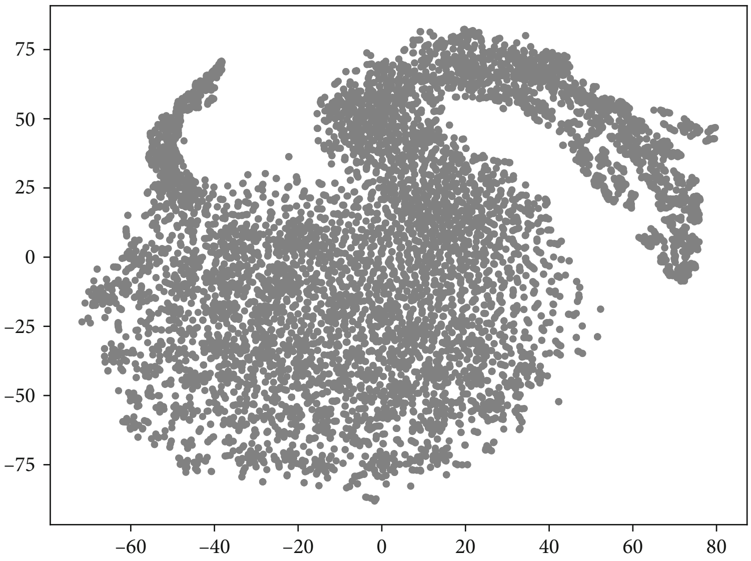


Figure 6: The visualization of each RSU feature vector.

$$q_{ij} = \frac{\exp\left(-\left\|y_i - y_j\right\|^2\right)}{\sum_{k\neq l}\exp\left(-\|y_k - y_l\|^2\right)}. \tag{7}$$

The difference in the joint probability distribution of the original data and the data after dimensionality reduction will be very small, even $p_{ij} = q_{ij}$. Based on this idea, Kullback-Leibler (KL) divergence is used to measure the difference between two distributions, shown in equation (8), where P is the joint probability distribution in the high-dimensional space (the original data), Q the joint probability distribution in the low-dimensional space (the data after dimensionality reduction), and C the cost or the sum of KL divergence difference value between two distributions. The smaller C is, the better the dimensionality reduction effect is.

$$C = KL(P\|Q) = \sum_i\sum_j p_{ij}\log\frac{p_{ij}}{q_{ij}}. \tag{8}$$

As shown in equation (9), for the purpose of minimizing C or the sum of the KL divergence, the gradient descent method is used so that a better dimensionality reduction effect can be obtained. Finally, the desired low-dimensional data will be obtained by continuous iterative solutions.

$$\frac{\delta C}{\delta y_i} = 4\sum_j\left(p_{ij} - q_{ij}\right)\left(y_i - y_j\right). \tag{9}$$

By using t-SNE, the low-dimensional representation of the traffic flow in each RSU can be obtained. Then, the K-means algorithm can be used for clustering RSUs, in order to determine a proper K value. The elbow method is used to fix the best number of clusters K.

Its main idea is to choose the optimal K according to the varying trend of the sum of squared errors (SSE) in a cluster. SSE is defined as equation (10).

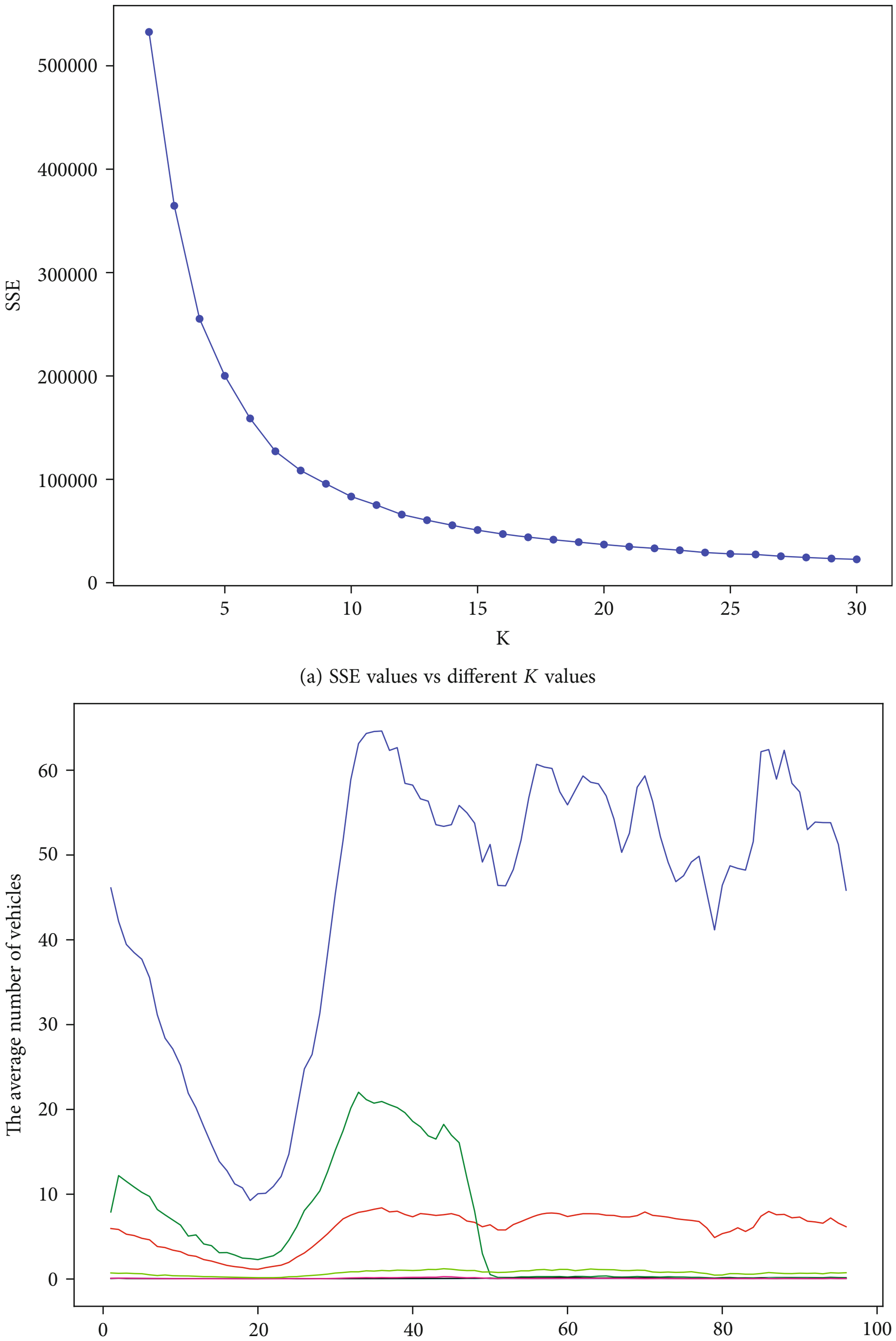
$$SSE = \sum_{i=1}^{k}\sum_{p\in C_i}|p - m_i|^2. \tag{10}$$

Here, C_i is the ith cluster, p is the sample vector of C_i, and m_i is the center of cluster C_i.When K is smaller than the actual cluster number, the value of SSE descends fast with K increasing. However, when a proper K is found, the value of SSE tends to be stable with K growing. Therefore, the inflection point is the best K value.

4.1.3. Deep Learning-Based Traffic Prediction. Once all the RSUs are clustered, the future traffic flow should be predicted based on the history data of RSUs. As real-time prediction is hard to achieve, we use the LSTM model to train the dataset offline.

First in the learning stage, we use the vectors from each RSU class as the training dataset individually. However, if the 96-dimension features of the vector are used as the training set, it will cost too much computing resource. Intuitively, the current traffic conditions are usually affected by the last few hours of traffic conditions. Thus, we use a method called the sliding window with size n to cut off the feature vector. So, in the learning stage, the consecutive n elements in a vector compose the feature sequence and the following element is the tag.

In the prediction stage, the consecutive n elements in a vector are used to predict the following element, shown as the left part in Figure 4. Another problem is how to predict the first n elements. As the traffic flow is changing periodically, the vector can be transformed to a clock-like vector, shown as the right part in Figure 4. For example, to predict the first element, we can use the last n elements of the vector as input.



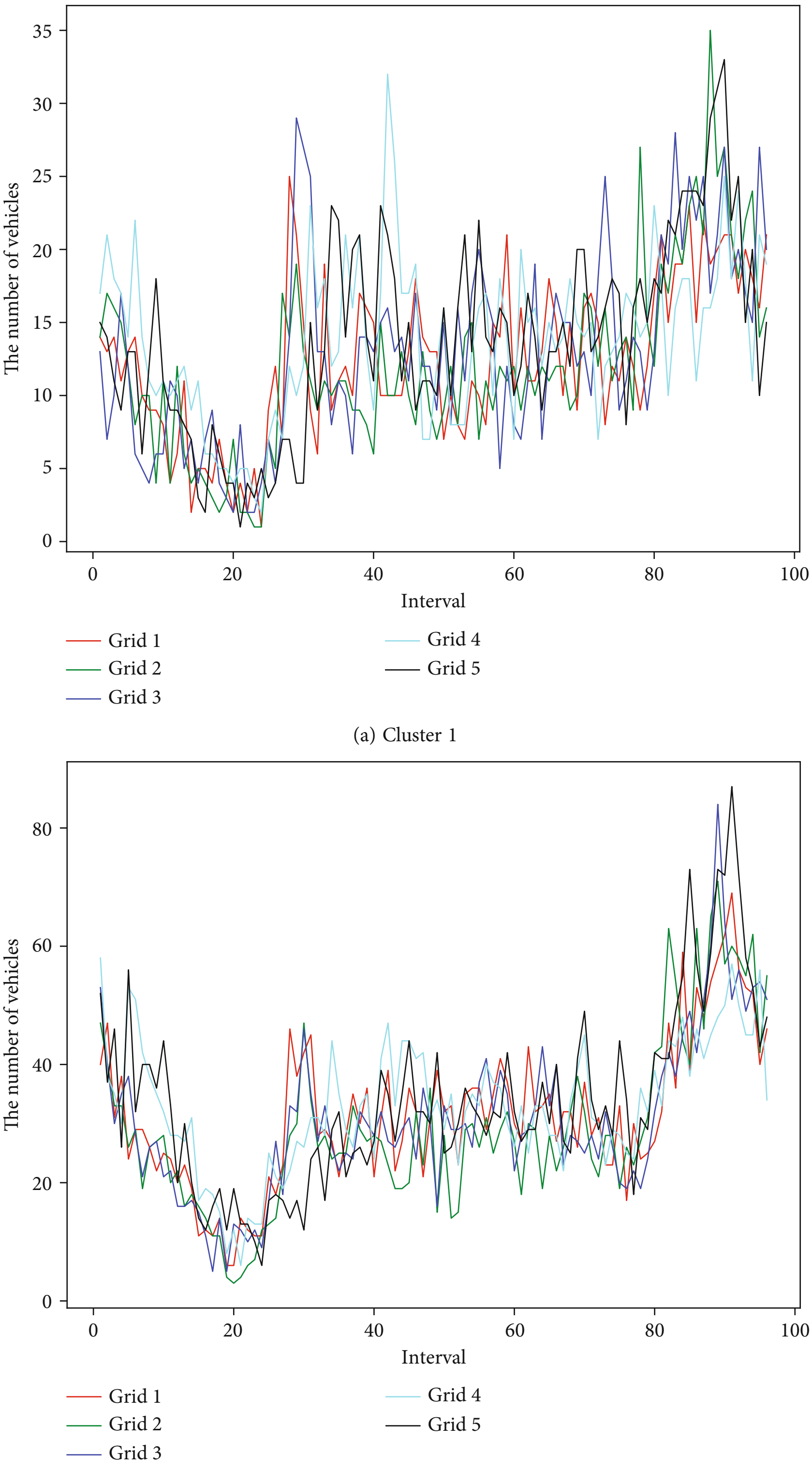
(a) SSE values vs different K values

(b) Clustering result of $K = 7$

Figure 7: Result of clusters.

4.2. Scheduling Algorithm. There are two parts in the RSU communication scheduling algorithm, namely, service ability distribution algorithm and the priority-based RSU access algorithm. When a new period starts, such as end of daytime, the energy of each RSU has already been charged during daytime. From then on, the energy is continually discharging until the next period. We assume that the energy charged during daytime will be used in the next period, rather than in the current period. Therefore, the energy should support the RSU to work for the entire period. Without a proper scheduling, the energy might be exhausted quickly, so that the vehicles traveling during the end of the period may not be served no matter its priority.

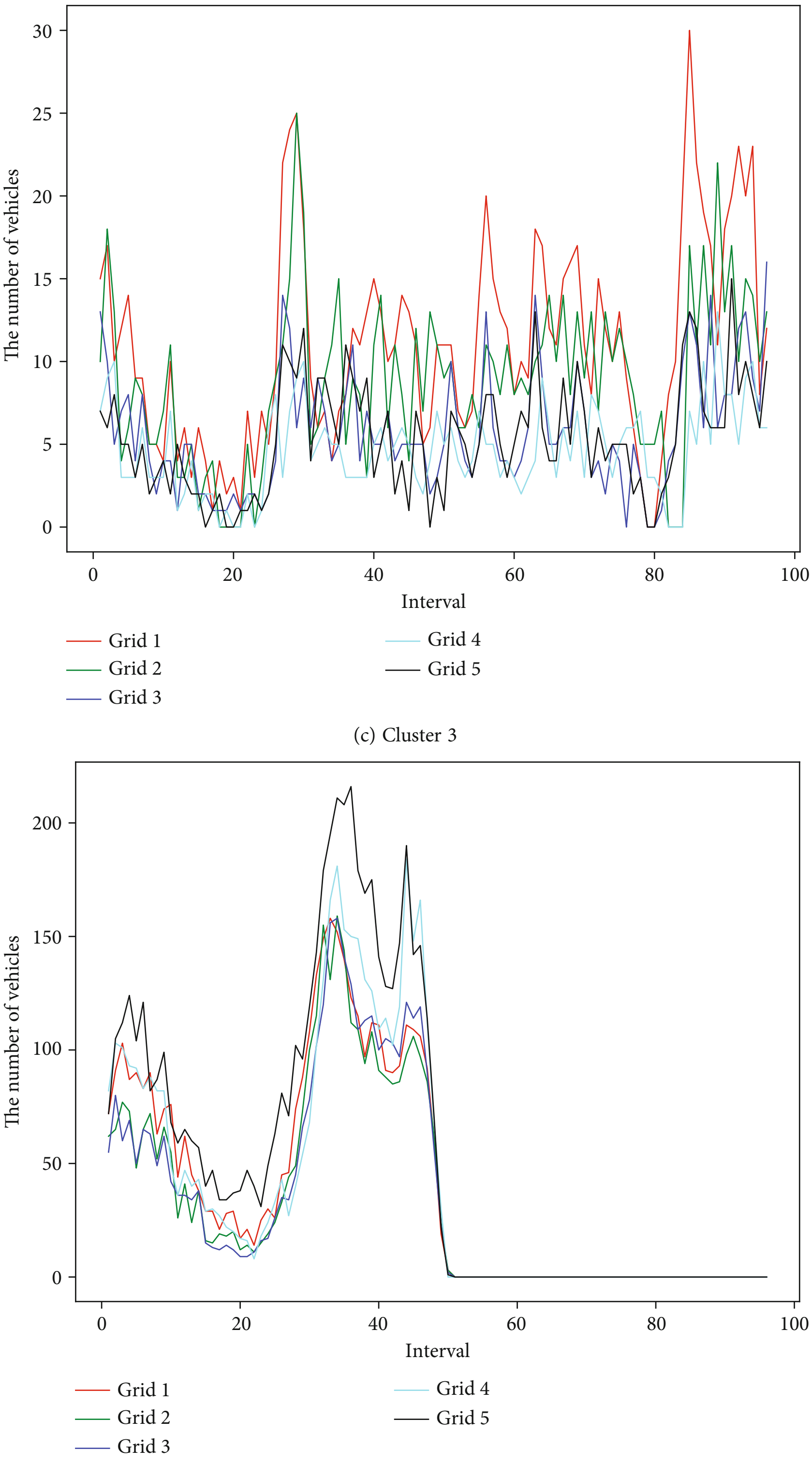
Therefore, we propose the service ability distribution algorithm to schedule the energy distribution for each RSU. The main idea of the Algorithm 1 is sketched below. First, when the energy harvested during daytime is determined, the total service ability in this period of the RSU will



(a) Cluster 1

(b) Cluster 2

Figure 8: Continued.



(c) Cluster 3

(d) Cluster 4

Figure 8: Examples of clusters.

be determined, too. Next, we will assign the total service ability to each time slot according to the predicted number of vehicles in each time slot. In order to predict the number of vehicles, we use the above traffic prediction algorithm to get the traffic flow from the following slot to the ending slot of the period. Then, the service ability for each time slot is assigned according to the total service ability multiplied by the ratio of the current slot traffic flow to the total traffic flow

Table 2: Loss for each cluster under various feature vector lengths.

Class	Length		
	5	10	15
1	3615	3608	3594
2	836	835	834
3	525133	439265	436616
4	27915	27420	27247
5	1282	1282	1282
6	857	857	856
7	63173788	60481962	59170782

Table 3: The average MSE loss of different models ($\times 10^4$).

L	N				
	16	32	64	128	256
1	6.856	5.958	4.453	4.159	4.149
2	6.973	5.761	4.436	4.438	4.146
3	8.71	6.409	5.121	4.779	5.132
4	9.945	7.108	5.433	6.115	5.375
5	12.049	8.158	6.573	7.048	6.394

Table 4: The prediction error under different batch sizes.

B	C				
	16	32	64	100	200
1	4.429	4.429	**4.429**	4.43	4.43
2	4.267	4.132	4.018	3.921	**3.838**
3	**7.124**	9.693	11.752	13.366	14.736
4	14.454	14.235	14.051	13.891	**13.767**
5	12.779	11.944	11.227	10.607	**10.063**
6	9.976	9.892	9.809	9.729	**9.65**
7	**12.739**	15.575	18.252	20.728	23.129

of the period. Once the service ability on each time slot for an RSU is assigned, it is necessary to determine the service vehicles and the service order.

Here, we propose a conception of service priority, defined as formula (11) (where we assume that the destination of each vehicle is known).

$$\text{Service priority} = \frac{1 + \text{original distance}}{1 + \text{destination distance}}. \qquad (11)$$

It means that the longer the time that the vehicle has travelled, the higher the priority it has, because it may have more data to submit. Besides, if a vehicle has a long way to its destination, it may have more chances to communicate with RSUs, so that the priority is small. Here, the distance can be Euclidean distance or Manhattan distance. Based on this algorithm, when several vehicles are in the coverage of an RSU. The RSU calculates the priorities of all vehicles and provides service in the descending order of the priority, shown as Algorithm 2.

5. Simulations

5.1. Simulation Scenario. We evaluate our mechanism in a scenario with real vehicle trajectories, which come from the taxis in Shenzhen city, and a snapshot of the taxi location at a time point is shown in Figure 5. In this scenario, we assume that all the RSUs are uniformly distributed with the coverage range of 1 km × 1 km. Then, it can be considered that any area in Shenzhen is covered by an RSU. The parameters of the scenario are shown in Table 1.

The simulation is composed of three parts, namely, RSU clustering, traffic flow prediction, and RSU communication scheduling process. First, in the RSU clustering process, a day is divided to 96 time slots with each 15 minutes, so the clustering is based on the dataset of 96-dimensional vectors as we introduced in IV-A2. Then, the best K value is chosen by comparing varying SSE values, and in the following part, this value is used to deploy the simulation. Third, the trajectory dataset lasting for seven days is selected as the training dataset and the dataset for the other day is the testing set. Fourth, we choose different lengths of feature vectors to train the models in order to compare the accuracy of prediction.

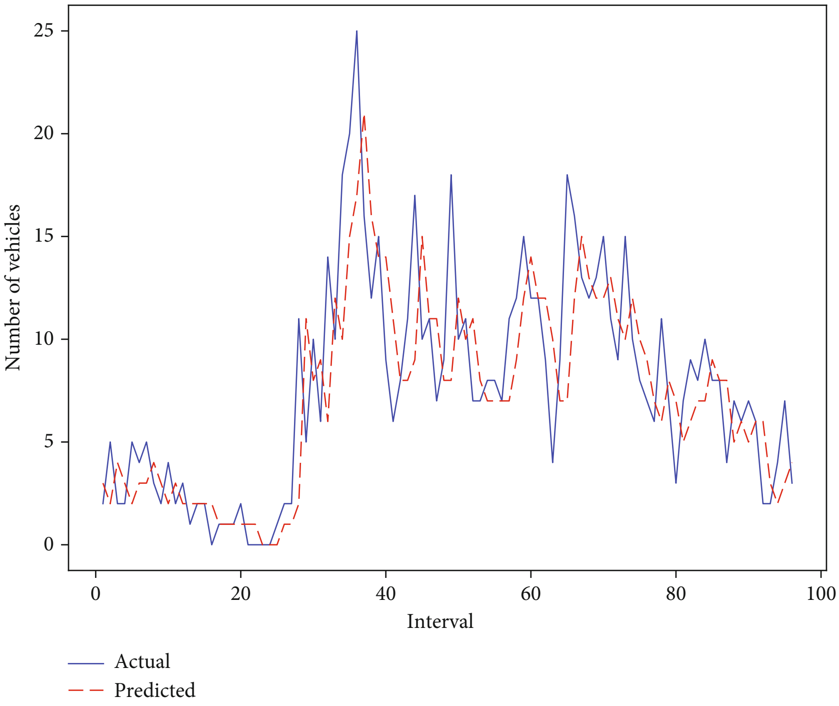
At last, in the RSU communication scheduling process, we compare the number of successfully accessed vehicles between our scheduling algorithm and other scheduling mechanisms when the energy is not sufficient for all the vehicles' communications.

5.2. Simulation Result

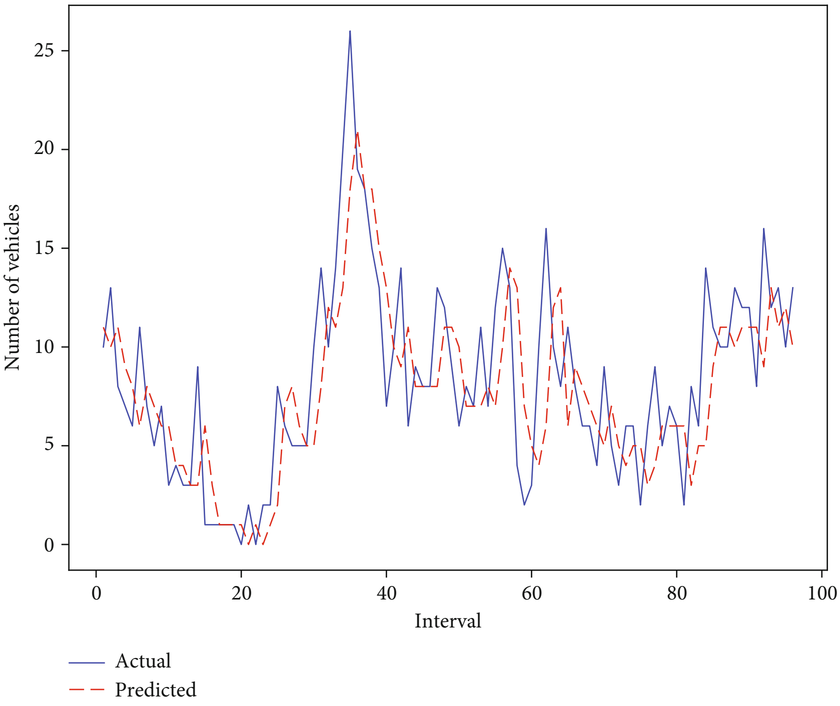
5.2.1. RSU Clustering. As described in IV-A1, the dimensionality of original traffic flow data needs to be reduced before RSUs can be clustered with t-SNE. For the convenience of data visualization, the 96D data is reduced to 2D in our simulation and the visualization of the original traffic flow is shown in Figure 6. Based on the dimensionality reduction result with t-SNE, the K-means algorithm can be performed. As shown in Figure 7(a), the SSE value decreases with K increasing. When K is 7, the SSE starts to descend slowly. Although the SSE value still falls down, the falling gradient becomes smaller. Therefore, it is believed that the best K value is 7. After the clustering, all the RSUs are divided to 7 clusters.

Then, in Figure 7(b), the clustering result is described with the average value of all the vectors in each cluster. It is found that 3 clusters are shown clearly and each of them has a special evolving rule. The other clusters are not very clear as most of elements in the vector are very small, even almost 0. In Figure 8, we take four clusters as examples, each of which contains several 96-dimensional vectors of RSUs. It is found that the vectors in a cluster show similar evolving trending, which can prove the correctness of our cluster mechanism.

5.2.2. Deep Learning-Based Traffic Prediction. In the deep learning-based prediction algorithm, the length of feature vectors is set to 5, 10, and 15. Table 2 shows the losses varying with various feature lengths for each cluster, in which the value is the Euclidean distance between the predicted vector and the original vector. When the length of feature is 5, the

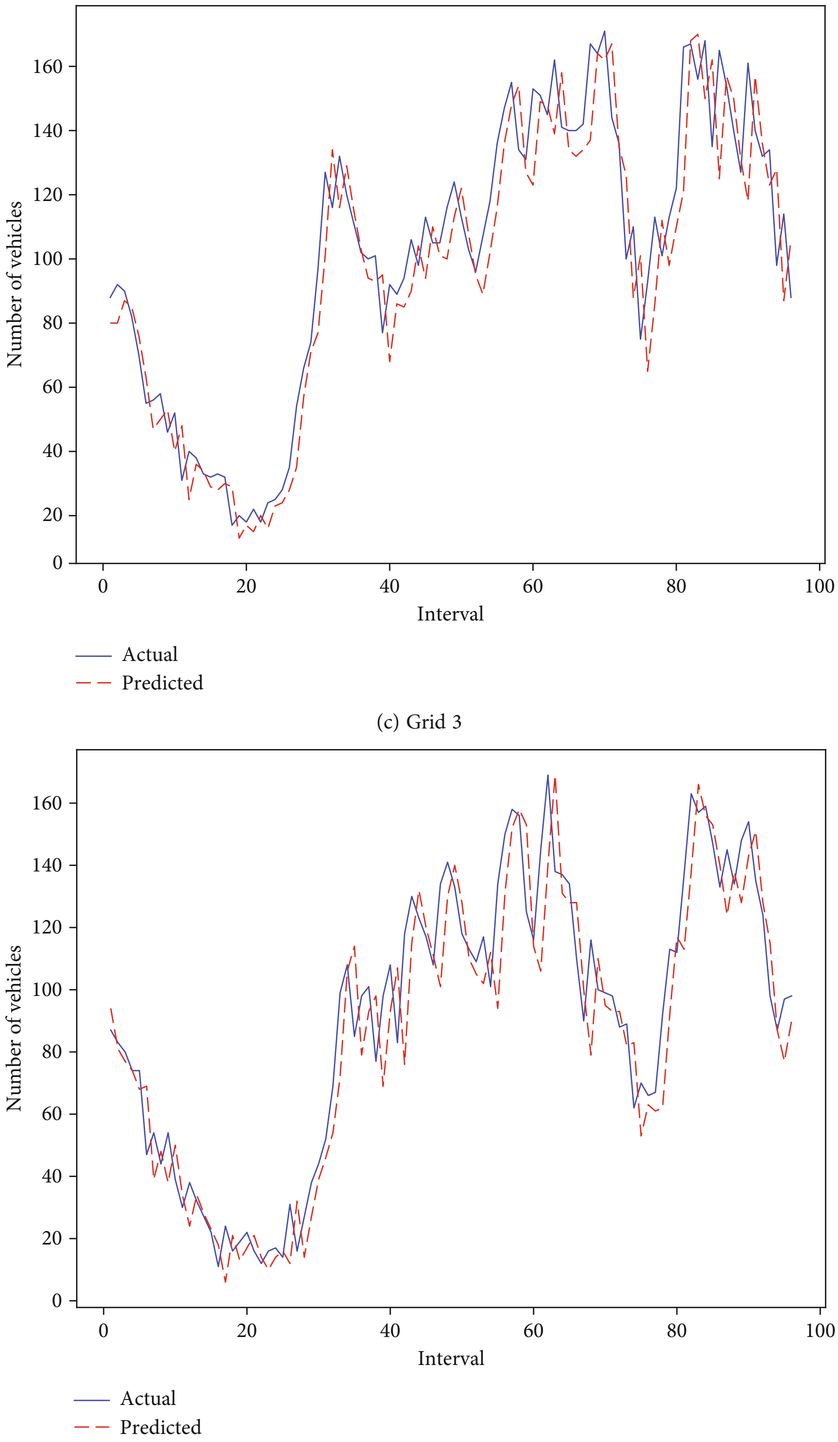


(a) Grid 1



(b) Grid 2

Figure 9: Continued.



(c) Grid 3

(d) Grid 4

Figure 9: Comparison between predicted vectors and original vectors.

loss is the highest compared to the others. For the length of the feature vector being 10 and 15, the losses for each cluster are almost the same. However, considering the computing resource exhausted, we choose the feature length as 10. In order to choose the best prediction model for RSUs in each class, the hyperparameters of the LSTM network are tuned by changing the number of layers and neurons of each layer at first. The activation function and loss function are set as "tanh" and "mse," respectively. Dropout and early stop are used to prevent overfitting, where the value of "dropout" is 0.2. The average MSEs of different models are shown in Table 3, in which "*N*" represents the number of neurons in each layer and "*L*" means the number of layers. As shown in Table 3, the model has less MSE loss when the layer is 1

Table 5: The energy consumption with different scheduling algorithms under two scenarios.

	FCFS	Random	Average	Priority
50%	429352	415540	415540	**414513**
75%	644000	622174	622174	**601213**

Table 6: The number of served vehicles with different scheduling algorithms under two scenarios.

	FCFS	Random	Average	Priority
50%	6355	6349	6193	**6396**
75%	6407	6371	6351	**6412**

and neuron is 256 in most cases. Therefore, the 1-layer LSTM model with 256 neurons in each layer is chosen to train the final prediction model in the flowing simulations. Moreover, the batch size also affects the prediction accuracy of training models. Thus, the models with different batch sizes are trained so that the best prediction model can be found in each class. The average prediction error with different batch sizes of each class is shown in Table 4, where "*B*" and "*C*" denote "batch size" and "class," respectively. The smallest error is shown in bold, and the corresponding model is selected as the final prediction model. Next, we show the comparison between the prediction vector and the original vector in four grids as examples. In Figure 9, four RSUs are randomly selected from four clusters, which reveals that each prediction model fits well for the corresponding cluster.

5.3. Scheduling Algorithm. In this part, we simulate the scenario that the RSUs do not have enough power to work through the entire period. Table 5 shows the energy consumption when the service ability is half and three quarters of requirement on different scheduling algorithms, in which "FCFS" denotes "first come first service," "random" means an RSU randomly chooses some vehicles to serve, "average" denotes the service ability of each interval is equal for each RSU, and "priority" represents that an RSU serves the vehicles according to the proposed scheduling algorithm in this paper. It can be seen that RSUs consume minimal energy in the two scenarios with the proposed scheduling algorithm.

Besides, the number of served vehicles under two scenarios are shown in Table 6. It is obvious that RSUs serve the maximized number of vehicles using minimized energy consumption with the proposed algorithm. Therefore, the performance of our proposed algorithm is proved.

6. Discussion and Conclusion

RSUs are playing a more and more important role in the IoVs. In terms of energy and deployment cost, renewable energy-powered RSUs will be used widely in the future. In this paper, we study the energy scheduling problem on renewable energy-powered RSUs. We propose a deep learning-based communication scheduling algorithm for RSUs in the IoVs, which is composed of RSU clustering, deep learning-based traffic prediction, service ability distribution algorithm, and priority based RSU accessing algorithm. At last, we conduct extensive simulation and the simulation results indicate that our algorithm can achieve a higher fairness rate while keeping a proper performance than the no scheduling algorithm.

Data Availability

The original data is the GPS data of taxis in Shenzhen, China, which were continuously sampled by the GPS devices during time period 2011/04/18–2011/04/26. Considering the large amount of original data, the processed data is submitted at the website https://http://github.com/pultoW/trajectory. If there is any question, you can contact the author via ljt_it@163.com.

Disclosure

A part of this manuscript has been related in the Conference "2020 92nd IEEE Vehicular Technology Conference, Oct. 4, 2020" with technique parts extended, and the simulation has been done with extended algorithms and mechanisms proposed in this paper [28].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part of the National Natural Science Foundation of China under Grant 61802004, the Scientific Research Project of Beijing Educational Committee under Grant KM202010009008, and Beijing Urban Governance Research Base in North China University of Technology.

References

[1] C. H. Ou, B. Y. Wu, and L. Cai, "GPS-free vehicular localization system using roadside units with directional antennas," *Journal of Communications and Networks*, vol. 21, no. 1, pp. 12–24, 2019.

[2] C. Wen and J. Zheng, "An RSU on/off scheduling mechanism for energy efficiency in sparse vehicular networks," in *2015 international conference on Wireless Communications & Signal Processing (WCSP).*, pp. 1–5, Nanjing, China, 2015.

[3] E. Benalia, S. Bitam, and A. Mellouk, "Data dissemination for Internet of vehicle based on 5G communications: a survey," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 5, article e3881, 2020.

[4] C. Wang, X. Cheng, J. Li, Y. He, and K. Xiao, "A survey: applications of blockchain in the Internet of vehicles," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, 16 pages, 2021.

[5] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 1–590, 2016.

[6] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[7] C. Wang, J. Li, Y. He, K. Xiao, and H. Zhang, "Destination prediction-based scheduling algorithms for message delivery in iovs," *IEEE Access*, vol. 8, pp. 14965–14976, 2020.

[8] M. M. Najm, M. Patra, and T. Venkatesh, "Cost-and-delay aware dynamic resource allocation in federated vehicular clouds," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6159–6171, 2021.

[9] W. S. Atoui, M. A. Salahuddin, W. Ajib, and M. Boukadoum, "Scheduling energy harvesting roadside units in vehicular ad hoc networks," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, Montreal, QC, Canada, 2016.

[10] Q. Ibrahim, "Design, implementation and optimisation of an energy harvesting system for vehicular ad hoc networks' road side units," *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 298–307, 2014.

[11] S. Mostofi, A. Hammad, T. D. Todd, and G. Karakostas, "On/off sleep scheduling in energy efficient vehicular roadside infrastructure," in *2013 IEEE International Conference on Communications (ICC)*, pp. 6266–6271, Budapest, Hungary, 2013.

[12] G. A. Audu, S. Bhattacharya, A. Muhtar, B. Qazi, and J. M. H. Elmirghani, "Reliability and quality of service of an off-grid wind powered roadside unit in a motorway vehicular environment," *Vehicular Communications*, vol. 9, pp. 176–187, 2017.

[13] Z. Ning, J. Huang, X. Wang, J. J. P. C. Rodrigues, and L. Guo, "Mobile edge computing-enabled Internet of vehicles: toward energy-efficient scheduling," *IEEE Network*, vol. 33, no. 5, pp. 198–205, 2019.

[14] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[15] A. A. Hammad, T. D. Todd, G. Karakostas, and D. Zhao, "Downlink traffic scheduling in green vehicular roadside infrastructure," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 3, pp. 1289–1302, 2013.

[16] B. L. Nguyen, D. T. Ngo, M. N. Dao, Q. T. Duong, and M. Okada, "A joint scheduling and power control scheme for hybrid I2V/V2V networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15668–15681, 2020.

[17] W. S. Atoui, W. Ajib, and M. Boukadoum, "Offline and online scheduling algorithms for energy harvesting RSUs in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6370–6382, 2018.

[18] A. Khezrian, T. D. Todd, G. Karakostas, and M. Azimifar, "Energy-efficient scheduling in green vehicular infrastructure with multiple roadside units," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 1942–1957, 2015.

[19] R. F. Atallah, C. M. Assi, and J. Y. Yu, "A reinforcement learning technique for optimizing downlink scheduling in an energy-limited vehicular network," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 4592–4601, 2017.

[20] R. Atallah, C. Assi, and M. Khabbaz, "Deep reinforcement learning-based scheduling for roadside communication networks," in *2017 15th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pp. 1–8, Paris, France, 2017.

[21] Y. J. Ku, P. H. Chiang, and S. Dey, "Real-time QoS optimization for vehicular edge computing with off-grid roadside units," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11975–11991, 2020.

[22] L. Dai, T. Chen, Y. Zhai, and G. Wang, "Energy-efficient distributed packet scheduling optimization strategy in cooperative vehicle infrastructure systems," *Wireless Communications and Mobile Computing*, vol. 2021, 11 pages, 2021.

[23] X. Wang, Z. Ning, X. Hu et al., "Future communications and energy management in the Internet of vehicles: toward intelligent energy-harvesting," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 87–93, 2019.

[24] Y. Chen, S. Tang, N. Bouguila, C. Wang, J. du, and H. L. Li, "A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data," *Pattern Recognition*, vol. 83, pp. 375–387, 2018.

[25] R. Elankavi, R. Kalaiprasath, and D. R. Udayakumar, "A fast clustering algorithm for high-dimensional data," *International Journal Of Civil Engineering And Technology (Ijciet)*, vol. 8, no. 5, pp. 1220–1227, 2017.

[26] J. Wang, N. Mao, X. Chen, J. Ni, C. Wang, and Y. Shi, "Multiple histograms based reversible data hiding by using FCM clustering," *Signal Processing*, vol. 159, pp. 193–203, 2019.

[27] W. Li, J. E. Cerise, Y. Yang, and H. Han, "Application of t-SNE to human genetic data," *Journal of Bioinformatics and Computational Biology*, vol. 15, no. 4, article 1750017, 2017.

[28] C. Wang, J. Li, X. Cheng, Y. He, L. Sun, and K. Xiao, "LSTM-based communication scheduling mechanism for energy harvesting RSUs in IoVs," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pp. 1–5, Victoria, BC, Canada, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5921181, 16 pages
https://doi.org/10.1155/2021/5921181

Research Article

Efficient Energy Utilization with Device Placement and Scheduling in the Internet of Things

Yanli Zhu,[1,2] Xiaoping Yang,[1] Yi Hong,[3,4] Youfang Leng,[1] and Chuanwen Luo[3,4]

[1] *School of Information, Renmin University of China, Beijing 100872, China*
[2] *School of Information and Engineering, Henan Institute of Science and Technology, Xinxiang 453003, China*
[3] *School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China*
[4] *Engineering Research Center for Forestry-Oriented Intelligent Information Processing of National Forestry and Grassland Administration, Beijing 100083, China*

Correspondence should be addressed to Chuanwen Luo; chuanwenluo@bjfu.edu.cn

Received 30 July 2021; Accepted 19 August 2021; Published 14 September 2021

Academic Editor: Yan Huang

The low-power wide-area network (LPWAN) technologies, such as LoRa, Sigfox, and NB-IoT, bring new renovation to the wireless communication between end devices in the Internet of things (IoT), which can provide larger coverage and support a large number of IoT devices to connect to the Internet with few gateways. Based on these technologies, we can directly deploy IoT devices on the candidate locations to cover targets or the detection area without considering multihop data transmission to the base station like the traditional wireless sensor networks. In this paper, we investigate the problems of the minimum energy consumption of IoT devices for target coverage through placement and scheduling (MTPS) and minimum energy consumption of IoT devices for area coverage through placement and scheduling (MAPS). In the problems, we consider both the placement and scheduling of IoT devices to monitor all targets (or the whole detection area) such that all targets (or the whole area) are (or is) continuously observed for a certain period of time. The objectives of the problems are to minimize the total energy consumption of the IoT devices. We first, respectively, propose the mathematical models for the MTPS and MAPS problems and prove that they are NP-hard. Then, we study two subproblems of the MTPS problem, minimum location coverage (MLC), and minimum energy consumption scheduling deployment (MESD) and propose an approximation algorithm for each of them. Based on these two subproblems, we propose an approximation algorithm for the MTPS problem. After that, we investigate the minimum location area coverage (MLAC) problem and propose an algorithm for it. Based on the MLAC and MESD problems, we propose an approximation algorithm to solve the MAPS problem. Finally, extensive simulation results are given to further verify the performance of the proposed algorithms.

1. Introduction

The Internet of things (IoT) is a flourishing paradigm in the scenario of modern wireless telecommunications, which has been provided a wide diversity applications for all walks of life in modern time, such as home automation, transportation, industry, agriculture, mobile device applications [1], and smart systems [2]. IoT applications are required a growing number of technologies to offer low-power operation and low-cost and low-complexity end devices that will be able to communicate wirelessly over long distances. With the development of the Low Power Wide Area Network (LPWAN) technologies, such as SigFox, NB-IoT and LoRa, the low power long-range wide-area communication has become a reality [3]. Since the long range communication of the LPWAN technologies is gradually used in the Internet of things, the IoT devices can only communicate with LPWAN gateways and not directly with each other. Taking the LoRa example, a single gateway can support as many as 10^5 IoT devices and three gateways are enough to cover all devices in the urban area within an approximate 15 km radius [4]. The architecture of the LPWAN-based Internet

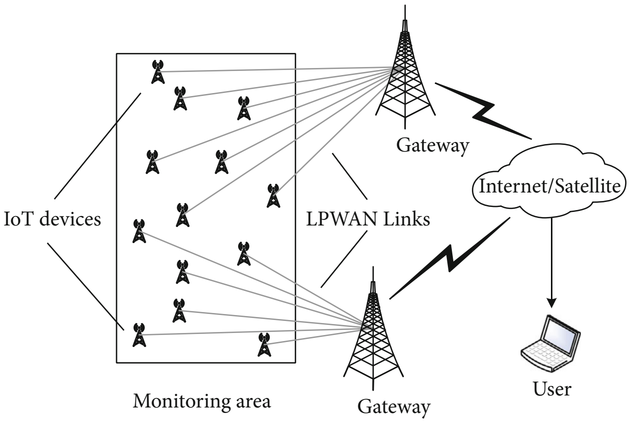


Figure 1: The architecture of the LPWAN-based Internet of things.

of things is shown in Figure 1 [5], in which IoT devices are deployed in the monitoring area to observe targets or the whole monitoring area; the installation of few gateways over the territory allows to gather data from IoT devices that are placed at different miles from the gateways. Then, the received data by the gateways are transmitted to the users through the Internet or satellite for further computational analysis to determine the appropriate response mechanism.

Therefore, we can directly deploy IoT devices to monitor all targets (or the whole monitoring area) in any region with the LPWAN-based network without considering other data transmission methods such as virtual backbone networks [6, 7] and mobile data collectors [8, 9]. Since many IoT devices are battery-powered sensors, for example, in wireless sensor networks (WSNs) and ad hoc networks (ANs), the power usage profile should be carefully designed in order to extend the battery lifetime. How to prolong the network lifetime is a classic problem in WSNs, which is called the coverage problem [10]. Given the m targets (or the entire monitoring area) and n IoT devices in the monitoring area, the coverage problem is to schedule the activity of the IoT devices such that all targets (or the whole monitoring area) are (or is) continuously observed and the network lifetime is maximized. Research on the coverage problem benefits a lot of applications, such as environment monitoring, battlefield surveillance, indoor guarding, smart space, industrial diagnostics, and military facility [11]. Recently, many researchers proposed various problems and corresponding algorithms for the coverage problem. In [12], Cardei et al. studied the target coverage problem with the objective of maximizing the network lifetime of a power constrained WSN deployed for detecting of a set of targets with known locations, in which they did not consider the placement of sensors. In [13], Akhlaghinia et al. studied the heterogeneous point coverage problem in sensor placement to cover a large number of target points with various coverage requirements using a minimum number of heterogeneous sensors. In the problem, they only investigated the placement of sensors without considering network lifetime. In [14], Mini et al. considered both the deployment locations and scheduling of the given IoT devices to maximize the network lifetime with the required coverage level. However, they deployed all available IoT devices to cover targets randomly without considering their candidate sites. In [15], Hanh et al. investigated the problem of maximizing the area coverage in heterogeneous WSNs. The goal of the problem is to find an optimal placement scheme for the given set of sensors so that the coverage area is maximized. In the problem, they only consider the placement of sensors without considering their candidate sites.

In the above literature, they only considered one of the deployment and scheduling of IoT devices or ignored the factor that candidate sites can be placed by IoT devices which has to be considered in some applications, such as a smart city. Actually, to minimize the total energy consumption of IoT devices, we not only need to consider the deployment of IoT devices but also their scheduling. Meanwhile, due to the emergence of wireless charging technologies and natural energy charging methods (e.g., solar charging) for the IoT devices, the current applications of IoTs have shifted from maximizing the network lifetime to working for a certain period of time. In this paper, we study the problems of the minimum energy consumption of IoT devices for target coverage through placement and scheduling (MTPS) and minimum energy consumption of IoT devices for area coverage through placement and scheduling (MAPS), where we consider both the placement and scheduling of IoT devices to monitor all targets or the entire monitoring area in a region such that all targets or the whole area are or is continuously observed for a certain period of time and the total energy consumption of all available IoT devices is minimized. The contributions of this paper are shown as follows:

(1) We propose two new practical models of minimizing the total energy consumption of all IoT devices by placing and scheduling them for continuously observing all targets or the entire detection area for a certain period of time. Then, we define the problems as the minimum energy consumption of IoT devices for target coverage through placement and scheduling (MTPS) and minimum energy consumption of IoT devices for area coverage through placement and scheduling (MAPS) and prove that they are NP-hard

(2) To solve the MTPS problem, we introduce two other problems, minimum location coverage (MLC) and minimum energy consumption scheduling deployment (MESD). Then, we propose an approximation algorithm for each of them. Afterwards, an approximation algorithm for the MTPS problem is proposed on the basis of the solutions for the MLC and MESD problems

(3) To solve MAPS problem, we introduce another problem, minimum location area coverage (MLAC). Then, we propose an approximation algorithm to solve the problem. Based on the problems MLAC and MESD, we propose an approximation algorithm to solve the MAPS problem

(4) We illustrate the effectiveness of the proposed algorithms by theoretical analysis and simulations

The remainder of this paper is organized as follows. We give the related works in Section 2. Section 3 introduces some models and definitions for the problems MTPS and MAPS. In Section 5, we propose an approximation algorithm to solve the MTPS problem. In Section 6, we propose an approximation algorithm to solve the MAPS problem. Simulations are shown in Section 7. Section 8 concludes this paper.

2. Related Works

In this section, we briefly review the major problems and methods related to the investigated problem in IoTs. As we all know, WSN is a special kind of IoTs. If there is no special explanation, the sensors in WSNs mentioned below represent IoT devices. According to the investigated problem, the related works can be categorized into three categories: IoT device placement problem, target coverage problem, and area coverage problem.

2.1. IoT Device Placement Problem. The IoT device placement problem aims at finding the least number of IoT devices and their locations within all known potential sensing locations for meeting requirements, such as [16–21].

In [16], Altinel et al. investigated the minimum cost point coverage problem with varying sensing quality and price and formulated a binary integer linear programming model for effective sensor placement on a grid-structured area. In [17], Wang introduced the sensor placement optimization problem, where the locations of targets to be covered are known and the candidate locations to place sensors are limited. The objective of the problem is to minimize the number of sensors to cover all targets, and the problem can be solved by the greedy algorithm for solving the set covering problem as shown in [22]. In [18], Gravalos et al. investigated the gateway placement problem for IoTs, which aims at finding the minimum number of gateways along with suitable IoT devices to optimize the overall installation cost without compromising the related QoS requirements. In [19], Jiang et al. proposed a group-greedy method to solve the sensor placement in linear inverse problems, which can find suboptimal solutions with near optimality guarantee using less computational cost compared with convex relaxation methods. In [20], Hasan and Al-Rizzo investigated the sensor deployment to improve the connectivity in IoT by presenting the bioinspired metaheuristics canonical particle multiswarm optimization algorithm. In [21], Jiang et al. studied the optimal sensor placement problem for an IoT-based power grid monitoring system. Then, they proposed a modified binary particle swarm optimization algorithm to determine the optimal number and location of sensors and estimate the ratio of conductor temperature alarms that can be covered by the proposed sensor placement.

2.2. Target Coverage Problem. In general, IoT devices are battery-powered sensors and there are often a lot of redundant sensors randomly placed in a region to cover a certain group of targets. How to schedule deployed sensors to maximize the network lifetime is an important problem in IoTs, which is called the maximum lifetime coverage problem (MLCP) and was proved NP-hard [12]. Currently, many researches devoted themselves to investigating the various problems of the MLCP problem, such as [23–26].

In [23], Berman et al. defined the MLCP problem as a sensor network life problem (SNLP) and proposed an approximation algorithm with a performance ratio of $1 + \ln n$ to solve the problem based on the minimum weigh sensor coverage problem (MWSCP) which aims at finding the minimum total weight of sensors to cover a certain set of targets, where n is the number of deployed sensors. In [24], Ding et al. first improved the algorithm for the MWSCP problem to $4 + \varepsilon$, where $\varepsilon > 0$. Then, they proposed an approximation algorithm with an approximate ratio of $4 + \zeta$ in the light of the MWSCP problem, where $\zeta > 0$. In [25], Lu et al. investigated the maximum lifetime coverage scheduling (MLCS) problem to address the scheduling problem for both target coverage and data collection in WSNs for maximizing the network lifetime. Then, they proposed an approximation algorithm with a constant factor for the problem. In [26], Shi et al. defined a new coverage problem in battery-free WSN, which is not only to maximize coverage quality but also to prolong network lifetime. Then, they proposed two centralized approximate algorithms and a distributed algorithm for solving the problem.

2.3. Area Coverage Problem. In [27], Xing et al. divided the detection area into grids and guaranteed the coverage of vertices of each grid to approximate the area coverage. However, the proposed approach as an approximation solution can not actually ensure the coverage of the whole area. In [28], Yu et al. studied the k-coverage problem, where a minimum subset of sensors among the deployed ones is selected such that each point in the detection area is covered by at least k sensors. In [29], Qin and Chen investigated the area coverage problem to maximize the coverage lifetime of wireless sensor networks for monitoring the area of interest. They proposed the area coverage algorithm based on differential evolution, which considered the balanced cost and minimal energy for sensors.

3. Model and Problem Definitions

In this sections, we introduce some parameters and problem definitions.

Let $\mathscr{A}$ be represented as a two-dimensional plane area; the whole of which can be observed by IoT devices that work together. There are m targets located on $\mathscr{A}$. Let $Q = \{r_1, r_2, \cdots, r_m\}$ represent the set of m targets located on $\mathscr{A}$. Due to the particularity of some detection areas, such as smart cities and farmlands, IoT devices can only be deployed on some fixed locations. We call these fixed locations as *candidate sites* of IoT devices. Let $L = \{L_1, L_2, \cdots, L_N\}$ denote the set of N candidate sites. Suppose that there exists n available IoT devices that can be deployed in the candidate sites to monitor targets in Q or the whole detection area $\mathscr{A}$. We use $S = \{s_1, s_2, \cdots, s_n\}$ to represent the set of n IoT devices, in which each IoT device s_i can be active continuously at most E_i time slots. In this paper, we assume that each IoT device $s_i \in S$ continuously works E_i time slots once it starts working and $E_i \leq T$, where T is the minimum time that the targets (or the detection area) are (or is) continuously observed. For any pair of $s_i \in S$ and $r_j \in Q$ (or $L_k \in L$ and $r_j \in Q$), let d_{s_i, r_j} (or d_{L_k, r_j}) denote the Euclidean distance between s_i and r_j (or L_k and r_j).

In this paper, we aim to find a subset $C \subseteq S$ of IoT devices which are deployed on some candidate sites in L to observe all targets in Q or the whole detection area and to minimize the total energy consumption of all IoT devices by scheduling IoT devices in C such that all targets or the whole detection area can be continuously observed by IoT devices at least T time. More formally, for two different kinds of problems, target coverage and area coverage, we call the research problems as the minimum energy consumption of IoT devices for target coverage through placement and scheduling (MTPS) and the minimum energy consumption of IoT devices for area coverage through placement and scheduling (MAPS), respectively, whose detailed definitions are shown in Definitions 1 and 2.

Definition 1 (MTPS). Given a set $Q = \{r_1, r_2, \cdots, r_m\}$ of m targets located on detection area $\mathscr{A}$, a set $S = \{s_1, s_2, \cdots, s_n\}$ of n IoT devices in which all IoT devices have the same coverage range R and each IoT device $s_i \in S$ can work E_i time slots, a set $L = \{L_1, L_2, \cdots, L_N\}$ of N candidate sites to be put on IoT devices, a positive time T, the minimum energy consumption of IoT devices for target coverage through placement and scheduling (MTPS) problem aims at finding a subset $C \subseteq S$ of IoT devices placed on the candidate sites in L and scheduling the IoT devices in C such that

(1) For any candidate site $L_k \in L$, it can be placed in more than one IoT device from C

(2) For arbitrary target $r_j \in Q$, it is continuously observed by IoT devices in C at least T time

(3) For each IoT device $s_i \in C$, it continuously works in E_i time slots once it starts working

(4) The total energy consumption of IoT devices in C, $M = \sum_{s_i \in C} E_i$, is minimized

Definition 2 (MAPS). Given a detection area $\mathscr{A}$, a set $S = \{s_1, s_2, \cdots, s_n\}$ of n IoT devices in which all IoT devices have the same coverage range R, and each IoT device $s_i \in S$ can work E_i time slots, a set $L = \{L_1, L_2, \cdots, L_N\}$ of N candidate sites to be put in IoT devices, a positive time T, the minimum energy consumption of IoT devices for area coverage through placement and scheduling (MAPS) problem aims at finding a subset $C \subseteq S$ of IoT devices placed on the candidate sites in L and scheduling the IoT devices in C such that

(1) For any candidate site $L_k \in L$, it can be placed in more than one IoT devices from C

(2) For arbitrary point $p \in \mathscr{A}$, it is continuously observed by IoT devices in C at least T time

(3) For each IoT device $s_i \in C$, it continuously works in E_i time slots once it starts working

(4) The total energy consumption of IoT devices in C, $M = \sum_{s_i \in C} E_i$, is minimized

4. Mathematical Formulation for the Problems

In this section, we will introduce the mathematical formulations for the problems MTPS and MAPS.

We first introduce some notations as follows:

i is the index of IoT devices, where $1 \leq i \leq n$. j is the index of targets, where $1 \leq j \leq m$. k is the index of candidate sites, where $1 \leq k \leq N$. t is the index of active time slots, where T time is divided into T time slots. We define the binary variables x_{ik}, a_{jk}, and y_{it} as follows:

$$x_{ik} = \begin{cases} 1, & \text{if } s_i \text{ is placed at } L_k, \\ 0, & \text{otherwise}, \end{cases}$$
$$a_{jk} = \begin{cases} 1, & \text{if } d_{L_k, r_j} \leq R, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$
$$y_{it} = \begin{cases} 1, & \text{if } s_i \text{ is active at } T \text{ time slot}, \\ 0, & \text{otherwise}. \end{cases}$$

4.1. Mathematical Formulation for the MTPS Problem. In this subsection, we will introduce the mathematical formulation for the MTPS problem. The problem can be formulated into an integer programming (IP) problem as follows:

$$\min \sum_{k=1}^{N} \sum_{i=1}^{n} x_{ik} \cdot E_i, \tag{2}$$

s.t.

$$\sum_{k=1}^{N} \sum_{i=1}^{n} a_{jk} \cdot x_{ik} \cdot y_{it} \geq 1, \quad j = 1, 2, \cdots, m, t = 1, 2, \cdots, T, \tag{3}$$

$$\sum_{k=1}^{N} \sum_{t=b_i}^{b_i+E_i-1} x_{ik} \cdot y_{it} = E_i \, \exists b_i \in \{1, 2, \cdots, T\}, \quad i = 1, 2, \cdots, n, \tag{4}$$

$$x_{ik} \in \{0, 1\}, \quad i = 1, 2, \cdots, n, k = 1, 2, \cdots, N, \tag{5}$$

$$a_{jk} \in \{0, 1\}, \quad j = 1, 2, \cdots, m, k = 1, 2, \cdots, N, \tag{6}$$

$$y_{it} \in \{0, 1\}, \quad i = 1, 2, \cdots, n, t = 1, 2, \cdots, T. \tag{7}$$

The function of equation (2) is to minimize the total energy consumption of IoT devices for continuous observing of all targets at least T time. Constraint (3) ensures that for each target $r_j \in Q$, there at least exists an IoT device $s_i \in S$

located at some candidate site $L_k \in L$ to cover r_j at any tth time slot. Constraint (4) guarantees that the IoT device $s_i \in S$ located on $L_k \in L$ will run out of energy as soon as it starts working. Constraints (5)–(7) define the domains of the variables.

4.2. Mathematical Formulation for the MAPS Problem. In this subsection, we will introduce the mathematical formulation for the MAPS problem. Let S_i be the sensing region of sensor s_i. We use $||A||$ to denote the area of $\mathscr{A}$. The problem can be formulated as follows:

$$\min \sum_{k=1}^{N}\sum_{i=1}^{n} x_{ik} \cdot E_i, \tag{8}$$

$$\text{s.t.} \left\| \left(\bigcup_{s_i \in C} S_i \cdot \sum_{k=1}^{N} x_{ik} \cdot y_{it} \right) \bigcap \mathscr{A} \right\| = ||A||, \quad t = 1, 2, \cdots, T, \tag{9}$$

$$\sum_{k=1}^{N} \sum_{t=b_i}^{b_i+E_i-1} x_{ik} \cdot y_{it} = E_i \, \exists b_i \in \{1, 2, \cdots, T\}, \quad i = 1, 2, \cdots, n, \tag{10}$$

$$x_{ik} \in \{0, 1\}, \quad i = 1, 2, \cdots, n, k = 1, 2, \cdots, N, \tag{11}$$

$$y_{it} \in \{0, 1\}, \quad i = 1, 2, \cdots, n, t = 1, 2, \cdots, T, \tag{12}$$

The function of equation (8) is to minimize the total energy consumption of IoT devices for continuous observing of the whole detection area $\mathscr{A}$ at least T time. Constraint (9) ensures that for any point $p \in \mathscr{A}$, there at least exists an IoT device $s_i \in S$ located at some candidate site $L_k \in L$ to cover p at any tth time slot. Constraint (10) guarantees that the IoT device $s_i \in S$ located on $L_k \in L$ will run out of energy as soon as it starts working. Constraints (11)–(12) define the domains of the variables.

4.3. NP-Hard Proofness. In the following, we will prove that the problems MTPS and MAPS are NP-hard. We first prove that the MTPS problem is NP-hard. Then, based on the MTPS problem, we prove that the MAPS problem is also NP-hard.

We consider a special case of the MTPS problem where we set $T = 1$, $N = n$, and $E_i = 1$ for any IoT device $s_i \in S$. At this point, the objective of the MTPS problem can be transformed to find a subset $C \subseteq S$ of IoT devices with minimum cardinality such that all targets are covered. Since for any IoT device $s_i \in C$, it needs to be placed at the corresponding candidate site to cover targets, the objective of the MTPS problem changes from finding the minimum subset C to looking for the minimum subset $L' \subseteq L$ of candidate sites, where $|C| = |L'|$. Therefore, the special case of the MTPS problem can be equivalently transformed into the minimum point cover (MPC) problem as shown in Definition 3.

Definition 3 (MPC). Given a set $Q = \{r_1, r_2, \cdots, r_m\}$ of m targets, a set $L = \{L_1, L_2, \cdots, L_N\}$ of N candidate sites to be put in IoT devices, and all IoT devices have the same coverage range R, the minimum point cover (MPC) problem is to find a subset $L' \subseteq L$ of candidate sites such that all targets are covered by IoT devices located on the candidate sites in L' and the number of candidate sites $|L'|$ is minimized.

In the MPC problem, for each candidate site $L_k \in L$, we use U_k to denote the set of targets covered by L_k where for each target r_j, $r_j \in U_k$ if and only if $d_{L_k, r_j} \leq R$. Let $F = \{U_1, U_2, \cdots, U_N\}$. Then, the MPC problem can be equivalently transformed into the set cover (SC) problem, as shown in Definition 4.

Definition 4 (SC). Given a set $Q = \{r_1, r_2, \cdots, r_m\}$ of m targets and a collection $F = \{U_1, U_2, \cdots, U_N\}$ of N sets, where each $U_k \in F$ is a subset of Q, the set cover (SC) problem is to find a subset $F' \subset F$ such that $\bigcup_{U_k \in F}{}' U_k = Q$ and $|F'|$ is minimum.

Theorem 5. *The MPC problem is NP-hard.*

Proof. According to the SC and MPC problems, we can obtain that the decision version of the MPC problem has a YES answer if and only if the decision version of the SC problem has a YES answer and $|F'| = |L'|$. Since the SC problem was proved NP-hard [30], the MPC problem is NP-hard. □

Theorem 6. *The MTPS problem is NP-hard.*

Proof. According to Theorem 5, we can verify that the MPC is NP-hard. Since the MPC problem is a special case of the MTPS problem, the MTPS problem is also NP-hard. □

Theorem 7. *The MAPS problem is NP-hard.*

Proof. Since the continuous region $\mathscr{A}$ is made up of an infinite set of points, we can take any set of discrete points in $\mathscr{A}$ as a special case of the MAPS where all the discrete points can be seen as targets for being observed by IoT devices. In such a case, the MAPS problem can be transformed into the MTPS problem. According to Theorem 6, we can verify that the MTPS is NP-hard. Therefore, the MAPS problem is also NP-hard. □

5. Algorithm for the MTPS Problem

According to the definition of the MTPS problem, we can obtain that the total energy consumption of the IoT devices depends on the number of IoT devices with their corresponding initial energy. Therefore, we need to deploy IoT devices as few as possible to cover targets and schedule the placement of IoT devices to minimize the total energy consumption such that every target in Q is continuously observed at T time slots by IoT devices. Based on these considerations, we can find that the MTPS problem consists of two subproblems, minimum location coverage (MLC) and minimum energy consumption scheduling deployment

(MESD), as shown in Definitions 8 and 9. In this section, we first propose an approximation algorithm for each of the problems MLC and MESD. Then, based on the problems MLC and MESD, we propose an approximation algorithm to solve the MTPS problem.

Definition 8 (MLC). Given a set $Q=\{r_1, r_2,\cdots,r_m\}$ of m targets, a set $L=\{L_1, L_2,\cdots,L_N\}$ of N candidate sites to be put in IoT devices with coverage range R, the minimum location coverage (MLC) problem is to find minimum subset $L' \subseteq L$ of candidate sites such that all targets are within the coverage range of candidate sites in L'.

Definition 9 (MESD). Given a set $S=\{s_1, s_2,\cdots,s_n\}$ of n IoT devices in which each IoT device $s_i \in S$ has active time E_i, a set $P=\{L_1, L_2,\cdots,L_K\}$ of K sites to be put IoT devices, a positive time T, the Minimum Energy consumption Scheduling Deployment (MESD) problem is to find a subset $C \subseteq S$ of IoT devices placed at the sites in P and to schedule the IoT devices in C such that

(1) for any site $L_k \in P$, it can be placed more than one IoT device from C

(2) for arbitrary site $L_k \in P$, the IoT devices located at L_k can cumulatively work at least T time

(3) for each IoT device $s_i \in C$, it continuous works E_i time once it starts working, and

(4) the total energy consumption of IoT devices in C, $M=\sum_{s_i\in C} E_i$, is minimized

5.1. Algorithm for the MLC Problem. In this subsection, we propose a greedy algorithm, called MLCA, to solve the MLC problem. Let U_k denote the set of targets within the coverage range of L_k. The MLCA algorithm consists of two steps. Firstly, for arbitrary $L_k \in L$, we compute its coverage set U_k. For any $r_j \in Q$, if $d_{L_k,r_j} \le R$, then, $U_k = U_k \cup \{r_j\}$. Secondly, we repeat the following steps until one of the conditions $L=\emptyset$ and $Q=\emptyset$ is satisfied.

(i) Select L_k with the maximum U_k from L

(ii) Execute the operations $L' = L' \cup \{L_k\}$, $L=L/\{L_k\}$ and $Q=Q/U_k$

(iii) For arbitrary $L_i \in L$, update its coverage set by deleting targets in $U_k \cap U_i$ from U_i

After executing the above algorithm, we can obtain a set $L' \subseteq L$ of candidate sites, which can cover all targets in Q. The pseudocode of the algorithm is shown in Algorithm 1. Then, we will analyze the performance of the MLCA algorithm.

Theorem 10. *Suppose that L^* is an optimal solution for the MLC problem. If there exists a solution for the MLC problem, then, we can verify that the approximation ratio of the MLCA algorithm is* $\ln m + 1$*, where m is the number of targets.*

```
Input: $Q=\{r_1, r_2,\cdots,r_m\}$, $L=\{L_1, L_2,\cdots,L_N\}$, $R$;
Output: $L'$;
Sets of $L'=\emptyset$, $U_k=\emptyset$ for any $L_k \in L$;
for arbitrary $L_k \in L$do
    for any $r_j \in Q$do
        if $d_{L_k,r_j} \le R$ then
            $U_k = U_k \cup \{r_j\}$;
        end
    end
end
while $L \neq \emptyset \&\& Q \neq \emptyset$ do
    Pick $L_k = \arg\ \max_{L_k \in L} U_k$;
    $L' = L' \cup \{L_k\}$, $L = L \setminus \{L_k\}$, $Q = Q \setminus U_k$;
    for any $L_i \in L$do
        $U_i = U_i \setminus U_k$;
    end
end
if $Q=\emptyset$ then
There is no solution for the MLC problem;
end
```

Algorithm 1: MLCA.

Proof. According to the MLCA algorithm, we can observe that the while loop terminates after at most m steps, since in each iteration of the while loop, there is at least one target that is covered by the candidate site L_k. Let Q_k denote the number of targets that are still not covered at iteration k of the while loop. In each iteration k, we can use all $|L^*|$ candidate sites in the optimal solution to cover all targets in Q. Therefore, there must exist a candidate site in L^* that covers at least $Q_k/|L^*|$ targets, which means at least $Q_k/|L^*|$ targets are covered in every iteration. In other words, we can obtain after iteration k; there are left at most $Q_k - Q_k/|L^*|$ targets that have not been covered by candidate sites, that is,

$$\begin{aligned} Q_{k+1} &\le \left(1-\frac{1}{|L^*|}\right)\cdot Q_k \le \left(1-\frac{1}{|L^*|}\right)^2 \cdot Q_{k-1} \le, \cdots, \\ &\le \left(1-\frac{1}{|L^*|}\right)^{k+1} \cdot Q_0 = \left(1-\frac{1}{|L^*|}\right)^{k+1} \cdot m, \end{aligned} \tag{13}$$

where the last equality depends on the fact that $Q_0 = m$, since all m tragets are not covered by candidate sites before the first iteration of the while loop. Notice that there exists $1 \le i \le m$ such that after executing i iterations of the while loop, $Q_i \le 1$. Based on the fact that $1 + x \le e^x$ for any $x \in (-\infty, +\infty)$, we have

$$\left(1-\frac{1}{|L^*|}\right)^i = \left(\left(1-\frac{1}{|L^*|}\right)^{|L^*|}\right)^{i/|L^*|} \le e^{-i/|L^*|}. \tag{14}$$

Based on inequations (13) and (14), we can obtain

$$m \cdot e^{-i/|L^*|} \le 1 \Leftrightarrow i \ge |L^*| \cdot \ln m. \tag{15}$$

Therefore, we can obtain that after $i = |L^*| \cdot \ln m$ iterations, the remaining number of targets in Q_i is smaller or equal to 1. Thus, the algorithm will terminate after at most $|L^*| \cdot \ln m + 1$ iterations, which can obtain $|L'| \leq |L^*| \cdot \ln m + 1 \leq (\ln m + 1) \cdot |L^*|$, since $|L^*| \geq 1$ and only one candidate site is added into L' in each iteration based on the algorithm MLCA. □

Theorem 11. *The time complexity of the MLCA algorithm is $O(mN)$, where m and N are the number of targets and the number of candidate sites, respectively.*

Proof. The MLCA algorithm consists of two phases. Firstly, the algorithm needs N iterations to compute the corresponding coverage sets for all candidate sites in L, as shown in the first while loop. In each iteration, since the number of targets is less than or equal to m, at most m steps are needed to determine which coverage set targets belong to. Therefore, we need at most mN steps to compute the coverage sets for all candidate sites in L. Secondly, at most $\min\{N, m\}$ iterations are needed in the while loop. In each iteration, we need to pick the candidate site L_k with the maximum U_k for executing N steps since $1 \leq k \leq N$. Then, we update all coverage sets of candidate sites with N steps. Therefore, we need at most $N \cdot \min\{N, m\}$ steps in the second while loop.

Consequently, the time complexity of the MLCA algorithm is $O(mN) = O(mN + N \cdot \min\{N, m\})$. □

5.2. Algorithm for the MESD Problem. In this subsection, we propose an approximation algorithm to solve the MESD problem, called MESDA. Before describing the algorithm, we introduce some notations. For any $1 \leq k \leq K$, we use C_k to denote the set of IoT devices placed at location L_k and let $\Phi = \{C_1, C_2, \cdots, C_K\}$. Let M_k represent the total energy consumption of the IoT devices in C_k.

The MESDA consists of two phases. The first phase is to find a subset $C_k \subseteq S$ of IoT devices from S for any $1 \leq k \leq K$ such that $\sum_{s_i \in C_k} E_i \geq T$. The second phase is to optimize M_k by replacing the high-energy-consuming IoT devices in C_k with low-energy-consuming ones from the remaining IoT devices in S for any $1 \leq k \leq K$. Afterwards, we compute $\Phi = \{C_1, C_2, \cdots, C_K\}$, $C = \bigcup_{1 \leq k \leq K} C_k$, and $M = \sum_{C_k \in \Phi} M_k$. The detailed description of the algorithm is shown as follows.

Initially, we set $\Phi = \emptyset$, $C = \emptyset$, $M = 0$, $C_k = \emptyset$, and $M_k = 0$ for each $1 \leq k \leq K$. The first phase of the MESD algorithm repeats the following four steps until the conditions $P = \emptyset$ and $S = \emptyset$ are satisfied.

(i) Select s_i with the maximum E_i from S, where if there exists two IoT devices $s_i, s_j \in S$ such that $E_i = E_j$, then, their maximum ID is selected

(ii) Pick L_k with the minimum M_k for any $1 \leq k \leq K$, where if there exist M_k and M_l such that $M_k = M_l$, then, their minimum ID is selected

(iii) Add s_i into C_k, and $M_k = M_k + E_i$. Then, $S = S/\{s_i\}$

(iv) Compare M_k with T. If $M_k \geq T$, then, $P = P/\{L_k\}$

After executing the first phase of the algorithm, we can obtain a set C_k of IoT devices for any $L_k \in P$, where the total working time M_k of IoT devices is greater than or equal to T. In the following, for any $1 \leq k \leq K$, we optimize M_k by replacing the high-energy-consuming IoT devices in C_k with low-energy-consuming ones in S.

The second phase of the algorithm repeats the following steps until $S = \emptyset$.

(i) Select s_i with the maximum E_i from S

(ii) Compute a tuple $< s_j, k \geq \arg \min_{s_j \in C_k, 1 \leq k \leq K} (M_k - E_j + E_i)$ such that $M_k - E_j + E_i \geq T$

(iii) Compare E_i with E_j. If $E_i \geq E_j$, then, s_i is deleted from S, otherwise, $C_k = C_k \cup \{s_j\}/\{s_i\}$, $M_k = M_k - E_i + E_j$, and s_i is removed from S

After executing the second phase of the algorithm, we can obtain a set C_k of IoT devices located on each $L_k \in P$ and the total energy consumption $M_k = \sum_{s_i \in C_k} E_i$ of IoT devices in C_k such that all targets covered by site L_k are continuously observed at least T time. Finally, we can obtain $\Phi = \{C_1, C_2, \cdots, C_K\}$, $M = \sum_{C_k \in \Phi} M_k$, and $C = \bigcup_{1 \leq k \leq K} C_k$. The pseudocode of the algorithm is shown in Algorithm 2.

We use C^* to represent the optimal set of IoT devices placed at sites in P for the MESD problem. Let M^* denote the total energy consumption of IoT devices in C^*. Without loss of generality, we use C_k^* to be the optimal set of IoT devices placed at $L_k \in P$ when C^* has been confirmed. Let M_k^* represent the total energy consumption of IoT devices in C_k^*.

Theorem 12. *If the MESDA algorithm has the feasible solution, then, we can verify that the approximation ratio of the algorithm is 2.*

Proof. According to the definition of the MESD problem, we have $M_k^* \geq T$ for any $1 \leq k \leq K$ and $M^* = \sum_{1 \leq k \leq K} M_k^* \geq K \cdot T$.

For any $1 \leq k \leq K$, we let $<s_i, E_i \geq \arg \max_{s_j \in C_k} E_j$, where C_k is obtained by the MESDA algorithm. We analyze the performance of the algorithm in the light of the following two cases.

(1) $0 < E_i < (T/2)$. Then, we can obtain that for any $s_j \in C_k$, $E_j < (T/2)$. Based on the algorithm, the last IoT device added into C_k makes M_k be greater than or equal to T, which means $M_k \leq T + E_j < (3T/2) \leq 1.5M_k^*$

(2) $(T/2) \leq E_i < T$. If there exists $E_j \in C_k$ such that $(T/2) < E_j \leq E_i$, then based on the algorithm, we can obtain $C_k = \{s_i, s_j\}$ and $M_k = E_i + E_j < 2T < 2 M_k^*$. Otherwise, we can derive $M_k \leq T + E_j \leq 1.5 M_k^*$

```
Input: $S = \{s_1, s_2, \cdots, s_n\}$, $E_i$ for each $s_i \in S$, $P = \{L_1, L_2, \cdots, L_K\}$, $T$;
Output: $C$, $M$, $\Phi$;
Sets of $C = \varnothing$, $M = 0$, $C_k = \varnothing$ and $M_k = 0$ for each $1 \le k \le K$;
while $P \neq \varnothing \&\& S \neq \varnothing$ do
    Pick $s_i = \arg\ \max_{s_i \in S} E_i$;
    Pick $L_k = \arg\ \min_{L_k \in P} M_k$;
    $C_k = C_k \cup \{s_i\}$, $M_k = M_k + E_i$, $S = S \setminus \{s_i\}$;
    if $M_k \ge T$ then
        $P = P \setminus \{L_k\}$;
    end
end
while $S \neq \varnothing$ do
    Pick $s_i = \arg\ \max_{s_i \in S} E_i$;
    Select $<s_j, k> = \arg \min_{s_j \in C_k, 1 \le k \le K} (M_k - E_j + E_i)$ such that $M_k - E_j + E_i \ge T$;
    if $E_i \ge E_j$ then
        $S = S \setminus \{s_i\}$;
    else
        $C_k = C_k \cup \{s_j\} \setminus \{s_i\}$, $M_k = M_k - E_j + E_i$, $S = S \setminus \{s_i\}$;
    end
end
$\Phi = \bigcup_{1 \le k \le K} \{C_k\}$, $M = \sum_{C_k \in \Phi} M_k$, $C = \bigcup_{C_k \in \Phi} C_k$;
```

Algorithm 2: MESDA.

From what have been discussed, we can obtain $M_k < 2 M_k^*$. Therefore, we have $M = \sum_{1 \le k \le K} M_k \le \sum_{1 \le k \le K} 2 M_k^* \le 2 M^*$, which means that the approximation ratio of the MESDA algorithm is 2. □

Theorem 13. *The time complexity of the MESDA algorithm is $O(n^3 + nK)$, where n and K are the number of IoT devices and the number of sites, respectively.*

Proof. According to the MESDA algorithm, we can verify that the algorithm contains two while loops running and the other operations with constant running time. The first while loop consists of at most n iterations, since $|S| \le n$. In each iteration, at most n steps are needed to select the IoT device s_i with the maximum energy and K steps are required to compute the site L_k with the minimum total energy consumption of IoT devices located on L_k. The other operations in the iteration can be executed in constant time. Therefore, the first while loop runs at most $O(n^2 + nK)$ time. The second while loop runs $|S|$ iterations. In each iteration, the algorithm runs at most $|S|$ steps to select the IoT device s_i with the maximum energy. Then, it calculates the total energy consumption with at most $O(n \cdot |S|)$ running time by replacing each IoT device that has been deployed on site in P with s_i and selects s_j with the minimum $M_k - E_j + E_i$ among all IoT devices except IoT devices in S. The other operations in the iteration can be executed in constant time. Therefore, the total running time of the second while loop is at most $O(n^3) = O(n \cdot |S|^2 + |S|^2)$.

Consequently, we can obtain that the time complexity of the MESDA algorithm is $O(n^3 + nK) = O(n^2 + nK) + O(n^3)$. □

5.3. Algorithm for the MTPS Problem. In this subsection, we propose an approximation algorithm, called MTPSA, to solve the MTPS problem based on the MLC and MESD problems. The algorithm consists of two steps corresponding to Algorithms 1 and 2. The detailed illustration of the algorithm is shown in Algorithm 3.

Suppose that C^*_{opt} is an optimal subset of S for the MTPS problem. Let M^*_{opt} be the total energy consumption of IoT devices in C^*_{opt}. Since each $r_j \in Q$ needs to be continuously observed at least T time, we divide T into equal T time slots. We use c_t^* to represent the minimum energy consumption of all active IoT devices for the MTPS problem such that all targets are covered at the tth time slot, and let L_t^* denote the set of candidate sites placed in the active IoT devices in the tth time slot.

Lemma 14. *For any $1 \le t \le T$, we have $c_t^* \ge |L^*|$, where L^* is the minimum set of candidate sites for the MLC problem.*

Proof. In the MTPS problem, all targets need to be covered by IoT devices placed at candidate sites in L for any $1 \le t \le T$, which means that there exists a subset $L_t \subseteq L$ that can cover all targets for the arbitrary tth time slot. Thus, based on the definitions of MTPS problem and MLC problem, we can obtain that for any such subset L_t, it is a feasible solution for the MLC problem. Therefore, we have $|L_t^*| \ge |L^*|$,

```
**Input**: $Q = \{r_1, r_2, \cdots, r_m\}S = \{s_1, s_2, \cdots, s_n\}$, $E_i$ for each $s_i \in S$, $L = \{L_1, L_2, \cdots, L_N\}$, $R$, $T$;
**Output**: $C$, $M$, $\Phi$;
**Step 1:** Compute the set $L'$ of candidate sites to cover all targets in $Q$ by executing Algorithm 1;
**Step 2:** Compute $\Phi$, $C$, $M$ based on $L'$ by executing Algorithm 2,
where $P = L'$;
```

Algorithm 3: MTPSA.

since L_t^* is a feasible solution for the MLC problem. Therefore, we can derive $c_t^* \geq |L^*|$ since $c_t^* = |L_t^*|$. □

Lemma 15. *We can obtain $M_{opt}^* \geq |L^*| \cdot T$.*

Proof. According to Lemma 14, we can obtain $c_t^* \geq |L^*|$ for arbitrary $1 \leq t \leq T$. Based on the definition of the MTPS problem, we have $M_{\text{opt}}^* \geq \sum_{1 \leq t \leq T} c_t^* \geq |L^*| \cdot T$. □

Theorem 16. *The performance ratio of the MTPSA algorithm is 2 $\ln m + 2$, where m is the number of targets.*

Proof. According to the MTPSA algorithm, we have $M = \sum_{L_k \in L}{}' M_k$, where M is the total energy consumption of IoT devices obtained by the MTPSA algorithm and M_k is the energy consumption of IoT devices located on L_k. On the basis of Theorems 10 and 12, we can derive $|L'| \leq \ln m \cdot |L^*| + 1$ and $M_k < 2T$ for any $1 \leq k \leq |L'|$. Based on Lemma 15, we can obtain

$$\begin{aligned} M &= \sum_{L_k \in L'} M_k \leq \left|L'\right| \cdot \max \left\{ M_k \mid L_k \in L' \right\} \\ &\leq (\ln m \cdot |L^*| + 1) \cdot 2T \leq 2(\ln m + 1) \cdot M_{\text{opt}}^*. \end{aligned} \tag{16}$$

The theorem has been proved. □

Theorem 17. *The time complexity of the MTPSA algorithm is $O(n^3 + mN + nN)$, where n is the number of available IoT devices in S, m denotes the number of targets in Q, and N represents the number of candidate sites in L.*

Proof. Based on Theorems 11 and 13, we can obtain that the time complexity of the MTPSA algorithm is $O(n^3 + mN + nN) = O(mN + n^3 + nK)$, since $K = |L'| \leq N$. □

6. Algorithm for the MAPS Problem

In this section, we propose an algorithm to solve the MAPS problem.

Similar to the definition of the MTPS problem, the total energy consumption of the IoT devices depends on the number of IoT devices with their corresponding initial energy. Therefore, we also need to deploy IoT devices as few as possible to cover the whole detection area and schedule the placement of IoT devices to minimize the total energy consumption such that the whole area is continuously observed T time slots by IoT devices. Therefore, we can find that the MAPS problem is also composed of two subproblems, minimum location area coverage (MLAC) and MESD, as shown in Definitions 18 and 9. The MESD problem has been solved by Algorithm 2. Thus, in this section, we first propose an approximation algorithm to solve the MLAC problem. Then, based on the problems MLAC and MESD, we propose an approximation algorithm to solve the MAPS problem.

We use $A(L_k)$ to denote the coverage region of L_k when the coverage range of IoT devices put on L_k is R, that is, for any point $p \in A(L_k)$, $d_{L_k,p} \leq R$. Let $D(L_k)$ be the border of $A(L_k)$.

Definition 18 (MLAC). Given a detection area $\mathscr{A}$, a set $L = \{L_1, L_2, \cdots, L_N\}$ of N candidate sites, $\Delta = \{A(L_1), A(L_2), \cdots, A(L_N)\}$, the minimum location area coverage (MLAC) problem is to find a minimum subset $L' \subseteq L$ of candidate sites such that $||\bigcup_{L_k \in L'} A(L_k) \bigcap \mathscr{A}|| = ||A||$.

In the following, we first introduce the definition of the Voronoi diagram of candidate sites on L, which is used to solve the MLAC problem.

6.1. Voronoi Diagram of Candidate Sites. The definition of the Voronoi diagram of candidate sites in L can be defined as the subdivision of the detection area $\mathscr{A}$ into N cells as shown in [31]. Any point $p \in \mathscr{A}$, in the cell corresponding to a candidate site L_k, is closer to L_k than to any other candidate site in L. Formally, the Voronoi cell corresponding to L_k can be defined as

$$\text{Cell}(L_k) = \bigcap_{k=1, j \neq k}^{n} \left\{ p \mid d_{L_k,p} \leq d_{L_j,p} \right\}. \tag{17}$$

In equation (17), two Voronoi cells meet along a Voronoi edge and three cells meet at a Voronoi vertex. For simplicity, we use $G(VV, VE)$ to denote the Voronoi diagram of candidate sites in L, where VV represents the set of Voronoi vertices and VE denotes the set of Voronoi edges.

6.2. Algorithm for the MLAC Problem. In this subsection, we propose an approximation algorithm to solve the MLAC problem, which is called the MLAC Algorithm (MLACA).

Before describing the algorithm, we introduce some definitions and parameters as follows:

```
**Input**: The dimensions of $\mathscr{A}$, the coverage range $R$, $L = \{L_1, L_2, \cdots, L_N\}$;
**Output**: $L'$;
Sets of $L' = L$, $VV = \varnothing$, $VE = \varnothing$, $NV(L_k) = \varnothing$ for any $L_k \in L$;
Compute the Voronoi diagram $G(VV, VE)$ and $Reg(L')$ of candidate sites on $L'$ by using algorithm in [32];
**for** any $L_k \in L'$ **do**
    Compute $NV(L_k)$ based on $G(VV, VE)$ and $Reg(L')$;
    Construct $G(VV(L_k), NE(L_k))$ on $NV(L_k)$ by using algorithm in [32];
    Compute $NI(L_k)$ based on $G(VV(L_k), NE(L_k))$;
    **if** $VV(L_k) \cup NI(L_k) \subset \bigcup_{L_j \in NV(L_k)} A(L_j)$ **then**
        $L' = L' \setminus \{L_k\}$;
        Compute the Voronoi diagram $G(VV, VE)$ of candidate sites in $L'$ by using algorithm in [32];
    **end**
**end for**
```

Algorithm 4: MLACA.

```
**Input**: $S = \{s_1, s_2, \cdots, s_n\}$, $E_i$ for each $s_i \in S$, $L = \{L_1, L_2, \cdots, L_N\}$, $R$, $T$;
**Output**: $C$, $M$;
**Step 1:** Compute the set $L'$ of candidate sites to cover the whole area by executing Algorithm 4;
**Step 2:** Compute $C$, $M$ based on $L'$ by executing Algorithm 2,
where $P = L'$;
```

Algorithm 5: MAPSA.

Definition 19 (neighbor Voronoi diagram). The Neighbor Voronoi diagram of a candidate site L_k is the Voronoi diagram of the Voronoi neighbors of L_k when L_k is excluded, where we call a site L_j a Voronoi neighbor of L_k if their cells share an edge.

Definition 20 (redundant candidate site). A candidate site L_k is said to be a redundant candidate site if $||\bigcup_{L_j \in L/L_k} A(L_j) \bigcap \mathscr{A}|| = ||A||$.

Definition 21 (neighbor Voronoi vertices). The neighbor Voronoi vertices of a candidate site L_k are the Voronoi vertices of the neighbor Voronoi diagram of Voronoi neighbors of L_k.

Definition 22 (neighbor Voronoi intersection vertices). Neighbor Voronoi intersection vertices of L_k are the intersections between edges of the neighbor Voronoi diagram and the circumcircle $D(L_k)$ of $A(L_k)$.

We use $\text{reg}(L) = \{\text{cell}(L_1), \text{cell}(L_2), \cdots, \text{cell}(L_N)\}$ to represent the set of all cells for $G(VV, VE)$. Let $NV(L_k)$ be the set of Voronoi neighbors of L_k. We use $G(VV(L_k), NE(L_k))$ to represent the neighbor Voronoi diagram of Voronoi neighbors of L_k, where $VV(L_k)$ denotes the set of the neighbor Voronoi vertices and $NE(L_k)$ represents the set of the Voronoi edge of $G(VV(L_k), NE(L_k))$. We use $NI(L_k)$ to be the set of the neighbor Voronoi intersection vertices of L_k.

Based on Theorem 10 in the work [31], we can obtain the following lemma.

Lemma 23. *A candidate site L_k is a redundant candidate site if and only if all the neighbor Voronoi vertices and neighbor Voronoi intersection vertices of L_k are covered by the Voronoi neighbors of L_k.*

In the following, we will introduce the detailed description of the MLACA algorithm.

Initially, we set $L' = L$, $VV = \varnothing$, $VE = \varnothing$, and $NV(L_k) = \varnothing$ for any $L_k \in L$. The execution of the algorithm consists of three steps as follows.

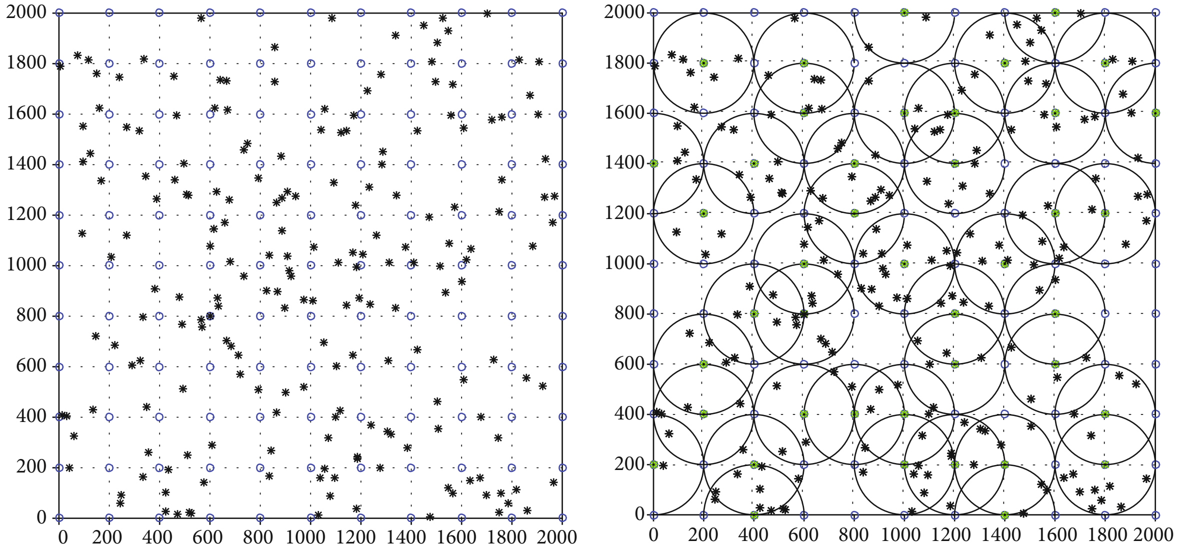
Firstly, we compute the Voronoi diagram $G(VV, VE)$ and $\text{reg}(L')$ of candidate sites in L' by using the algorithm in [32].

Secondly, we delete all redundant candidate sites from L' . For any $L_k \in L'$, we compute the set $NV(L_k)$ based on $G(VV, VE)$ and $\text{Reg}(L')$. Then, we construct the Voronoi diagram $G(VV(L_k), NE(L_k))$ on $NV(L_k)$ by using the algorithm in [32]. Next, we compute the set $NI(L_k)$ based on $G(VV(L_k), NE(L_k))$. Afterwards, we judge whether or not all points in $VV(L_k)$ and $NI(L_k)$ are covered by coverage regions of all Voronoi neighbors. If yes, we delete L_k from L' and update the Voronoi diagram $G(VV, VE)$ of candidate sites in new L' by using the algorithm in [32].

Finally, the algorithm returns L'.

The pseudocode of the algorithm is shown in Algorithm 4.

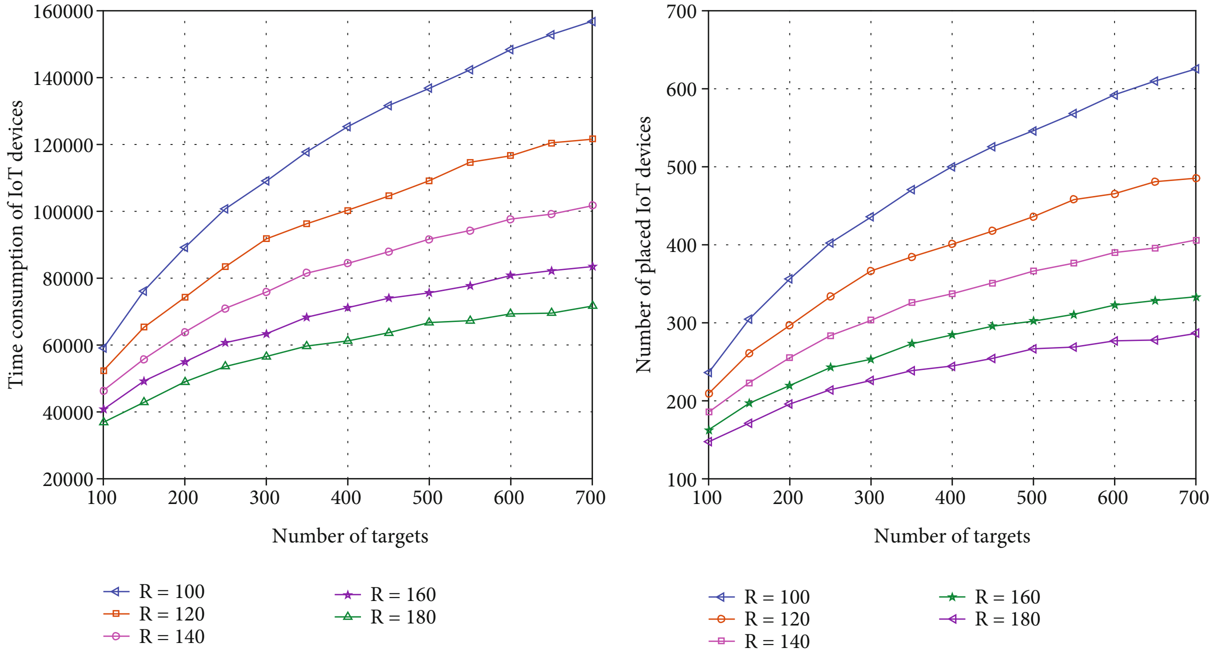
Theorem 24. *The time complexity of the MLACA algorithm is $O(N^2 \log N)$, where N represents the number of candidate sites in L.*



(a) The grid points as candidate sites

(b) The green points as placed sites

Figure 2: A coverage solution for an instance of the MTPS problem, where targets are deployed on the 2000 m × 2000 m detection area.



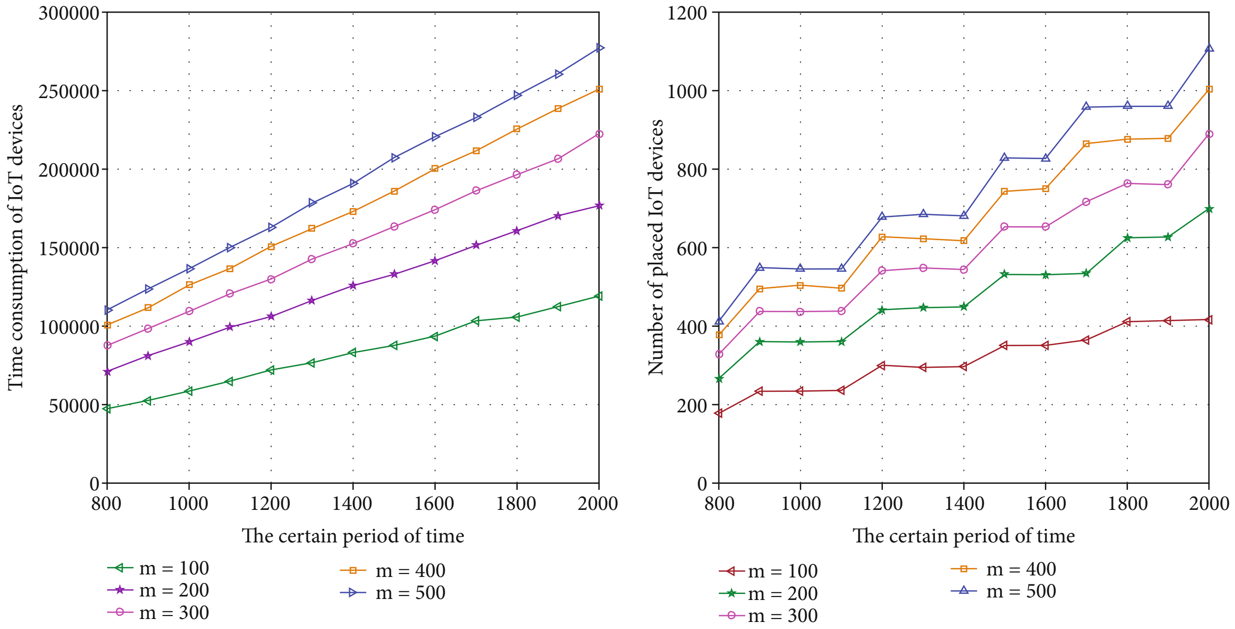
(a) Performance of MTPSA

(b) Effects of m and R

Figure 3: Simulations by varying m from 100 to 700 under different R.

Proof. According to the MLACA algorithm, we can find that the algorithm consists of three steps. Firstly, we compute the Voronoi diagram of candidate sites in L with $O(N\log N)$ time by using the algorithm in [32]. Secondly, we need at most N iterations in the for loop since $|L| = N$. In each iteration, we first need at most N times to compute the set $NV(L_k)$ of Voronoi neighbors of L_k. Then, we need at most $O(|NV(L_k)|\log|NV(L_k)|)$ time to compute the neighbor Voronoi diagram on $NV(L_k)$. Next, we need at most N times to compute $NI(L_k)$. We need at most another $O(N\log N)$ time to update the Voronoi diagram by using the algorithm in [32] when L_k is a redundant candidate site. Therefore, we need at most $O(N^2\log N) = O(N*(N+|NV(L_k)|\log|NV(L_k)|+N\log N))$ time for the for loop since $|NV(L_k)| \leq N$.

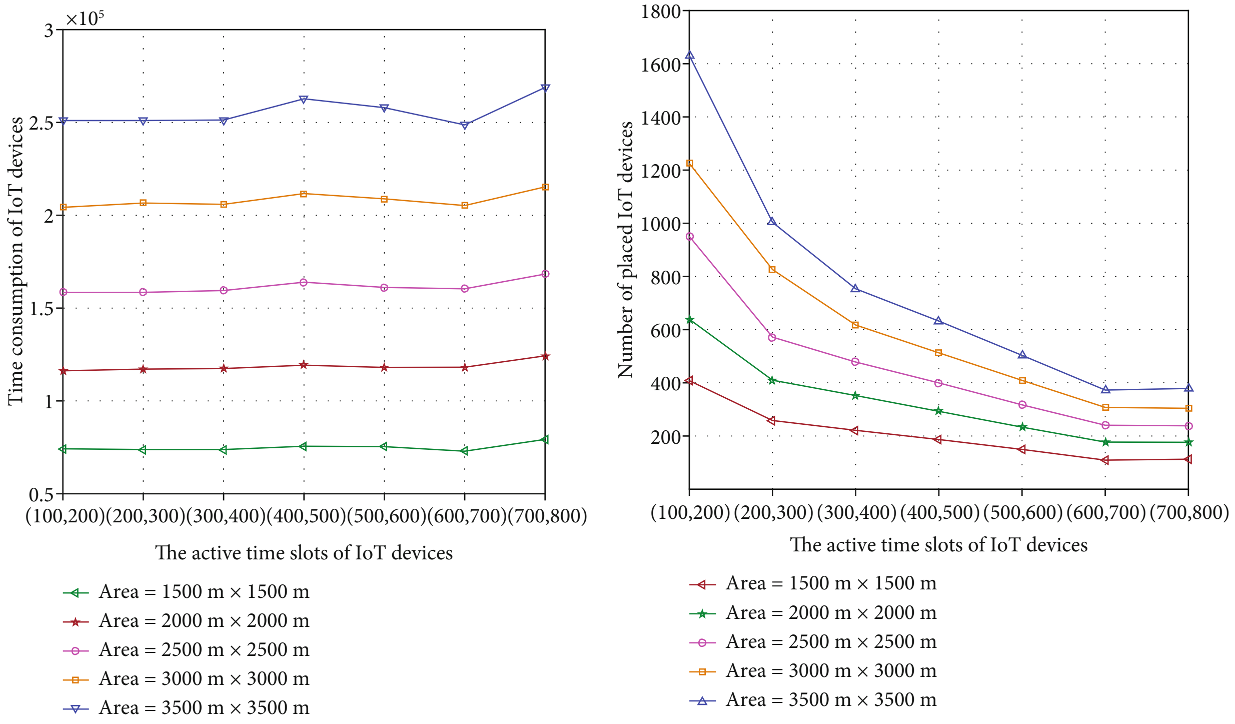
From what has been discussed, we can verify that the time complexity of the MLACA algorithm is $O(N^2\log N)$. □



(a) Performance of MTPSA

(b) Effects of T and m

Figure 4: Results by varying T from 800 to 2000 under different m.

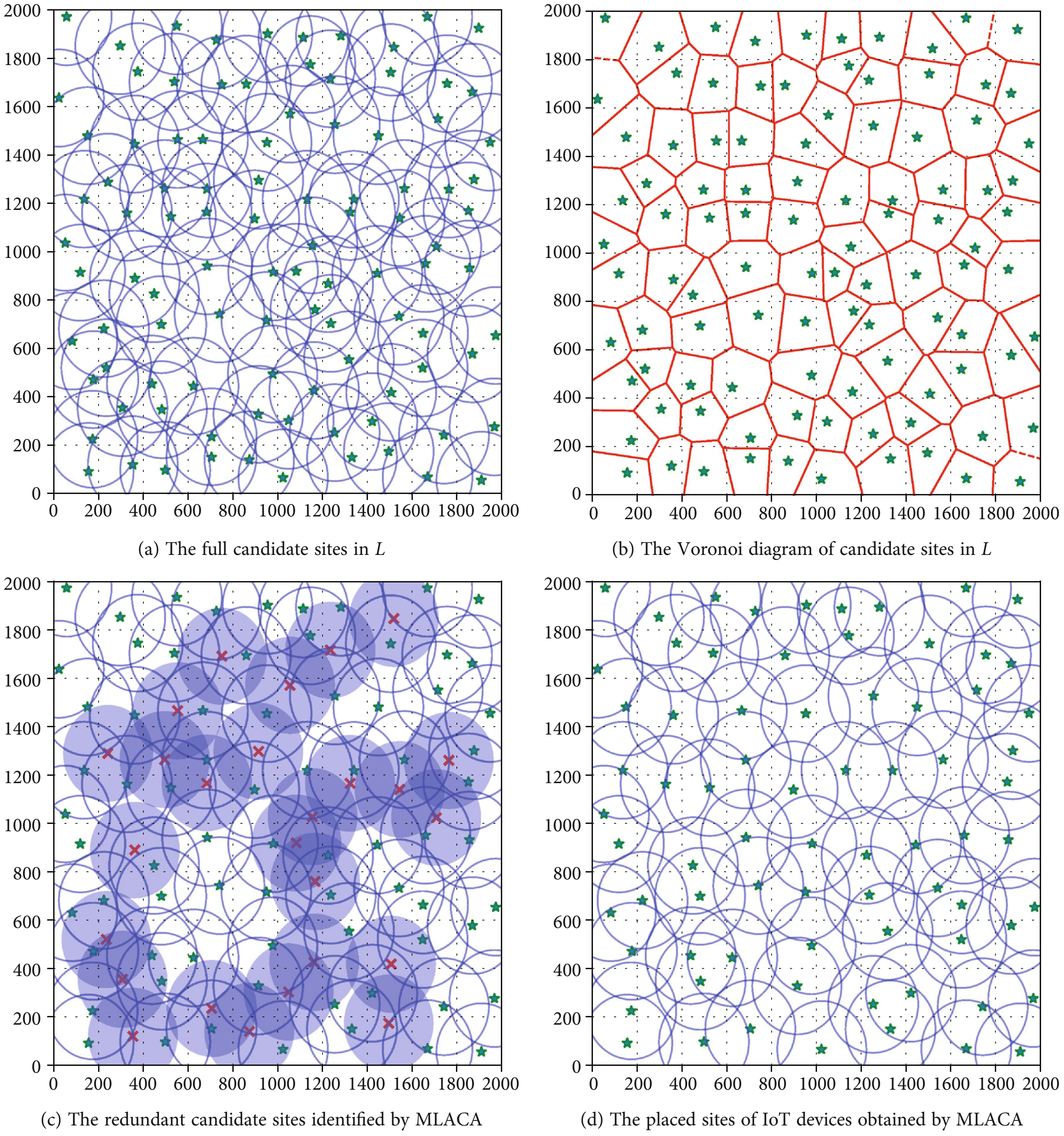


(a) Performance of MTPSA

(b) Effects of the size of the detection area and active time slots E_i

Figure 5: Simulations by varying E_i from [100, 200] to [700, 800] under different monitoring areas.

6.3. Algorithm for the MAPS Problem. In this subsection, we propose an approximation algorithm to solve the MAPS problem, which is called MAPSA. Similar to the MTPSA algorithm, the MAPSA algorithm consists of two phases. The first phase is to find a subset $L' \subseteq L$ of candidate sites by using Algorithm 4 such that the whole area $\mathscr{A}$ is covered by the coverage region of candidate sites in L'. The second phase is to find a subset $C \subseteq S$ of IoT devices placed at the sites in L' by executing Algorithm 2 such that for any point $p \in \mathscr{A}$, it is observed by IoT devices in C at least T time. The



(a) The full candidate sites in L

(b) The Voronoi diagram of candidate sites in L

(c) The redundant candidate sites identified by MLACA

(d) The placed sites of IoT devices obtained by MLACA

Figure 6: The executing process of the MLACA.

detailed illustration of the algorithm is shown in Algorithm 5.

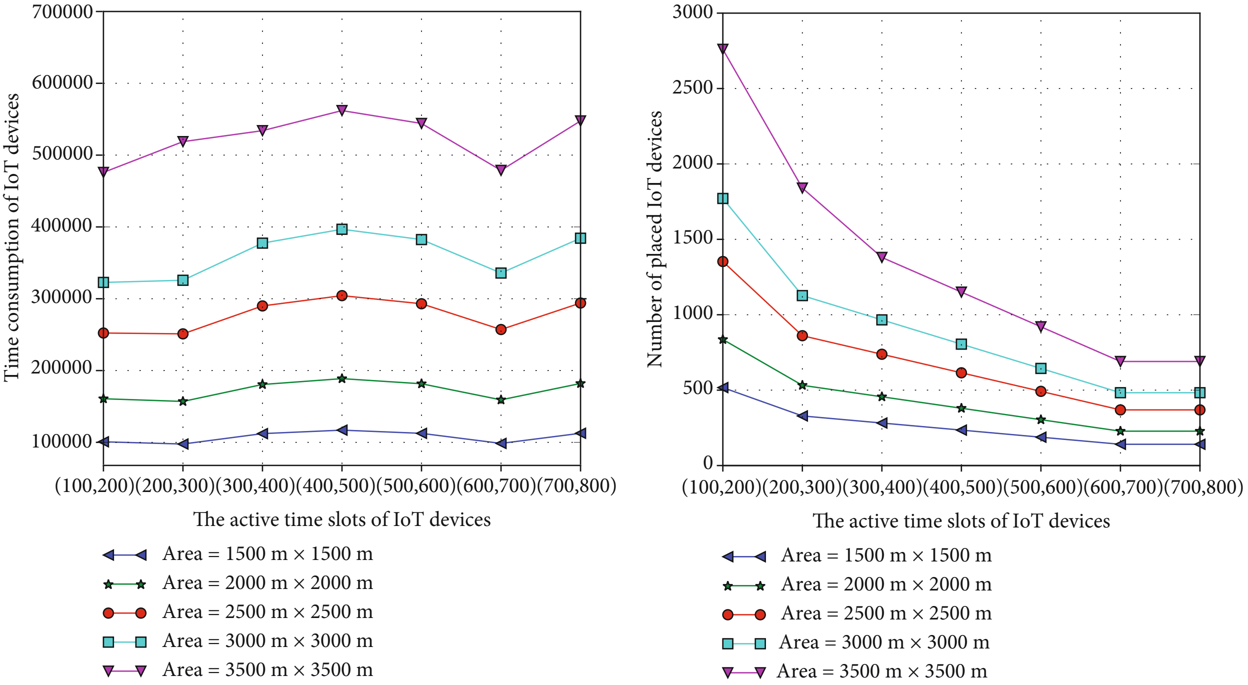
Theorem 25. *The time complexity of the MAPSA algorithm is* $O(n^3 + nN + N^2 \log N)$, *where n is the number of available IoT devices in S, and N represents the number of candidate sites in L.*

Proof. Based on Theorems 13 and 24, we can verify that the time complexity of the MAPSA algorithm is $O(n^3 + nN + N^2 \log N)$. □

7. Simulations

In this section, we evaluate the average performance of the approximation algorithms MTPSA and MAPSA depending on simulations with several critical performance metrics under different configurations. The code of the algorithms is implemented using MATLAB 2016a. For every group of parameter settings, we create 100 instances, execute the simulations, and obtain the average results.

7.1. Simulations for the MTPSA Algorithm. In order to ensure that all targets can be covered by candidate sites, we use a grid to cover the detection area where the distance



(a) Performance of MLACA

(b) Effects of active time slots E_i

Figure 7: Performance of MAPSA under different monitoring areas.

between neighbor grid points is R except for all grids in the boundary of the area. As an instance shown in Figure 2(a), all targets are randomly deployed on a 2000 m × 2000 m detection area and the grid points as the candidate sites are located on the area, where we set T =1000, n =2000, and R = 200 m; pick the active time slots E_i of each IoT devices $s_i \in S$ from range [100, 200]. After executing the algorithm MTPSA for the instance, we can verify that the number of the placed sites of IoT devices is 42, as the green nodes shown in Figure 2(b); the number of IoT devices placed at the sites is 252 and the total time consumption of all used IoT devices is $4.2072e + 04$.

In the following, we will evaluate how the network settings, such as the number of targets m, the certain period of time T, the active time slots E_i for each IoT device $s_i \in S$, the coverage range R and the size of the detection area, affect the performance of the MTPSA algorithm.

Firstly, we evaluate how the number of targets m and the coverage range R affect the performance of the MTPSA algorithm when we set $n = 1000$ and $T = 1000$; use the interval [200, 300] to pick a uniformly distributed random active time slots E_i for each IoT device $s_i \in S$; change R to 100, 120, 140, 160, and 180 m; and vary m from 100 to 700 in the 2000 m × 2000 m detection area, as shown in Figure 3. It is observed that the total time consumption of IoT devices increases with the increasing of m, as shown in Figure 3(a). This is because the number of IoT devices that are placed at candidate sites increases as m grows and the active time slots of IoT devices are fixed, as shown in Figure 3(b). Figure 3(a) also shows that the total time consumption of the used IoT devices decreases with increasing R since the less candidate sites are used to place IoT devices as R grows, which leads to a decline in the number of IoT devices being used, as shown in Figure 3(b).

Figure 4 illustrates the impact of the certain period of time T and the number of targets m on the performance of the MTPSA algorithm when we set $n = 2000$; pick the active time slots E_i of each IoT device $s_i \in S$ from [200, 300], $R =$ 100 m, m=100, 200, 300, 400, and 500; and change T from 800 to 2000 in the 2000 m × 2000 m detection area. Figure 4(a) shows that the total time consumption of used IoT devices is becoming larger with increasing T, since more IoT devices are needed to be placed on candidate sites for continuously observing targets as T increases when the active time slots of IoT devices are fixed, as shown in Figure 4(b). It is also observed that the performance gap is becoming smaller with the increasing of m. This is because the increase of the number of sites placing IoT devices is becoming smaller as more and more targets are randomly deployed on the detection area, which results in a reduction of the number of IoT devices placed at the sites, as shown in Figure 4(b). We also find that the number of used IoT devices increases as T grows since each IoT device $s_i \in S$ continuously works E_i time slots once it starts working.

Figure 5 evaluates the impact of the active time slots of IoT devices and the size of the detection area on the performance of the MTPSA algorithm when we set $n = 2000$, $R =$ 200 m, $m = 600$, and $T = 2000$; deploy targets in the monitoring area 1500 m × 1500 m, 2000 m × 2000 m, 2500 m × 2500 m, 3000 m × 3000 m, and 3500 m × 3500 m; and assign the active time slots of each IoT device $s_i \in S$ from [100, 200], [200, 300], [300, 400], [400, 500], [500, 600], [600, 700], and [700, 800], respectively. Figure 5(a) shows that the total time consumption of used IoT devices levels off when $E_i < [700,800]$ with increasing of the active time slots of IoT devices and increases as the size of the monitoring area increases. This is because when $E_i < [700,800]$, at least three IoT devices should be placed at each placement site to continuously observe target T time, which is obviously greater than T.

7.2. Simulations for the MAPSA Algorithm. In order to ensure that the whole area can be covered by candidate sites, we select candidate sites evenly in the monitoring area. As an instance shown in Figure 6(a), given a $2000\,\mathrm{m} \times 2000\,\mathrm{m}$ monitoring area, 100 candidate sites are located on the area and we set $R = 200$. The process of the MLACA algorithm is shown in Figures 6(b)–6(d). Firstly, we construct the Voronoi diagram of the candidate sites in L, as shown in Figure 6(b). Secondly, we find all redundant candidate sites from L such that all remaining candidate sites can cover the whole area, as the red points shown in Figure 6(c). Finally, we can obtain the placed sites of IoT devices, as shown in Figure 6(d).

In the following, we will evaluate how the network settings, the active time slots, and the size of the detection area affect the performance of the MAPSA.

Figure 7 evaluates the impact of the active time slots of IoT devices and the size of the detection area on the performance of the MAPSA algorithm when we set $n = 3000$, $R = 200\,\mathrm{m}$, and $T = 2000$; deploy targets in the monitoring area $1500\,\mathrm{m} \times 1500\,\mathrm{m}$, $2000\,\mathrm{m} \times 2000\,\mathrm{m}$, $2500\,\mathrm{m} \times 2500\,\mathrm{m}$, $3000\,\mathrm{m} \times 3000\,\mathrm{m}$, and $3500\,\mathrm{m} \times 3500\,\mathrm{m}$; and assign the active time slots of each IoT device $s_i \in S$ from [100, 200], [200, 300], [300, 400], [400, 500], [500, 600], [600, 700], and [700, 800], respectively. Figure 7(a) shows that the total time consumption of used IoT devices increases with the increase of the active time slots of IoT devices when $E_i < [400,500]$. This is because when $E_i < [400,500]$, at least five IoT devices should be placed at each placement site to continuously observe the area T time and their total energy consumption increases with E_i increasing. When $E_i > [400,500]$, the total time consumption decreases with the increasing of E_i, since as E_i grows, the less devices are needed for working a certain time. Figure 7(b) shows that the number of used devices decreases with the increasing of E_i for each device. We also find that the number of placed devices decreases as the size of area decreases.

8. Conclusion

In this paper, we investigate the problems minimum energy consumption of IoT devices for target coverage through placement and scheduling (MTPS) and minimum energy consumption of IoT devices for area coverage through placement and scheduling (MAPS), which focuses on finding the placement locations of IoT devices from candidate sites and scheduling them to cover all targets or the whole monitoring area such that all targets or the entire area are (or is) continuously observed for a certain period of time and the total energy consumption of the placed IoT devices is minimized. We first propose the mathematical models for the proposed problems and prove that they are NP-hard. Then, we propose an approximation algorithm for each of them. Finally, extensive simulation results are shown to further verify the performance of the proposed algorithm.

Data Availability

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

Disclosure

A preliminary version of this paper appeared in the WASA 2021.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities under grants 2021ZY88 and BLX201921 and also supported by the National Natural Science Foundation of China under Grant 62002022.

References

[1] T. Zhu, T. Shi, J. Li, Z. Cai, and X. Zhou, “Task scheduling in deadline-aware mobile edge computing systems,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4854–4866, 2018.

[2] Z. Cai and X. Zheng, “A private and efficient mechanism for data uploading in smart cyber-physical systems,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2018.

[3] D. Ismail, M. Rahman, and A. Saifullah, “Low-power wide-area networks: opportunities, challenges, and directions,” in *Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking*, New York, NY, USA, 2018.

[4] A. M. Yousuf, E. M. Rochester, B. Ousat, and M. Ghaderi, “Throughput, coverage and scalability of lora lpwan for Internet of things,” in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, Banff, AB, Canada, 2018.

[5] B. Buurman, J. Kamruzzaman, G. Karmakar, and S. Islam, “Low-power wide-area networks: design goals, architecture, suitability to use cases and research challenges,” *IEEE Access*, vol. 8, pp. 17179–17220, 2020.

[6] C. Luo, J. Yu, D. Li, H. Chen, Y. Hong, and L. Ni, “A novel distributed algorithm for constructing virtual backbones in wireless sensor networks,” *Computer Networks*, vol. 146, no. DEC.9, pp. 104–114, 2018.

[7] Q. Chen, Z. Cai, L. Cheng, and H. Gao, “Structure-free broadcast scheduling for duty-cycled multihop wireless sensor networks,” *IEEE Transactions on Mobile Computing*, p. 1, 2021.

[8] C. Luo, M. N. Satpute, D. Li, Y. Wang, W. Chen, and W. Wu, “Fine-grained trajectory optimization of multiple uavs for efficient data gathering from wsns,” *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 162–175, 2021.

[9] Z. Cai and Z. He, "Trading private range counting over big Iot data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019.

[10] C. Luo, Y. Hong, D. Li, Y. Wang, W. Chen, and Q. Hu, "Maximizing network lifetime using coverage sets scheduling in wireless sensor networks," *Ad Hoc Networks*, vol. 98, p. 102037, 2020.

[11] Y. Wang, S. Wu, Z. Chen, X. Gao, and G. Chen, "Coverage problem with uncertain properties in wireless sensor networks: a survey," *Computer Networks*, vol. 123, no. aug. 4, pp. 200–232, 2017.

[12] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, FL, USA, 2005.

[13] R. Akhlaghinia, S. Hashemi, and B. Shadgar, "Sensor Placement for Heterogenous Point Coverage," in *2010 Second International Conference on Computer and Network Technology*, Bangkok, Thailand, 2010.

[14] S. Mini, S. K. Udgata, and S. L. Sabat, "Sensor deployment and scheduling for target coverage problem in wireless sensor networks," *IEEE Sensors Journal*, vol. 14, no. 3, pp. 636–644, 2014.

[15] N. T. Hanh, H. T. T. Binh, N. X. Hoai, and M. S. Palaniswami, "An efficient genetic algorithm for maximizing area coverage in wireless sensor networks," *Information Sciences*, vol. 488, pp. 58–75, 2019.

[16] İ. K. Altınel, N. Aras, E. Güney, and C. Ersoy, "Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks," *Computer Networks*, vol. 52, no. 12, pp. 2419–2431, 2008.

[17] B. Wang, "Coverage problems in sensor networks," *ACM Computing Surveys*, vol. 43, no. 4, pp. 1–53, 2011.

[18] I. Gravalos, P. Makris, K. Christodoulopoulos, and E. A. Varvarigos, "Efficient gateways placement for Internet of things with qos constraints," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Washington, DC, USA, 2016.

[19] C. Jiang, Z. Chen, R. Su, and Y. C. Soh, "Group greedy method for sensor placement," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2249–2262, 2019.

[20] M. Z. Hasan and H. Al-Rizzo, "Optimization of sensor deployment for industrial Internet of things using a multiswarm algorithm," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10344–10362, 2019.

[21] J. A. Jiang, J. C. Wang, H. S. Wu et al., "A novel sensor placement strategy for an iot-based power grid monitoring system," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7773–7782, 2020.

[22] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, 2nd edition, 2001.

[23] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, "Efficient energy management in sensor networks," *Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing*, vol. 2, pp. 71–90, 2005.

[24] L. Ding, W. Wu, J. Willson, L. Wu, and W. Lee, "Constant-approximation for target coverage problem in wireless sensor networks," in *Proceedings IEEE Infocom*, pp. 1584–1592, Orlando, FL, USA, 2012.

[25] Z. Lu, W. W. Li, and M. Pan, "Maximum lifetime scheduling for target coverage and data collection in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 714–727, 2015.

[26] T. Shi, J. Li, H. Gao, and Z. Cai, "A novel framework for the coverage problem in battery-free wireless sensor networks," *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[27] G. Xing, C. Lu, R. Pless, and J. A. O'Sullivan, "Co-grid: an efficient coverage maintenance protocol for distributed sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pp. 414–423, New York, NY, USA, 2004.

[28] J. Yu, S. Wan, X. Cheng, and D. Yu, "Coverage contribution area based k-coverage for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8510–8523, 2017.

[29] N.-N. Qin and J.-L. Chen, "An area coverage algorithm for wireless sensor networks based on differential evolution," *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, 2018.

[30] U. Feige, "A threshold of lnnfor approximating set cover," *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[31] A. Boukerche and X. Fei, "A voronoi approach for coverage protocols in wireless sensor networks," in *IEEE GLOBECOM 2007- IEEE Global Telecommunications Conference*, pp. 5190–5194, Washington, DC, USA, 2007.

[32] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, vol. 501, John Wiley & Sons, 2009.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 4873574, 20 pages
https://doi.org/10.1155/2021/4873574

Research Article

CGPP-POI: A Recommendation Model Based on Privacy Protection

Gesu Li ,[1] Guisheng Yin,[1] Zuobin Xiong,[2] and Fukun Chen[1]

[1]*College of Computer Science and Technology, Harbin Engineering University, Heilongjiang, China*
[2]*Department of Computer Science, Georgia State University, GA, USA*

Correspondence should be addressed to Gesu Li; lgs7788@hrbeu.edu.cn

Received 17 June 2021; Accepted 18 August 2021; Published 31 August 2021

Academic Editor: Yan Huo

At present, with the popularization of intelligent equipment. Almost every smart device has a GPS. Users can use it to obtain convenient services, and third parties can use the data to provide recommendations for users and promote relevant business development. However, due to the large number of location data, there are serious data sparsity problems in the data uploaded by users. At the same time, with great value comes great danger. Once the user's location information is obtained by the attacker, severe security issues will be caused. In recent years, a lot of researchers have studied the recommendation of point of interests (POIs) and the privacy protection of location. Yet, few of them have explored both together, which induces some drawbacks on the combination of them. This paper combines POI recommendation with a privacy protection mechanism. Besides providing user with POI recommendation service, it also protects the privacy of user's location. We proposed a POI recommendation model with privacy protection mechanism, termed POI recommendation model for community groups based on privacy protection (CGPP-POI). This model can ensure the recommendation accuracy and reduce the leakage of user location information via taking advantages of the characteristics of location. At the same time, it deals with the problem of poor recommendation performance caused by sparse data. In addition, through the expansion of location, random and other methods are used to protect the user's real check-in information. First, the data processed at the terminal satisfied local differential privacy. At the same time, we use the data to build a recommendation model. Then, we use a community of user in the model to improve the availability of these disturbed data, explore the relationship between users, and expand check-ins within the community. Finally, we provide the POI recommendations to users. Based on the traditional evaluation criteria, we adopted four metrics, i.e., accuracy, recall rate, coverage rate, and popularity in evaluation part, where intensive experiments conducted on real datasets Gowalla and Brightkite demonstrate that our approach outperforms the baseline methods significantly.

1. Introduction

With the advent of 5G techniques, people's lifestyles have been changed thoroughly. Smartphones have become a ubiquitous part in our daily life, e.g., using smartphone to seek information, shopping online, and performing navigation. Hou et al. [1] pointed out that with the help of smart devices, sharing information has become a normal part of people's life. Especially during the epidemic, intelligent equipment has a great impact on China, for example, scanning QR codes to report trips and taking online classes. In short, it has accelerated the rapid development of e-commerce industry and internet technology and driven the overall economic development. But the convenient life requires users to provide more information, and the most useful information is GPS location. Location information is a very special data, because it has bonded with people's routine, e.g., navigation, shopping, and social activity, and thus has strong private attributes. Almost all software and applications require users to provide GPS information if they want to get a better experience. Currently, in order to get recommendation service, users still need to upload their exact GPS data to third parties. According to this location, the third party can provide nearby researched results. The exploration of POI recommendations is still in early stages, while, with the rapid development, people need to get more fresh information in a shorter time. For example, a person's daily pattern is two dots in a line, which means he/she is only active at home and work place. Suppose that one day he/she needs to go somewhere unfamiliar, searching on the internet for information is necessary. However, it is hard to figure out useful information among such a huge information flow. To solve this problem, recommendation system emerged, aiming at providing users with POI that they might be interested and saving searching time for them.

In recent years, recommendation research based on POI has focused on machine learning. Generally, the research on POI falls into the following categories: (1) tensor decomposition; (2) Markov chain model; and (3) neural network model. These POI recommendation algorithms have achieved good results in certain scenarios, but they lack generality. At the same time, they ignore the privacy issues caused by user trajectories. In the work of Xue et al. [2], the authors pointed that the friend relationship has an influence on users' future behavior, which shows that social information plays a role in the process of recommendation. Concurrently, the paper proposed that when excluding the influence of work location and family information, the influence of social relationship is more apparent. In addition, an attacker can infer a specific user by using friends and some background knowledge. Undoubtedly, social relationships indeed increase the risk of users' privacy leakage.

The development of GPS positioning and network technology not only brings convenience to users' lives but also increases the risk of users' location information leakage. Liang et al. [3] demonstrate that the prediction accuracy of tap position inference can be at least 90 percent by utilizing convolutional neural networks. As special information, location can be regarded as both public information and user's private information. For example, on a map, where every location is public, this information is available, and third parties can use this information and certain statistics to build user profiles. But there is no doubt that a specific user's trajectory is private. Once leaked, it will cause security and privacy risks to the user. If the leakage happens in the data collection stage, it will cause a huge disaster for both the company and the user, such as the AOL [4] and Netflix [5] data breaches, which cause immeasurable social and economic consequences.

For the preceding reasons, we should take into account the particularity of location information. We propose a POI recommendation model based on privacy protection, which explores the relationship between users and the range of check-in. For example, in Figure 1, there are three users, represented by red, yellow, and blue dots. Each user has a certain number of check-ins in a certain period of time. Intuitively, blue users and yellow users are more closely related because they are more likely to be in the same area. As we can see from the figure, each user's check-in is their exact coordinates. If the same check-in map is obtained by an attacker, the attacker can easily predict where a user will go in the future. But if we expand the scope of a user's check-in location, the amount of information it contains will be ambiguous. Take the large blue check-in in Figure 1 as an example. Within radius r_1, the check-in scope contains only one check-in. When it was extended to r_4, it had four check-ins. Therefore, if the r_4 range is uploaded for service, the probability of an attacker inferring a true check-in will be decreased. Therefore, the user's real location can be protected. Again, all the check-ins in Figure 1 and where they are located can be considered public information without identifying the user who visited them. This public information is useless to the attacker. But for users, these check-ins can provide them with more options for future utility. In addition, on the establishment of the recommendation model, we hope to enhance the relevance among users by expanding the scope of real check-in, as the blue and yellow users in Figure 1. However, when the scope is extended, as shown in the figure, many check-ins of yellow user and blue user are partitioned into the same range, which increases the connection between blue user and yellow user and adds stronger correlated options to the prediction. It also shows that in complex network, community is a suitable way to solve the relationship between users [6]. In this paper, we build a satisfactory mechanism to provide users with POI recommendations through the user check-in information, while preserving user privacy protection for location information. Experiments show that this model can provide users with good recommendation choices under the privacy protection mechanism. Meanwhile, this model is constructed in a hierarchical way which reduces the amount of computation and effectively reduces the time cost. The innovation points of this paper are as follows:

(i) Integrating community detection mechanisms with location network. The community detection model is improved according to the needs of this paper and is adapted to suit the location network. At the same time, it can solve the problem of data sparsity in location recommendation

(ii) Instead of the traditional knowledge graph built by all users, the knowledge graph is processed hierarchically. We explore the relationships between users, locations, and check-ins within the community

(iii) In the process of recommendation, privacy protection mechanism is considered. To deal with real location coordinates, we apply local differential privacy on it. At the same time, the privacy protection mechanism is also added at the end of the recommendation, protecting the user's real location and future travel safety.

The content of this paper is organized as follows: Section 2 is related work, discussing the current research status of POI from two aspects of recommendation model and privacy protection. Section 3 is the preliminary knowledge introduction of the model. The basic knowledge of community detection and LDP involved in the model is introduced. Section 4 is the problem definition of the model. Section 5 is the main part of the model, which is elaborated from two parts: privacy protection mechanism and recommendation. Section 6 is the experiment of the paper, including the introduction of data set, setting, evaluation standard, and the

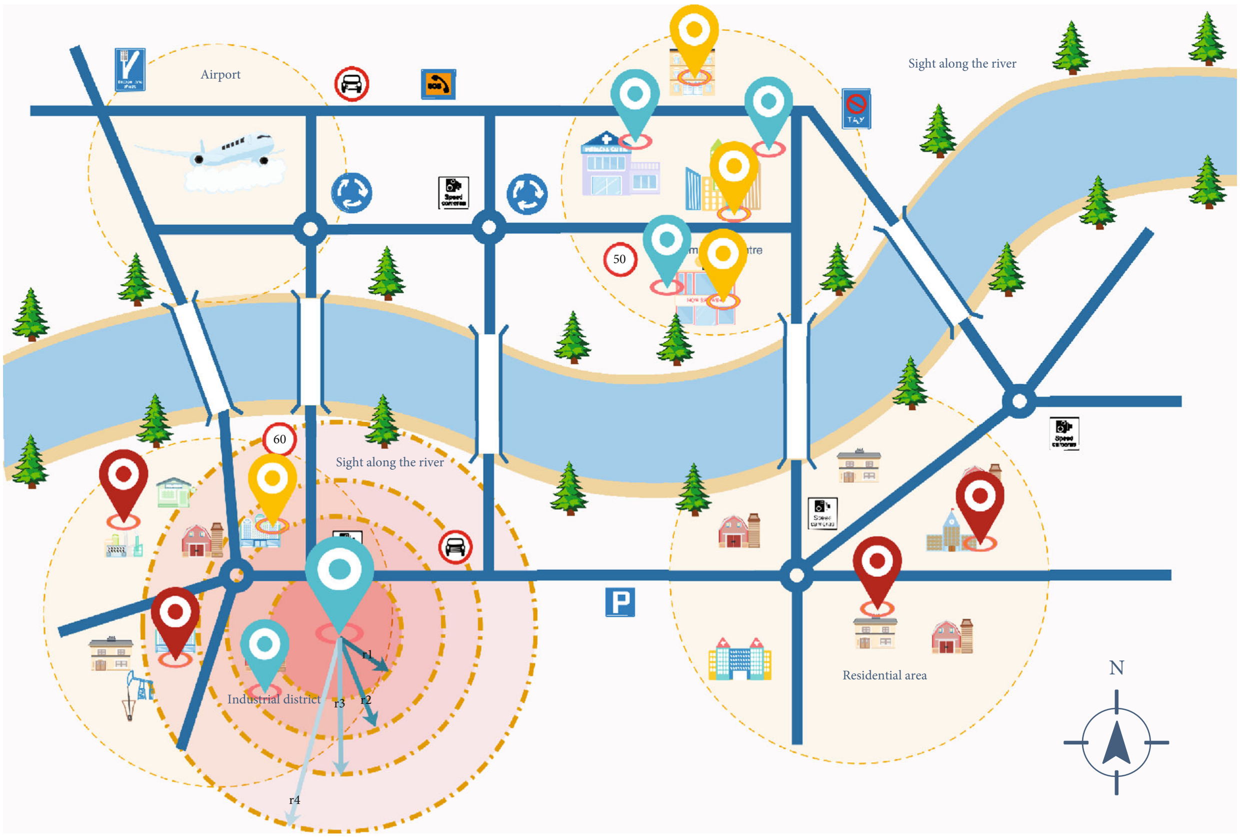


Figure 1: The example of users check-ins.

analysis of experimental results. Finally, the conclusion and future work of this paper is summarized in Section 7.

2. Related Work

In this section, we introduce the research on recommendation and privacy protection based on POI in recent years. We will discuss the current research status of these two aspects and point out the existing problems and the direction for improvement in this paper.

2.1. The Recommendation Model on POI. Recommendation based on POI has attracted much attention in recent years. Many scholars focus on solving the problem of data sparsity. In [7, 8], the authors mentioned that the density of check-in data is usually around 0.1%, while the data density of Netflix movie recommendations is around 1.2%. Data sparsity has a greater impact on POI recommendations. Zhao et al. [9] use context to predict the next POI through associative learning, in which authors construct an STGN to construct personalized sequences for users' long-term and short-term preferences. Li et al. [10] proposed a dynamic network recommendation algorithm based on HMM, which also includes privacy protection mechanism. Collaborative filtering, as a widely used recommendation algorithm, is also widely used in POI. Koren et al. [11–13] recommend algorithms based on matrix decomposition and its variants. Huang et al. [14] proposed a retrieval algorithm based on convolutional neural network, which is a recommendation technology for 3D and other Internet of Things devices, wherein Mnih and Salakhutdinov [13] proposed a PMF model containing Gaussian noise based on Bayesian framework. This model can maximize the posterior probability of potential features under the Gaussian hypothesis. Yin et al. [15] proposed an implicit probability generation model for the phenomenon of interest drift across geographic regions. This model can learn individual interests according to the check-in of individual interests in each region. It used social and spatial information to enhance regional dependence. At the same time, the author designed an effective pruning algorithm to overcome the dimension problem. Wang et al. [16] proposed a trust-based probabilistic recommendation model for social networks. In order to solve the problem of users' cold start, the authors consider their latent factor to find potentially similar users. The work of Li et al. [17] proposed a recommendation algorithm based on community division, which can also protect the privacy of users. At present, researchers have proposed a variety of solutions to the problem of data sparsity. With further understanding, most methods improve accuracy by adding features and increasing the complexity of the model. The result is an increased cost of time, which also requires users to upload more personal information when using third-party software to access the corresponding services. This is easy to cause the leakage of privacy.

2.2. The Privacy Protection on POI. Location-based services typically allow users to upload their current location first.

The third party obtains the information and provides recommendation service for the user according to the surrounding situation. Usually, the information uploaded by the user is exact location coordinates. In this paper, we denote this exact coordinate as L. But when a user provides location to a third party, there is no way to determine whether the third party can be trusted. At the same time, there is a risk of leakage during the process of information collection. Therefore, researchers have proposed some solutions to protect users' location privacy. K-anonymity is the most widely used privacy protection method. Kido et al. and Shankar et al. [18, 19] proposed a method to protect the user's identity information, so that the attacker cannot deduce which specific user is querying. Xue et al. and Bamba et al. [2, 20] focus on protecting the user's real location, where authors hide the user's real location by using virtual location. The preceding privacy protection methods can be boiled down to hiding. By hiding real data in other data, it prevents the attacker from obtaining the user's real location. There are other privacy protective mechanisms. Cover [21] introduced a stratagem-based protection mechanism. Papadopoulos et al. and Ghinita et al. [22, 23] proposed methods based on private information retrieval. Cai et al. [24] proposed a data sanitization method that collectively manipulates user profile and friendship relations. Besides sanitizing friendship relations, the proposed method can take advantages of various data-manipulating methods. Memon et al. [25] proposed a multimixed region privacy protection method which is dedicated to the mapping service of vehicles on the road network. Hashem and Kulik [26] proposed the use of mobile devices to form personal self-organizing networks. The authors proposed a decentralized method to protect the privacy of visitors' location. Chen et al. [27] proposed a method to protect user location privacy based on unobservability, in which the author designed, implemented, and evaluated a protection system named LISA, which is a location information scrambler. The system can adjust the location noise, protect the user's location privacy, and save the resources (especially the power) on mobile devices. Although these privacy protection mechanisms can protect the user's location security to a certain extent, they are lying under the assumption that the background knowledge of the attacker is limited, thus with some limitations. Some scholars have proposed spatial transformation, namely encryption, to protect the user's location information, yet these approaches are difficult to implement on mobile devices because of the huge computation cost even though the bandwidth has been increased with the help of 5G technology.

Moreover, more researches [28–32] have investigated the privacy protection problem in the background of 5G, IoT, and Big Data. As a representative of them, differential privacy [33], a privacy concept in the field of statistical databases, has been widely adopted. Its goal is to publish statistics about the database while protecting users' personal data. Differential privacy requires that modifying the data of a single user has negligible impact on the query results. Zhao et al. [34] proposed a top-down partitioning algorithm based on local differential privacy (LDP). It can generate undifferentiated location record data. At the same time, a new adaptive user assignment scheme and a series of optimization techniques are used to improve the accuracy of published data. It can be seen from the above studies that some researchers usually adopt mechanisms such as anonymity and disturbance to protect data privacy, and others also adopt encryption or differential privacy. This paper integrates multiple privacy protection mechanisms. We build protection mechanisms for data in the aspect of query, publication, and prediction, which can fully protect the location privacy of users.

3. Preliminaries

In this section, we introduce two main technologies covered in this paper, i.e., community detection and local differential privacy.

3.1. Community Detection. Community detection is typically applied to complex networks such as social networks [35], where each user is represented as a node. It focuses on the relationship between users. Tight-knit users are often divided into a community [36]. At present, communities are divided into overlapping communities and nonoverlapping communities. Overlapping community is more consistent with the division of communities in real life. Therefore, it is widely used. This paper makes an improvement on the traditional community detection by making it blend with the location characteristics. We choose the fast unfolding [37] algorithm as the basic algorithm of community detection and make improvement upon it. The algorithm has the advantages of fast computation and suitable for overlapping community division.

Fast unfolding algorithm is an iterative algorithm, which is mainly divided into two stages. The first stage is the module optimization stage. It mainly divides each node into the community where the adjacent nodes are located, thus increasing the value of modularity. The second stage is the community aggregation stage. It is to aggregate the communities divided in the first step into a single point. In other words, aggregation stage restructures the network. The algorithm is performed by repeating the above two steps until the network structure is stable. The formula involved is as follows:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{i,j} - \frac{k_i k_j}{2m} \right] \sigma(c_i, c_j), \quad (1)$$

where $m = 1/2 \sum_{i,j} A_{i,j}$ is all the weights in the network, $A_{i,j}$ is the weight between u_i and u_j, and $k_i = \sum_i A_{i,j}$ is the weight of the edge linked to the vertex. c_i is the community to which the vertex is assigned. $\sigma(c_i, c_j)$ is used to determine whether u_i and u_j are divided into the same community. If it is divided into the same community, it returns 1. Otherwise, it returns 0. Whether it is divided into a community depends on whether ΔQ is positive or not. In other words, if a node is divided into the existing community, the change of modularity is positive.

3.2. Local Differential Privacy. Dwork et al. [38] introduced the concept of differential privacy in 2006. The initial work is concerned with the processing of central difference privacy [33, 38, 39]. Differential privacy is a privacy preserving statistical query method based on statistics. The purpose is that even if all record in the database is changed, the statistical result will not change too much. One problem with centralized differential privacy, however, is that the third party must be trusted. In fact, the third party is not necessarily trustworthy in reality. To solve this problem, local difference privacy (LDP) is proposed. In LDP, the data is processed by the user locally and then uploaded to the central server; therefore, it provides better privacy protection. The definition of LDP is as follows.

A randomized algorithm f is $\epsilon-\mathrm{LDP}$ if for any two check-in records t and t', $t, t' \in Dom(f)$, and for any possible output $t* \in Ran(f)$, the following equation is satisfied:

$$\Pr\left[f(t)=t^*\right] \le e^{\epsilon} \times \Pr\left[f\left(t'\right)=t^*\right]. \tag{2}$$

The perturbation mechanism commonly used for LDP is random response technique. In LDP, each user perturbs their data and uploads it to the central processor. As a result, the data of any two users cannot be obtained from each other, so the concept of global sensitivity does not exist.

4. Problem Definition

The purpose of this paper is to predict POI for users in a recommendation model. Particularly, we use user location to build a recommendation mechanism. In addition to that, this model also needs to protect users' location privacy and to prevent an attacker from using all of the user's data to infer the real location. To facilitate better understanding of the model for readers, we define the concepts and formulate the problems involved in the model as follows.

Definition 1. Location network G. The location network graph based on user check-in is defined as G. $G=(V, E)$ is made up of nodes and edges, where V and E represent nodes and edges, respectively, wherein G is formed by n subgraphs. The n subgraphs are obtained by community detection. That is, each community forms a subgraph. $G=\{g_1, g_2, .., g_n\}$, $n \in |C|$. The node in the graph is user U.

$U=\{u_1, u_2, \cdots, u_i\}$, $i \in |U|$. Each user has its own check-in sequence. $u_i = L_{u_i} = \{l_1, l_2, \cdots, l_j\}$, $j=|L_{u_i}|$, $L_{u_i} \subseteq L$, where L is the check-in set of all user. However, in order to ensure the privacy of user location data, we mix up the order of check-ins before uploading. $L=(LOC,\ LOT)$, where LOC is the set of the check-in label in L, and LOT is the set of precise coordinates corresponding to the check-in label. The following formula is a summary of the relationships in the graph.

$$\begin{cases} G=(U, E) \\ G=\{g_1, g_2, \cdots, g_n\}, n=|C| \\ U=\{u_1, u_2, \cdots, u_i\}, i=|U| \\ u_i=\left\{l_1, l_2, \cdots, l_j\right\}, j=|L| \\ L=(LOC, LOT) \\ loc_m=(lat_m, lon_m). \end{cases} \tag{3}$$

Definition 2. $L, \widehat{L}, \tilde{L}$. In order to ensure the privacy of user location information, we mix up the order of check-ins before uploading. Here, we focus on the following three check-in definitions. (1) L is the real check-in set of users; (2) $\widehat{L}$ is the disturbed check-in set uploaded; and (3) $\tilde{L}$ is the expanded check-in set of true check-in. The relationship between the three check-in sets is as follows.

$$\begin{cases} L=\left\{l_1, l_2, \cdots, l_j\right\}=\widehat{L}=\left\{\widehat{l}_1, \widehat{l}_2, \cdots, \widehat{l}_j\right\}, l_j \neq \widehat{l}_j, j=|L| \\ R=\{r_1, r_2, \cdots, r_m\}, m=|R| \\ \tilde{l}_q=l_j \times r_m=\{l_1, l_2, \cdots, l_k\}, l_j \in L, r_m \in R, \tilde{l}_q \subseteq L \\ P\left(\widehat{l}_j=l_k\right)=\dfrac{1}{\left|\tilde{l}_q\right|}, l_k \in \tilde{l}_q \\ \tilde{L}=\left\{\tilde{l}_1, \tilde{l}_2, \cdots, \tilde{l}_q\right\}, q=\left|\tilde{L}\right| \le L, \end{cases} \tag{4}$$

wherein the extended check-in set $L\sim$ is all the check-ins within this range obtained by taking the real check-ins l_j as the center and r_m as the radius. The perturbation check-in $l\hat{}_j$ is the replacement check-in randomly selected within the perturbation range of the real check-in. So, even though the set of true check-ins and the set of perturbed check-ins have the same content, each perturbed check-ins are a random substitution of its corresponding true check-ins. Take u_i as an example. L_{ui} and $L\hat{}_{ui}$ are both subsets of the check-in set, but these two sets are not the same. The elements in the perturbation set $l_u\hat{}_{i,j}$ are randomly selected and replaced from all the check-ins within the radius of r_m centering on the real check-in $l_{ui,j}$. $l_u\hat{}_{i,j}$ is randomly selected from the extended set, used to replace the original check-in l_j. The other real check-ins of u_i are replaced by the above method. Finally, the perturbation set $L\hat{}_{ui}$ of u_i is obtained.

$$\begin{cases} \widehat{l}_{u_i,j} \in \widehat{L}_{u_i}, \widehat{l}_{u_i,j} \in \widehat{l}_{u_i,j} \\ L_{u_i} \neq \widehat{L}_{u_i} \\ L_{u_i}, \widehat{L}_{u_i} \subseteq L \\ \tilde{l}_{u_i,j}=l_{u_i,j} \times r_m=\{l_1, l_2, \cdots, l_k\}, \tilde{l}_{u_i,j} \subseteq \tilde{L}, l_{u_i,j} \in L_{u_i} \\ P\left(\tilde{l}_{u_i,j}=l_k\right)=\dfrac{1}{\left|\tilde{l}_{u_i,j}\right|}, l_k \in \tilde{l}_{u_i,j}. \end{cases} \tag{5}$$

Definition 3. Similarity. Community detection is based on close relationships between users. We use the similarity between users as the weight, which is used to indicate the closeness of the relationship between users. The higher similarity implies the closer relationship, and the more likely it is to be divided into the same community. Specially, the Jaccard similarity index is used to measure the similarity between two users.

$$Sim(i,j)=\frac{\tilde{L}_{u_i}\cap\tilde{L}_{u_j}}{\tilde{L}_{u_i}\cup\tilde{L}_{u_i}}. \tag{6}$$

In Equation (6), $Sim(i,j)$ represents the similarity between u_i and u_j. $\tilde{L}_{u_i}$, $\tilde{L}_{u_j}$ are the check-ins of extended scopes u_i and u_j, respectively. The numerator is the check-in that both u_i and u_j have visited, and the denominator is the check-in set of two users.

Definition 4. Community. The first step of community detection is to establish an adjacency matrix between users. Different from existing works, this paper takes users and their check-in as the research object. There is no directionality between users. At the same time, the similarity of check-in between users replaces the weight of edge in traditional social network. That is, the higher the similarity between u_i and u_j, the easier it is to be divided into the same community. The node of the community is the user. C is the set of all communities. $C=\{c_1,c_2,\cdots,c_n\}, n\in|C|$. $c_n=\{u_i,\cdots,u_j\}$. According to the definition of community detection in this paper, the formula of modularity is improved, and the improved formula of modularity is as follows:

$$Q=\frac{1}{2m}\sum_{i,j}\left[Sim(i,j)-\frac{\sum_j Sim(i,*)\sum_i Sim(j,*)}{2m}\right]\sigma\left(c_i,c_j\right). \tag{7}$$

Definition 5. Incidence matrix. M is the relationship matrix between the user and the extended check-in $\tilde{L}$. The weight in M is the number of paths that the user reaches $\tilde{L}$ after passing k hops. Since each community is a subgraph, the relationship can be formulated as follows.

$$\begin{cases} m_n\subset M, n\in|C| \\ M=U\times\tilde{L}. \end{cases} \tag{8}$$

Definition 6. Privacy. For any user, there is a privacy algorithm f. Its domain is $Dom(f)$, and range is $Ran(f)$. If the algorithm f gets the same output t^* on any two records t and t', it is said that function f satisfies $\epsilon-$LDP.

$$\Pr\left[f(t)=t^*\right]\le e^{\epsilon}\times\Pr\left[f\left(t'\right)=t^*\right]. \tag{9}$$

Then, the relationship amid perturbation check-in, radius, and target check-in is given as follows.

$$\begin{gathered} P\left(\hat{l}_j=l_k\right)=\frac{1}{\left|\tilde{l}_q\right|}, l_k\in\tilde{l}_q, \\ \tilde{l}_q=l_j\times r_m=\{l_1,l_2,\cdots,l_k\}.k=\left|\tilde{l}_q\right|, \end{gathered} \tag{10}$$

wherein $\tilde{l}_q$ is the set of check-ins that l_j gets according to r_m. The intensity of privacy protection is determined by r_m and the number of check-ins contained within its scope, where the range of r_m is set by the user. In other words, the bigger range, the more check-ins in $\tilde{l}_q$, and the stronger the perturbation. $\hat{l}_j$ is a random selection of perturbation data from $\tilde{l}_q$ to replace l_j.

Since it is a random replacement of the original check-in, the randomness satisfies the random disturbance mechanism of LDP. This approach takes advantage of the scalability of the location. The statistical results within a certain range are hardly affected by the expansion of the range. That is, data characteristics are preserved. In other words, the data has good utility. According to this algorithm, $\epsilon=\ln\left(1/(|\tilde{l}_q|-1)\right)$ can be obtained. The data uploaded by each user is $\hat{L}_{u_i}$, which is the perturbed check-in. In the third-party statistical process, it satisfies the LDP, and the statistical results obtained after perturbation are basically the same as the real ones.

The specific derivation formula is given by the following example.

Suppose there are n users using the software, where the true percentage of users check-in at A is π. The third party hopes to get the true number of users who check in at A through calculating the data uploaded by users for answers. If the probability of A being checked in the true answer of the data uploaded by u_i is $1/k$, the probability of getting the other answer is $1/(k-1)$. According to the answers of n users, the number of people who have checked in A can be derived. If the statistics show that the number of people who have been to A is n_1, the number of people who get the other result is $(n-n_1)$. According to this statistic, the likelihood function LH is constructed.

$$LH=\left[\pi\frac{1}{k}+(1-\pi)\left(1-\frac{1}{k}\right)\right]^{n_1}\left[(1-\pi)\frac{1}{k}+\pi\left(1-\frac{1}{k}\right)\right]^{n-n_1}, \tag{11}$$

where the maximum likelihood estimate of π is obtained after the logarithmic derivative:

$$\hat{\pi}=-\frac{kn_1+n-nk}{nk-2n}. \tag{12}$$

By further solving the mathematical expectation of π, we can verify that the above estimation is an unbiased estimation of π. Therefore, the number of people who have been to A is calculated as follows:

$$N = \widehat{\pi} \times n = \frac{1-k}{2-k} n + \frac{kn_1}{2-k}. \quad (13)$$

Accordingly, based on the total number of people, we can get the number of people who went to A through perturbation probability. If we get that the number of people who have been to A is n_1, and the perturbation probability $1/k$, we can get the real number of users who have been to A. By definition, the privacy budget of this inference result is calculated by Equation (14):

$$\epsilon = \ln \frac{1}{k-1}, \quad (14)$$

where k is the number of elements in $\tilde{l}_j$, $k \in (1, +\infty)$.

5. CGPP-POI

As a special information, location data is of great significance to users. Using location properly can provide great convenience for users' daily life and can also bring more benefits to the third party. But location is also important privacy information because it records the user's trajectory. Once the information is obtained by the attacker, the user's private property and even personal safety will be threatened. Considering the particularity of location information, when we build the recommendation model, we should add privacy protection mechanism to the user location. In this work, the privacy of users is protected from three aspects: publication, inquiry, and recommendation. We propose a recommendation model based on privacy protection, namely the CGPP-POI model. This model builds the privacy protection mechanism based on LDP, which can effectively prevent the leakage of real location. At the same time, by extending the original location range, the coincidence degree of activity range among users is improved. Moreover, the user community is divided based on this, leveraging communities to extend user data to address sparse data and provide users with POI recommendation service. Figure 2 shows the overall framework of CGPP-POI, where the operation of LDP is in the blue dotted box. This part processes the original data of the user at the user side and uploads the data after adding perturbation. The perturbed data can be used for third-party statistical queries and further recommendation model use because of the satisfaction of $\epsilon-$ LDP. The middle part is the POI recommendation model. At the bottom part, the specific recommendation of the POI check-ins is also random. So even if an attacker has a prediction result, they cannot accurately infer the user's future location.

The following section we will introduce the CGPP-POI model from two aspects, privacy protection mechanism and recommendation model.

5.1. Privacy Protection Mechanism. Figure 1 shows the check-in information of three users; all of them have checked in at different ranges. From these check-ins, we can see the range of activities that each user often does. As can be seen from the activity range, the activity range of blue user and yellow user is highly overlapped. And as we can see from the lower left corner of the range area, the larger the scope, the more check-ins it covers, centered on one of the blue users' check-ins. These collections of check-ins can be considered public information. However, in the user data, it is regarded as the user's private information. For example, when we open Google Map, we can see many places marked, such as schools, hotels, and parks. These are considered public data. However, in the user's check-in records, they are the user's private information. We will use this public information to predict the user's POI. L is the set of these check-ins. Take the blue user u_i in Figure 1 as an example. We expand the blue check-in at the lower left. When $r = r_2$, $\tilde{l}_{u_i,j}$ contains two real check-ins, $r = r_3$, $|\tilde{l}_{u_i,j}| = 3$. When u_i selects r_3, we randomly choose $\widehat{l}_{u_i,j}$ from $\tilde{l}_{u_i,j}$ to replace l_j. The other check-ins for the u_i are replaced as described above then uploaded to the third-party unified collection.

Algorithm 1 satisfies the stochastic perturbation mechanism of LDP, and the derivation process is shown in Definition 6. Except LDP processing before uploading, the specific check-in mechanism is also added to the random in the recommended stage. The purpose of this is to prevent attackers from using the recommendation results to attack users. The pseudo code of related algorithm privacy protection of CGPP-POI is as follows.

5.2. CGPP-POI Recommendation Model. The data of each user is processed by Algorithm 1 and uploaded to the third party. The third party collects the data of all users, analyzes the data information, and constructs the recommendation model. The recommendation model are built based on location scalability characteristics. First of all, the data are extended uniformly. There are several reasons for this: (1) generalize the check-in coordinates to increase data information; (2) with extended check-in, the overlapping information among users increases; and (3) the extended scope may contain the real data of the user, which reduce the interference of disturbed data. In other words, $l_{ui,j}$ and $l_{u\,\hat{}\,i,j}$ may be the same extended check-in $l\sim_j$. So, there is more real information in the data, and it is easier to predict u_i's real preferences. Equation (6) is used to obtain the similarity between users. It acts as the weight value between users in the adjacency matrix. Through community detection, users with high similarity are divided into the same community, which can reduce data sparsity. According to the users and the check-in $L\sim$ within the community, the bipartite graph is constructed to obtain the relationship matrix m_n. The recommendation $LA_{ui} = \{l\sim_1, l\sim_2, \cdots, l\sim_j\}$ is the recommendation list for u_i, which includes j extended check-ins $l\sim_j$, and each $l\sim_j$ includes k -specific check-ins. To obtain the recommended list A of extended check-ins by sorting algorithm, we are going to pick a random number of these k check-ins and collect them into the recommendation list B for u_i. The recommendation algorithm of CGPP-POI is described as follows:

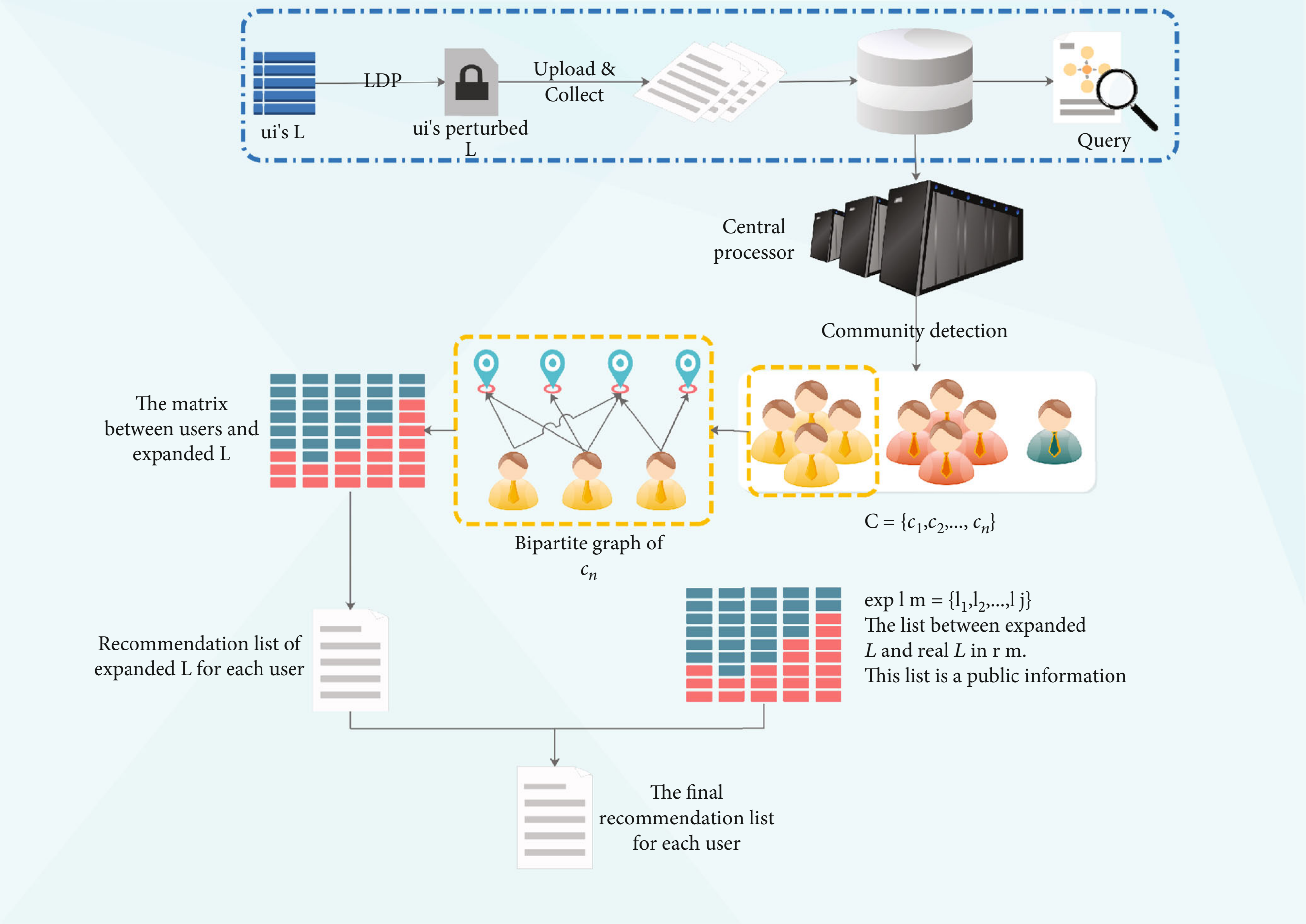

LDP
Upload & Collect
ui's L
ui's perturbed L
Query
Central processor
Community detection
The matrix between users and expanded L
Bipartite graph of c_n
C = {c_1,c_2,..., c_n}
exp l m = {l_1,l_2,...,l j}
The list between expanded L and real L in r m.
This list is a public information
Recommendation list of expanded L for each user
The final recommendation list for each user

Figure 2: The framework of CGPP-POI.

```
Input: $l_{u_i} = \{l_{u_i,1}, l_{u_i,2}, \cdots, l_{u_i,j}\}$;
       $r = \{r_1, r_2, \cdots, r_m\}$;
       $\tilde{L} = \{\tilde{l}_1, \tilde{l}_2, \cdots, \tilde{l}q\}$
Output: $\widehat{l}_{u_i} = \{\widehat{l}_{u_i,1}, \widehat{l}_{u_i,2}, \cdots, \widehat{l}_{u_i,j}\}$
  $u_i$ selects the extension radius $r_m$
  for $j \in l_{u_i}$ do
      $\tilde{l}_{u_i,j} = l_{u_i,j} \times r_m = \{l_1, l_2, \cdots, l_k\}$
      $P(\widehat{l}_{u_i,j} = l_k) = 1/k$
      $\widehat{l}_{u_i,j}$ replace $l_{u_i,j}$
  end for
  Get $\widehat{l}_{u_i} = \{\widehat{l}_{u_i,1}, \widehat{l}_{u_i,2}, \cdots, \widehat{l}_{u_i,j}\}$
```

Algorithm 1: Privacy protection of CGPP-POI.

6. Experiments

6.1. Datasets. The datasets we use in this paper are the Gowalla dataset and Brightkite dataset. They are real-world location-based social network, and both of them contain friend relationship information. However, in the preprocessing stage, we only retain the user's check-in information. The check-in information includes the check-in label *LOC*, the corresponding coordinates *LOT*, and the time information. The time information is kept for experimental purposes only and used to divide the train set and test set. The time context is not involved in data publishing and POI recommendation. Table 1 shows the collated information for the two datasets.

The Gowalla dataset is a location-based social network sourced from Stanford University and collected using public APIs. It is an undirected network containing users' location information by check-ins. It consists 196,581 nodes and 950,327 edges. The data were collected from February 2009 to October 2010. A total of 6,442,890 users were collected. The raw data consists two texts, one is the user's friendship and the other is the user's check-in information. The check-in information includes the user ID, check-in time, check-in label, and the corresponding exact coordinates.

The Brightkite dataset is also a location-based social network. Similar to Gowalla, users can check in to share their location. The data is collected based on the site's public API. It contains 58,228 nodes and 214,087 edges. The network was originally a directed graph, but the collector made it an undirected network. In this paper, friend

```
Input: $G = (\widehat{U}, E)$;
       $\widehat{L}_{u_i,j} = \{\widehat{l}_{u_i,1}, \widehat{l}_{u_i,2}, \cdots, \widehat{l}_{u_i,j}\}$;
       $\tilde{L} = \{\tilde{l}_1, \tilde{l}_2, \cdots, \tilde{l}_m\}$;
       $\tilde{l}_q = \{l_1, l_2, \cdots, l_k\}$;
Output: Recommendation list $LB_{u_i}$;
To choose the extend range $r_m$
$\tilde{L} = \{\tilde{l}_1, \tilde{l}_2, \cdots, \tilde{l}_m\}$
for $i \in U$ do
        $\tilde{L}_{u_i} = \{\tilde{l}_{u_i,1}, \tilde{l}_{u_i,2}, \cdots, \tilde{l}_{u_i,j}\}$
end for
$Sim(i, j) = \tilde{L}_{u_i} \cap \tilde{L}_{u_j} / \tilde{L}_{u_i} \cup \tilde{L}_{u_j}$
$G' = G \cap Sim$
Initialize each node to a separated community
repeat
    for $i \in V$ do
        for $j \in V$ do
            Remove i from its community, place to $j$'s community
            Compute the composite modularity gain $\Delta$
        end for
        Choose $j$ with maximum positive gain (if exists) and move $i$ to $j$'s community
        Otherwise, $i$ stays in its community
    end for
until No further improvement in modularity
Get $C = \{c_1, c_2, \cdots, c_n\}$, which are also independent subgraphs
if the number of users in $c_n > 1$ then
    $m_{c_n} = U_{c_n} \times \tilde{L}_{c_n}$
    Sort the recommended label for each user, and get recommendation list $LA_{u_i}, \tilde{l}_j \in LA_{u_i}$
else
    Sort by number of check-ins $\tilde{L}_{u_i}$ based on $u_i's$ history. And take the first $n$ and build the recommendation list $A$.
    end if
for $i \in U$ do
    for $j \in LA_{ui}$ do
        $P(l_j = l_k) = 1/|\tilde{l}_j|$, to choose n $l_j s$, and $LB_{u_i}$ is obtained.
    end for
end for
```

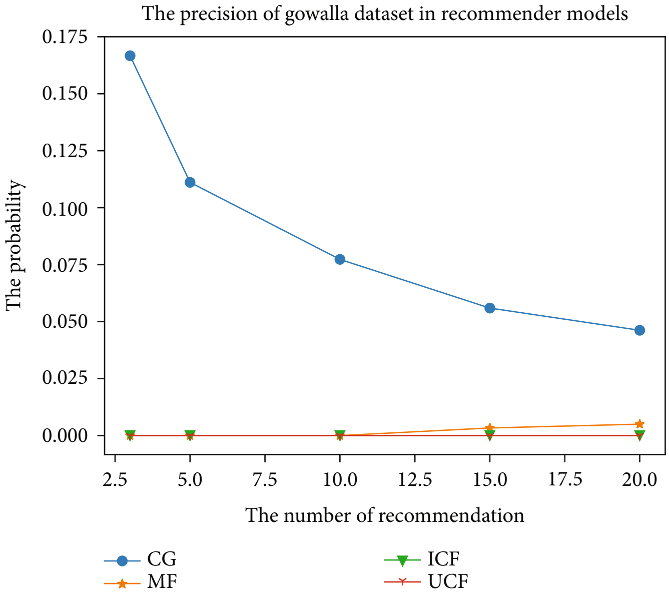
Algorithm 2: The recommendation of CGPP-POI.

Table 1: General statistics about the two datasets.

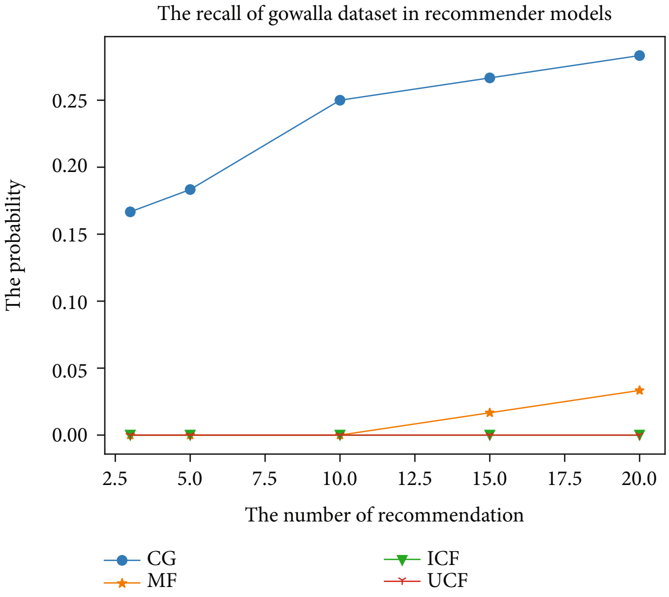
Network property	Gowalla	Brightkite
Number of nodes	196,591	58,228
Number of edges	950,327	214,078
Number of check-ins	6,442,890	4,491,143
Average number of check-ins per user	254	92
Maximum check-in number of user	2175	2100
Minimum check-in number of user	1	1

links are not used and will be deleted during preprocessing. The data was collected from April 2008 to October 2010, and a total of 4,491,143 users were collected. The original data set contains two files, one is a user's check-in file and the other is a friend relationship file. We only keep the first file for experiments. The check-in file contains the user ID, check-in time, specific latitude and longitude information, and check-in label.

6.2. Evaluation Methods. The main research objective of this paper is POI recommendation. Meanwhile, privacy protection is added to the original data to ensure the privacy of user's location. In terms of recommendation, the evaluation criteria for the quality of traditional models are usually taken as the evaluation index from four aspects: accuracy, recall rate, coverage rate, and popularity. POI recommendation, in principle, is roughly the same as item recommendation. The accuracy measures whether the prediction of the user's future behavior at the next moment is accurate. The recall rate judges whether the same check-in is visits multiple times. Popularity can be thought of as areas where users like to go. These measurements have significant guidance for the construction of other project models by third parties. At the same time, for users, people generally like going to places with many people, e.g., business districts and tourist areas. Therefore, adding these recommendations to the list provides a convenient service for users. The specific formulas are as follows:

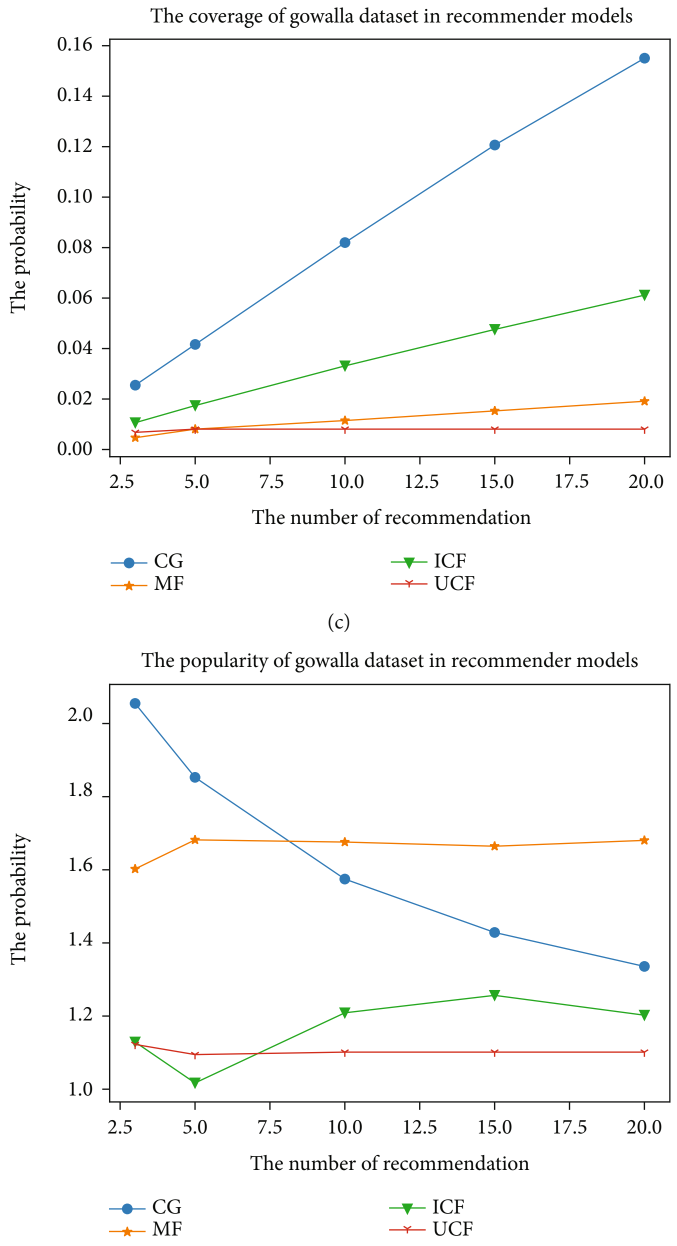


(a)



(b)

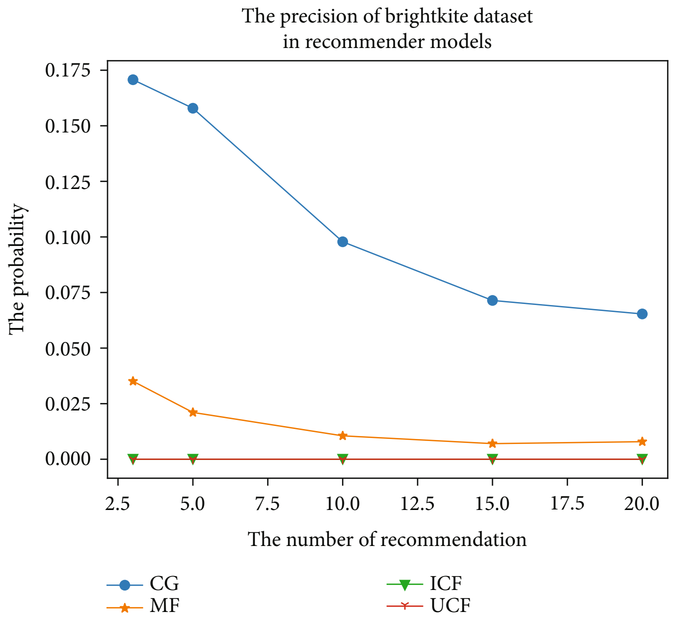
Figure 3: Continued.



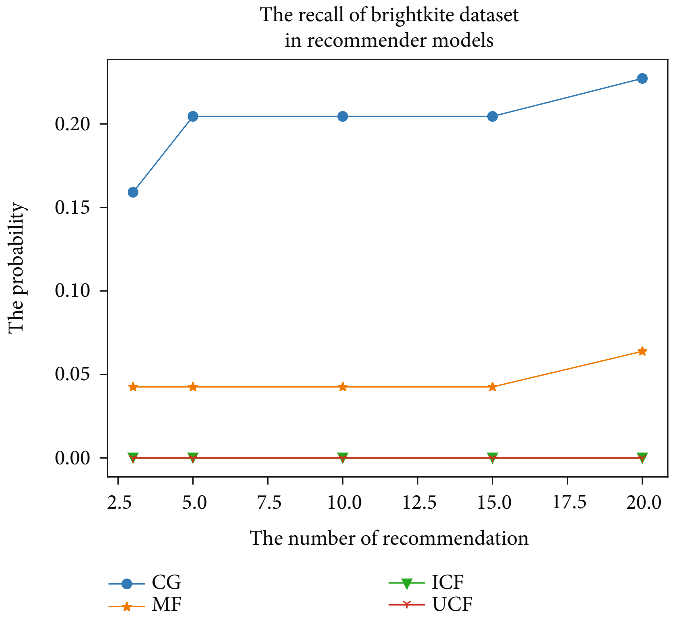
(c)

(d)

Figure 3: Continued.

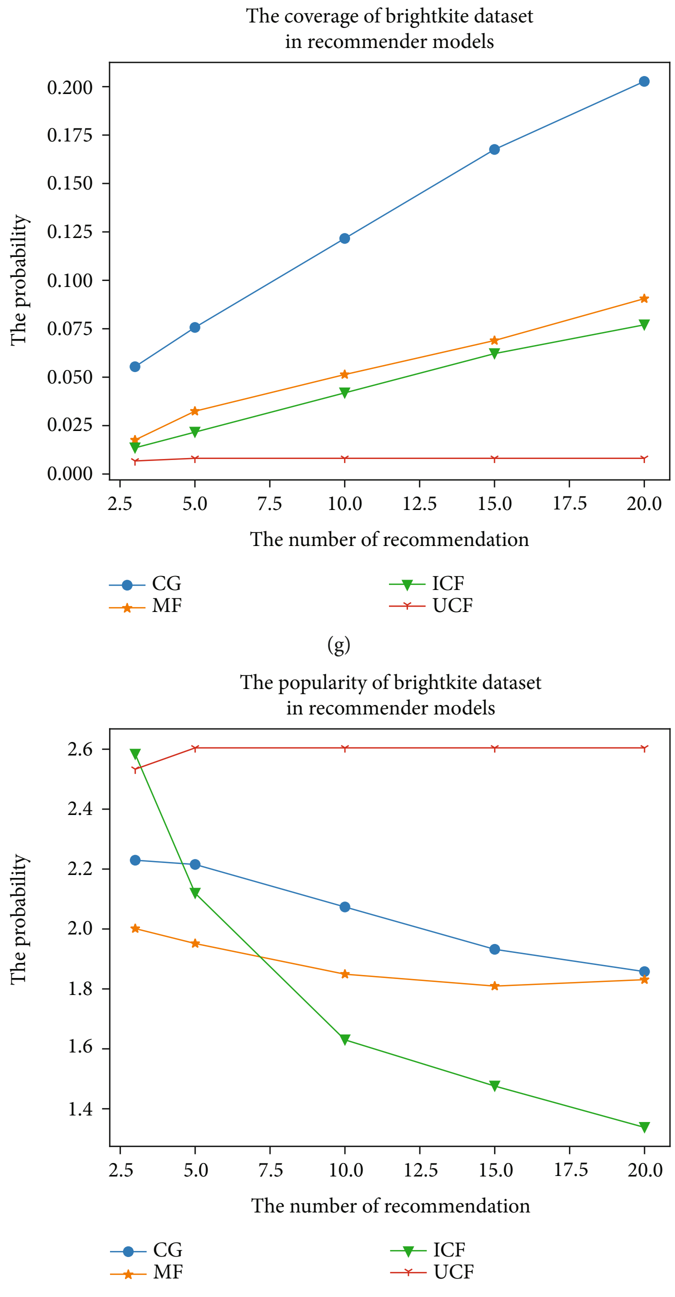


(e)



(f)

Figure 3: Continued.



(g)

(h)

Figure 3: The recommendation result of the Gowalla and Brightkite datasets based on real check-ins. (a–d) Results of the Gowalla dataset. (e–h) Results of the Brightkite dataset.

Accuracy is as follows:

$$\text{Precision} = \frac{\sum_{u\in U} |R(u) \cap T(u)|}{\sum_{u\in U} |R(u)|}. \tag{15}$$

Recall rate is as follows:

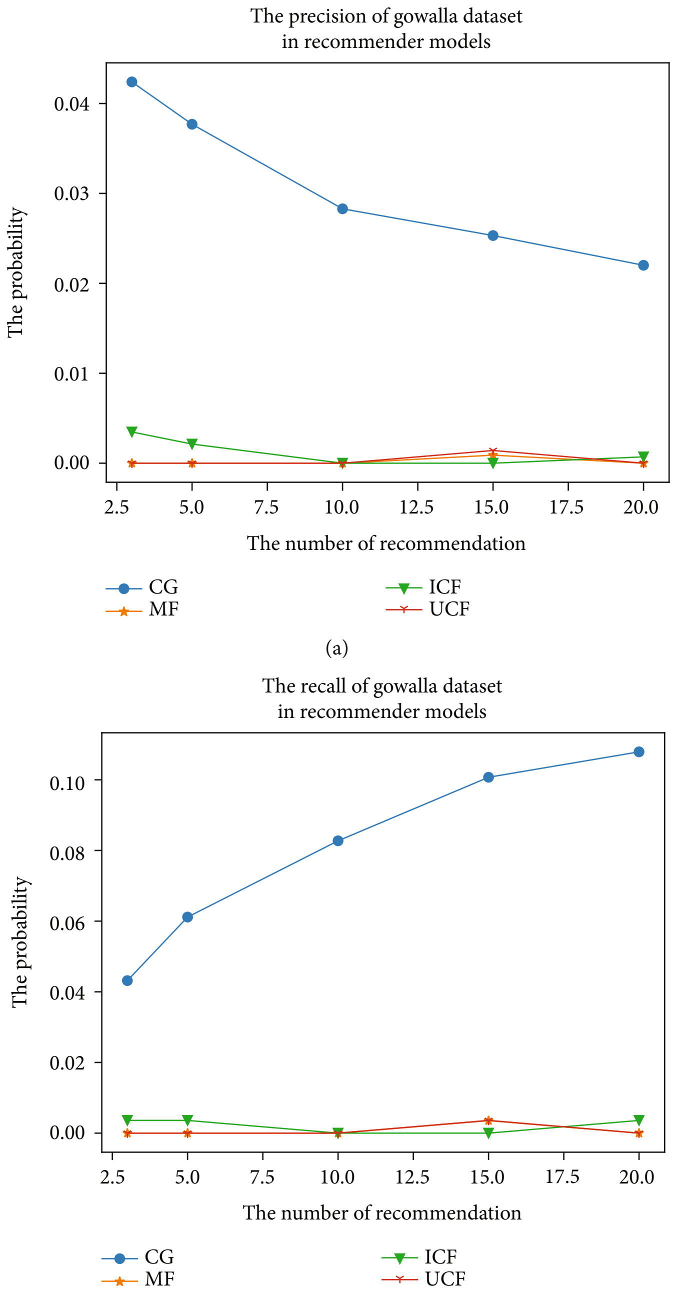
$$\text{Recall} = \frac{\sum_{u\in U} |R(u) \cap T(u)|}{\sum_{u\in U} |T(u)|}. \tag{16}$$

Coverage rate is as follows:

$$\text{Coverage} = \frac{U_{u\in U} R(u)}{LOC}. \tag{17}$$

Popularity is as follows:

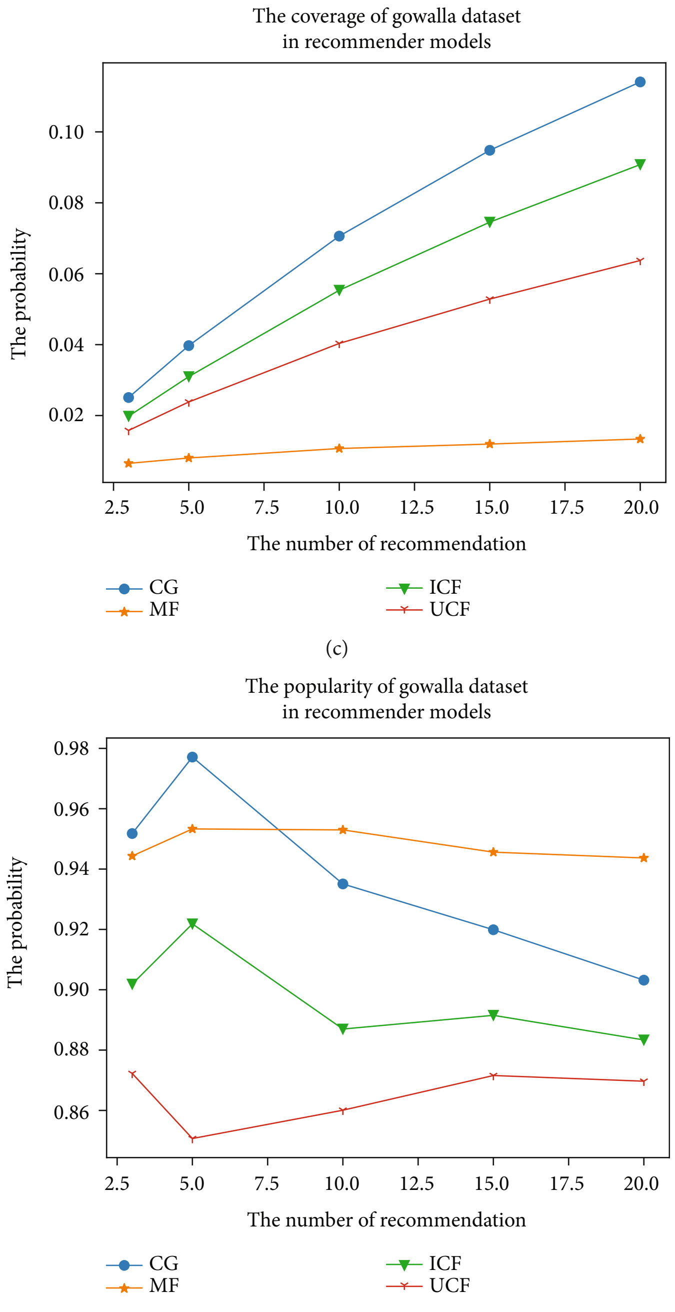
$$\text{Popularity} = \frac{1}{U} \sum_{l_j \in R(u)} \frac{d_{l_j}}{|R(u)|}. \tag{18}$$



(a)

(b)

Figure 4: Continued.



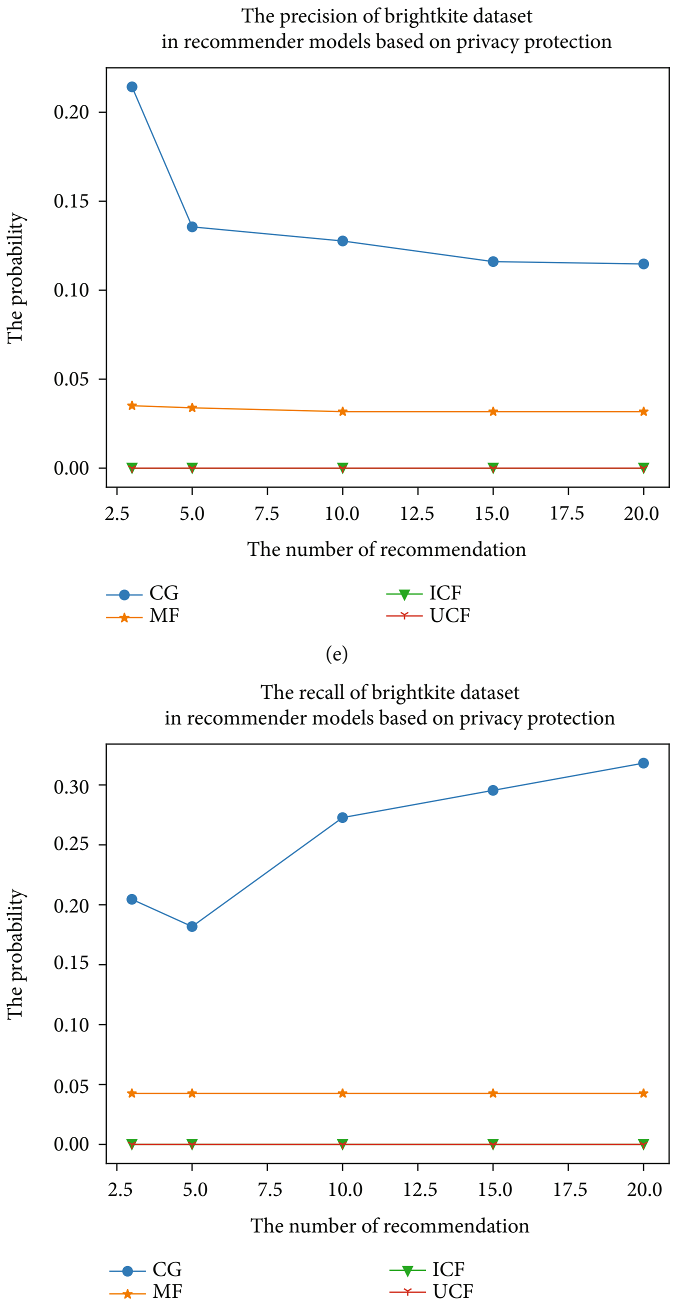
The coverage of gowalla dataset in recommender models

(c)

The popularity of gowalla dataset in recommender models

(d)

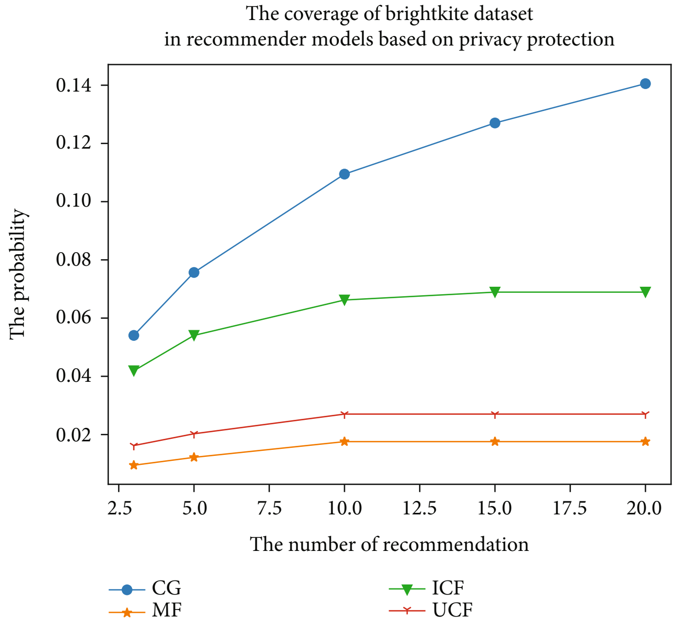
Figure 4: Continued.



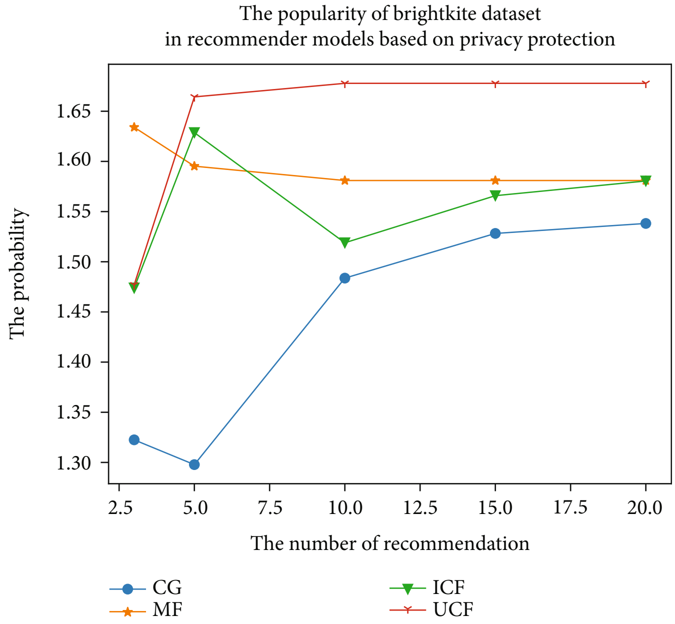
(e)

(f)

Figure 4: Continued.



(g)



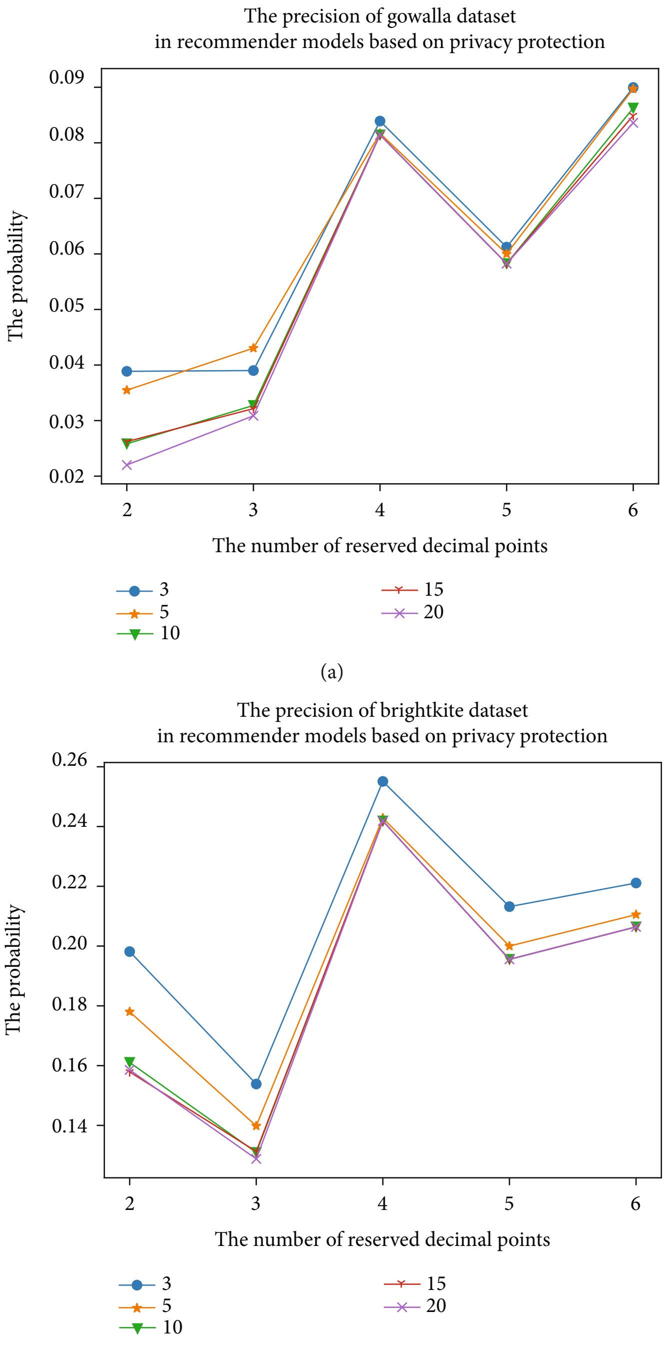
(h)

Figure 4: The recommendation result of the Gowalla and Brightkite datasets based on privacy protection. (a–d) Results of the Gowalla dataset. (e–h) Results of the Brightkite dataset.

$R(u)$ is a list of check-ins of length n recommended by the recommendation model to each user. $T(u)$ is the actual list of users. U is the collection of users, and L is the set of check-in. d_{l_j} is how many people have visited l_j, which represents the popularity of the place.

In addition to evaluating the recommended results, we also evaluate the impact of different extension ranges on recommendations.

6.3. Experimental Setup. In this section, the specific experimental settings and some operations in the experiment are addressed. In the preprocessing stage, we deleted the friend relationship and time information in the original data. The training set and test set are divided according to the number of users, and the partition ratio is 7 : 3. Then, we retain the most recent three check-in records of users to compare with the predicted results. We choose three benchmark algorithms: item-based collaborative filtering, user-based collaborative filtering, and MF as the baseline algorithms. The reasons for choosing these three are as follows. (1) These three algorithms are recommended classical algorithms; (2) the location-based recommendation in this paper is not path



(a)

(b)

Figure 5: Privacy protection results in different degrees. (a) Result of the Gowalla dataset. (b) Result of the Brightkite dataset.

prediction, so many algorithms are not applicable; and (3) these three comparison algorithms have good results in various indicators, and the prediction results are stable and widely applicable in the recommendation algorithm. The number of neighbors in CF is set to 3. The recommended number of LA_{ui} is 3. The recommended number of LB_{ui} is set to $\{3, 5, 10, 15, 20\}$. r_m is the number of decimal points retained in the experiment. We round r_m to the nearest hundredth for latitude and longitude in recommendation experiment. While in the experiment on the impact of the extended range on the recommended results, the number of decimal points retained for latitude and longitude is $r_m = \{2, 3, 4, 5, 6\}$.

6.4. Experimental Results. We perform two types of experiments. The first one is the recommendation experiment. The experiment sets up two specific scenarios in which the evaluation results were obtained. One scenario is preprocessing recommendations without privacy protection. Another scenario is a recommendation experiment to perturb the data during processing. The second experiment is to explore the impact of the intensity of privacy protection on recommendations. The results are given in the following two sections.

6.4.1. Recommendation Results. Figures 3 and 4 both contain eight subfigures. Figures 3(a)–3(d) and 4(a)–4(d) are the results of the Gowalla dataset. Figures 3(e)–3(h) and 4(e)–4(h) are the results of the Brightkite dataset. We evaluated our proposed method and three baselines from four aspects of accuracy, recall rate, coverage rate, and popularity. Figure 3 is the recommendation result without noise, and Figure 4 is the recommendation result with noise. We can see from these figures that, except for popularity, the results of CGPP-POI algorithm are all superior to the other three evaluation indexes. The reason that CGPP-POI's result is less popular than other algorithms is that CGPP-POI divides users into communities, where recommendation results are obtained from. Some communities have a low level of popularity, which can lead to a decrease in popularity. Other algorithms do not have the concept of community, so it is a global selection of check-in recommendation and can derive a higher popularity. We can see the accuracy from Figures 3 and 4. The accuracy after adding noise is affected and is much lower than without noise. That is because we choose to round to the nearest hundredth. That is, the larger extended range and more perturbation will lead to the lower accuracy. Accuracy also decreases as the number of recommendations increases. This is because we are setting the number of comparison check-ins to be reserved at 3. So, as the number of recommendations increases, the accuracy decreases. As can be seen from the recall rate, users are more likely to visit places they have been before.

6.4.2. The Impact of r_m on Recommendation. Figure 5 shows the changes of recommendation accuracy under different privacy protection levels. Figure 5(a) shows the predicted results of the Gowalla dataset, and Figure 5(b) shows the predicted results of the Brightkite dataset. $r_m = \{2,3,4,5,6\}$ is the number of digits reserved after the decimal point. The smaller the number, the wider it extends in the coordinates. The more information it contains, the more perturbations it has. As can be seen from the results of the two figures in Figure 5, the bigger extension range results in a lower the accuracy. The prediction accuracy of $r_m = 3$ is lower than when $r_m = 2$, because the community division does not work well at $r_m = 3$. It can be seen from the two figures that when the number of recommendation is 3, the accuracy is higher than other numbers. This illustrates that (1) both the number of recommendation and the reserved decimal points will impact the result and (2) the higher the accuracy, the closer the recommended quantity is to the reserved comparison quantity. Therefore, it is not necessary to recommend a lot of information to the user, which will increase the time cost of the user.

7. Conclusions

In this paper, we proposed CGPP-POI which is a recommendation model with privacy protection mechanism. Through this model, users can set their own privacy protection scope and then upload data to the third party after LDP. The location information of the user can be protected through LDP processing, while the data remains acceptable utility. The data uploaded after perturbation can satisfy the LDP in the statistical query and maintain its statistical results. But when we try to use this data to build recommendation models, the results are terrible because every user's preferences is different. Therefore, we generalize the disturbed data and build the community by taking advantage of the overlap between users after location expansion. We enhance the coupling between users with high similarity and search for the relationship between users and check-in within the community. Finally, the obtained recommendation results satisfy local differential privacy. We compare the baseline recommendation algorithms on two real data and found that it was superior to the baseline recommendation algorithm in terms of accuracy, recall rate, coverage, and popularity. We also explore the impact of different privacy intensities on recommendations. As can be seen from the experimental results, the bigger the extension range, the lower the accuracy. Our future goal is to improve the recommendation model of POI based on privacy protection by extending model parameters. We aim to improve the accuracy of prediction and protect user location information simultaneously.

Data Availability

The data used to support the research is generally unavailable due to public releasability constraints. Please contact the corresponding author for special release consideration.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Our research is funded by the Fundamental Research Funds for the Central Universities (No. 3072021CF0609).

References

[1] Q. Hou, M. Han, and Z. Cai, "Survey on data analysis in social media: a practical application aspect," *Big Data Mining and Analytics*, vol. 3, no. 4, pp. 27–47, 2020.

[2] M. Xue, P. Kalnis, and H. K. Pung, "Location diversity: enhanced privacy protection in location based services," in *International Symposium on Location-and Context-Awareness*, pp. 70–87, Berlin, Heidelberg, 2009.

[3] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[4] https://en.wikipedia.org/wiki/aol_search_data_leak.

[5] https://www.wired.com/2009/12/netflix-privacy-lawsuit/.

[6] Q. Luo, X. Cheng, J. Yu, Z. Cai, D. Yu, and L. Zhang, "Fast skyline community search in multi-valued networks," *Big Data Mining and Analytics*, vol. 3, no. 3, p. 171, 2020.

[7] R. M. Bell and Y. Koren, “Lessons from the netflix prize challenge,” *Acm Sigkdd Explorations Newsletter*, vol. 9, no. 2, pp. 75–79, 2007.

[8] Y. Liu, T.-A. N. Pham, G. Cong, and Q. Yuan, “An experimental evaluation of pointof-interest recommendation in location-based social networks,” *Proceedings of the VLDB Endowment*, vol. 10, no. 10, pp. 1010–1021, 2017.

[9] P. Zhao, A. Luo, Y. Liu et al., “Where to go next: a spatio-temporal gated network for next poi recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, pp. 5877–5884, 2019.

[10] G. Li, G. Yin, J. Yang, and F. Chen, “Sdrm-ldp: a recommendation model based on local differential privacy,” *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6640667, 15 pages, 2021.

[11] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[12] Y. Koren, “Factorization meets the neighborhood: a multifaceted collaborative filtering model,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, Las Vegas, Nevada, USA, 2008.

[13] A. Mnih and R. R. Salakhutdinov, “Probabilistic matrix factorization,” *Advances in neural information processing systems*, vol. 20, pp. 1257–1264, 2007.

[14] C. Huang, Y. Cheng, W. Zhou, and J. Jia, “Web3d learning framework for 3d shape retrieval based on hybrid convolutional neural networks,” *Tsinghua Science and Technology*, vol. 25, no. 1, p. 93, 2020.

[15] H. Yin, X. Zhou, B. Cui, H. Wang, K. Zheng, and Q. V. H. Nguyen, “Adapting to user interest drift for poi recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2566–2581, 2016.

[16] Y. Wang, G. Yin, Z. Cai, Y. Dong, and H. Dong, “A trust-based probabilistic recommendation model for social networks,” *Journal of Network and Computer Applications*, vol. 55, pp. 59–67, 2015.

[17] G. Li, Z. Cai, G. Yin, Z. He, and M. Siddula, “Differentially private recommendation system based on community detection in social network applications,” *Security and Communication Networks*, vol. 2018, Article ID 3530123, 18 pages, 2018.

[18] H. Kido, Y. Yanagisawa, and T. Satoh, “Protection of location privacy using dummies for location-based services,” in *21st International Conference on Data Engineering Workshops (ICDEW’05)*, pp. 1248–1248, Tokyo, Japan, 2005.

[19] P. Shankar, V. Ganapathy, and L. Iftode, “Privately querying location-based services with sybilquery,” in *Proceedings of the 11th international conference on Ubiquitous computing*, pp. 31–40, Orlando, Florida, USA, 2009.

[20] B. Bamba, L. Liu, P. Pesti, and T. Wang, “Supporting anonymous location queries in mobile environments with privacy-grid,” in *Proceedings of the 17th International Conference on World Wide Web*, pp. 237–246, Beijing, China, 2008.

[21] J. Reagle and L. F. Cranor, “The platform for privacy preferences,” *Communications of the ACM*, vol. 42, no. 2, pp. 48–55, 1999.

[22] S. Papadopoulos, S. Bakiras, and D. Papadias, “Nearest neighbor search with strong location privacy,” *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 619–629, 2010.

[23] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, “Private queries in location based services: anonymizers are not necessary,” in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 121–132, Vancouver, Canada, 2008.

[24] Z. Cai, Z. He, X. Guan, and Y. Li, “Collective data-sanitization for preventing sensitive information inference attacks in social networks,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2016.

[25] I. Memon, Q. A. Arain, M. H. Memon, F. A. Mangi, and R. Akhtar, “Search me if you can: multiple mix zones with location privacy protection for mapping services,” *International Journal of Communication Systems*, vol. 30, no. 16, article e3312, 2017.

[26] T. Hashem and L. Kulik, “"Don’t trust anyone": privacy protection for location-based services,” *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 44–59, 2011.

[27] Z. Chen, X. Hu, J. Xiaoen, and K. G. Shin, “LISA: location information scrambler for privacy protection on smartphones,” in *2013 IEEE Conference on Communications and Network Security (CNS)*, pp. 296–304, National Harbor, MD, USA, 2013.

[28] Z. Cai, Z. Duan, and W. Li, “Exploiting multi-dimensional task diversity in distributed auctions for mobile crowdsensing,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 8, pp. 2576–2591, 2021.

[29] Z. Cai and Z. He, “Trading private range counting over big iot data,” in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019.

[30] Z. Xu and Z. Cai, “Privacy-preserved data sharing towards multiple parties in industrial iots,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[31] Z. Cai and Z. Xu, “A private and efficient mechanism for data uploading in smart cyber-physical systems,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2018.

[32] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, “Generative adversarial networks,” *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.

[33] C. Dwork, “Differential privacy: a survey of results,” in *International Conference on Theory and Applications of Models of Computation: Theory and Applications of Models of Computation*, Springer, 2008.

[34] X. Zhao, Y. Li, Y. Yuan, X. Bi, and G. Wang, “Ldpart: effective location-record data publication via local differential privacy,” *IEEE Access*, vol. 7, pp. 31435–31445, 2019.

[35] Y. S. R. Xin, J. Zhang, and Y. Shao, “Complex network classification with convolutional neural network,” *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 447–457, 2020.

[36] E. J. Newman Mark, “Networks: an introduction,” *Astronomische Nachrichten*, vol. 327, no. 8, pp. 741–743, 2010.

[37] M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” *Physical Review E*, vol. 69, no. 6, article 066133, 2004.

[38] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Theory of cryptography conference*, Springer, 2006.

[39] F. McSherry and K. Talwar, “Mechanism design via differential privacy,” in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS’07)*, pp. 94–103, Providence, RI, USA, 2007.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 7557361, 8 pages
https://doi.org/10.1155/2021/7557361

Research Article

SAN-GAL: Spatial Attention Network Guided by Attribute Label for Person Re-identification

Shaoqi Hou,[1] **Chunhui Liu**,[1] **Kangning Yin**,[2] **Yiyin Ding**,[2] **Zhiguo Wang**,[2] **and Guangqiang Yin**[2]

[1]*School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China*
[2]*School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China*

Correspondence should be addressed to Guangqiang Yin; yingq@uestc.edu.cn

Received 21 May 2021; Accepted 13 August 2021; Published 30 August 2021

Academic Editor: Yan Huang

Person Re-identification (Re-ID) is aimed at solving the matching problem of the same pedestrian at a different time and in different places. Due to the cross-device condition, the appearance of different pedestrians may have a high degree of similarity; at this time, using the global features of pedestrians to match often cannot achieve good results. In order to solve these problems, we designed a Spatial Attention Network Guided by Attribute Label (SAN-GAL), which is a dual-trace network containing both attribute classification and Re-ID. Different from the previous approach of simply adding a branch of attribute binary classification network, our SAN-GAL is mainly divided into two connecting steps. First, with attribute labels as guidance, we generate Attribute Attention Heat map (AAH) through Grad-CAM algorithm to accurately locate fine-grained attribute areas of pedestrians. Then, the Attribute Spatial Attention Module (ASAM) is constructed according to the AHH which is taken as the prior knowledge and introduced into the Re-ID network to assist in the discrimination of the Re-ID task. In particular, our SAN-GAL network can integrate the local attribute information and global ID information of pedestrians without introducing additional attribute region annotation, which has good flexibility and adaptability. The test results on Market1501 and DukeMTMC-reID show that our SAN-GAL can achieve good results and can achieve 85.8% Rank-1 accuracy on DukeMTMC-reID dataset, which is obviously competitive compared with most Re-ID algorithms.

1. Introduction

The biggest feature of smart city is to make full use of the new generation information technology of all walks of life in the city, so as to improve the efficiency of urban management and the quality of citizens' life. As the representative of the new generation of information technology, Internet of Things technology [1–3] and artificial intelligence technology have been more and more widely used.

As a hot field in artificial intelligence (more specifically, in the field of computer vision.), person Re-ID makes up for the deficiency of face recognition technology in cross-camera surveillance images and has a wide application prospect in intelligent video surveillance fields such as airports and supermarkets. However, due to the differences between different cameras and the characteristics of both rigid and flexible pedestrians, its appearance is easily affected by clothing, scale, occlusion, posture, and perspective, which makes person Re-ID become a hot topic with both research value and challenges in the field of computer vision.

In order to solve the above problems, scholars at home and abroad have made many explorations over these years. The traditional Re-ID algorithm relies on some manual features such as color and texture and measures the correlation by calculating the feature distance [4–6]. Due to the complexity of calculation and poor representational ability, these algorithms based on manual features are gradually phased out. With the development of convolutional neural network (CNN), since 2014, scholars began to use deep learning models to solve the problem of person Re-ID [7, 8].

At present, person Re-ID algorithms based on deep learning are mainly divided into two categories: metric learning and representation learning. Metric learning restricts feature space by designing a distance measurement function, so that intraclass spacing of pedestrian features is decreased and interclass features are increased. Classical methods such as triplet loss [9], quadruple loss [10], and group consistent similarity learning [11], the key of such methods lies in sample selection, especially the mining of difficult samples.

Different from metric learning, representational learning takes person Re-ID as a classification task and focuses on designing robust and reliable pedestrian feature representation. At present, scholars generally adopt the method of obtaining global features to solve the Re-ID problem; that is, only the pedestrian ID label is used, and the loss function constraint is adopted to make the network automatically learn the features that are more discriminative for different pedestrian IDs from the entire pedestrian images [12]. In order to enhance the adaptability of the model under the scenes of scale, occlusion, and blur, some scholars [13–15] introduced the attention mechanism into the Re-ID task, so as to improve the models' attention to the salient information in the global features of pedestrians, while suppressing irrelevant noises. However, since different pedestrians may have a similar appearance and the same pedestrian varies greatly in different environments, they cannot be correctly matched from the perspective of global appearance alone. Studies show that [16], as a kind of prior knowledge, the attributes of pedestrians (such as gender, whether they wear hats, whether they carry backpacks, etc.) contain rich semantic information and can provide key discriminant information for Re-ID. However, the relevant datasets are not easy to collect because of involving privacy issues [17, 18]. In addition to the pedestrian attribute labels marked by Lin et al. [16] on DukeMTMC-reID [19] and Market1501 [20] on the person Re-ID datasets, the current datasets do not mark the related areas of pedestrian attributes.

In order to solve these problems, we proposed a SAN-GAL network, which combines pedestrian attribute labels and attention mechanism, and can introduce fine-grained attribute features into the Re-ID network for auxiliary discrimination without additional attribute region labeling. The main contributions are summarized as follows:

(1) *Locate the Attribute Area*. in the pedestrian attribute classification network, the attribute labels are used to guide, and the Grad-CAM algorithm [21] is combined to generate AAH

(2) *Obtainment of Attribute Spatial Attention*. in the person Re-ID network, feature maps of different locations and sizes are selected and combined with the corresponding size of attention heat maps generated by the attribute classification network; ASAM is constructed to assist the discrimination of Re-ID task

(3) *Design Dual-Trace Network*. the pedestrian attribute classification network and Re-ID network are trained jointly to achieve the purpose of information interaction and mutual optimization

2. Related Works

2.1. Attribute Recognition for Re-ID. Person Re-ID based on attribute classification can accurately and quickly mark the target pedestrians in the pedestrian database according to the predicted attribute labels. In 2017, Lin et al. [16] proposed an Attribute Person Recognition (APR) joint recognition network in order to improve the overall accuracy of person Re-ID network. This network included an identity recognition convolutional neural network and an attribute classification model, which can predict attributes through identity recognition and at the same time integrate attribute learning to improve the Re-ID network. In particular, Lin et al. also marked pedestrian attribute labels on DukeMTMC-reID and Market1501, which helped domestic and foreign scholars to improve Re-ID performance by pedestrian attributes.

2.2. Attention Mechanism for Re-ID. The essence of the attention mechanism is to imitate the human visual signal processing mechanism, in order to selectively observe the area of interest, while ignoring other noise information. Inspired by this, in the field of image scene, Liu et al. [22] proposed a classic network model HPNet (HydraPlus-Net) with advantages in fine-grained feature recognition based on attention neural network in 2017. It is mainly aimed at enhancing recognition by the feedback of multilayer attention to different layers in multiple directions. In the field of video scene, Li et al. [15] innovatively used multiple spatial attention module and diversified regular terms to ensure that each spatial attention module learned different parts of the body. Based on that, image features in the sequence were fused through the temporal attention module, and problems such as pedestrian occlusion and misalignment in the video sequence were well solved.

3. Method

Firstly, we introduced the overall architecture and logical relationship of the proposed SAN-GAL network in Section 3.1; then, we introduced the generative process of AAH in Section 3.2; finally, we described the construction method of ASAM in Section 3.3.

3.1. Spatial Attention Network Guided by Attribute Label (SAN-GAL). In order to introduce the local features of pedestrian salience into the Re-ID task without adding additional regional annotation, we design SAN-GAL, as shown in Figure 1. As a dual-trace network, SAN-GAL consists of two branches: pedestrian attribute classification network and Re-ID network. Both branches are based on the pretrained ResNet50 [23], in which the attribute classification task provides attribute prior information to assist the discrimination of the Re-ID task.

The general process is as follows:

Firstly, the attribute classification network extracts features in the gradient forward propagation process, and the extracted features are aggregated (BN-FC-Softmax) to connect attribute classification losses.

Then, the activation map output from Softmax layer (in the attribute pretraining module) calculates the AAH on the activation map of different positions and sizes in the

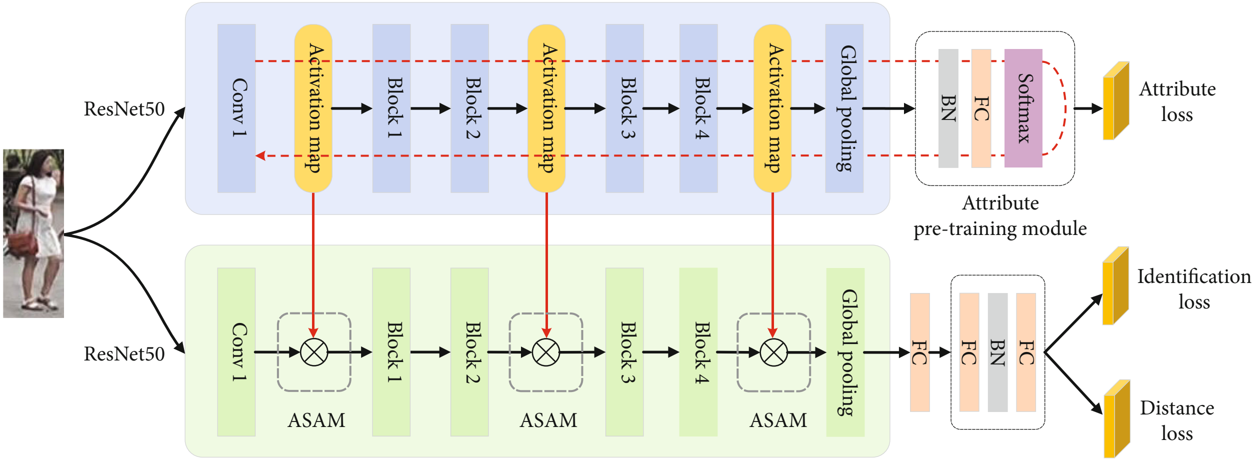


Figure 1: SAN-GAL overall structure diagram.

backbone network through the Grad-CAM algorithm, so as to locate the key area of the attribute.

Finally, in order to enhance the ability of the Re-ID network to extract salient attribute information, the generated AAH is combined with the activation map of the corresponding position in the Re-ID network to construct the ASAM. At the same time, the Re-ID network is optimized after further training.

In particular, in the actual training of SAN-GAL, we have added some special skills:

(1) In the attribute classification network, attribute pre-training module is added only in the first 10 epochs of training and removed after 10 epochs

(2) The two network branches have the same backbone structure but do not share parameters. Since different levels of the network have great differences in the amount of spatial information and semantic information, we introduce the attention mechanism on all three residual blocks with feature map scale changes to enhance the feature processing ability of the model at different fine granularity

(3) In order to achieve the training of the dual-trace network, the attribute classification network needs to be backpropagated twice. After the first calculation of Grad-CAM, the gradient in the activation maps is discarded, and the optimization is achieved after the second update of network parameters. In the actual experiment, in order to improve the computational efficiency, the attribute classification network will complete the pretraining in advance, and the attention parameters calculated offline (from AAH) will be used in the Re-ID network training. In addition to simple implementation, off-line training can also avoid the impact of meaningless AAH at the initial stage of training on the convergence of the Re-ID network, so that the AAH obtained from the same sample calculation is stable and reproducible

3.2. *Attribute Attention Heat Map (AAH).*

Attribute classification network relies on Grad-CAM algorithm to provide spatial focus area for Re-ID. A major advantage of Grad-CAM is that it does not need to transform the model and add additional data annotation, so it is suitable for attention generation algorithm in Re-ID task.

Before calculating Grad-CAM, the output probability y^k predicted by an attribute on Softmax layer in the attribute classification network should be first calculated, and then, the partial derivatives of all pixels on three feature maps of different sizes (i.e., activation maps) on the trunk should be calculated, which can be represented as

$$\frac{\partial y^k}{\partial A_{ij}^c}, \tag{1}$$

where k represents an attribute, (i, j) represents the element coordinates on the current feature map, and c represents the channel of the current feature map.

This result can measure the relevance of some parts of the current feature map to the attribute classification results. Next, the above results are weighted and summed as the coefficients of each channel in the current feature map after global average pooling. After activated by ReLU function, the AAH on the classification of an attribute is obtained, which can be represented as

$$L_{\text{Grad-CAM}}^k = \text{ReLU}\left(\sum_c \left[A^c \times \frac{1}{hw}\left(\sum_{j=1}^{w}\sum_{i=1}^{h}\frac{\partial y^k}{\partial A_{ij}^c}\right)\right]\right), \tag{2}$$

where $L_{\text{Grad-CAM}}^k$ represents the two-dimensional regional heat map generated by the classification attribute k and w and h represent the width and height of the current feature map, respectively. Different pedestrian attributes also have different areas of concern in the current feature map. This concept is shown in Figure 2.

3.3. *Attribute Spatial Attention Module (ASAM).*

The ASAM introduces the attribute prior knowledge into the Re-ID network to assist the discrimination. In order to reduce network complexity and computation consumption, ASAM keeps the structure and size of the activation maps in the Re-ID

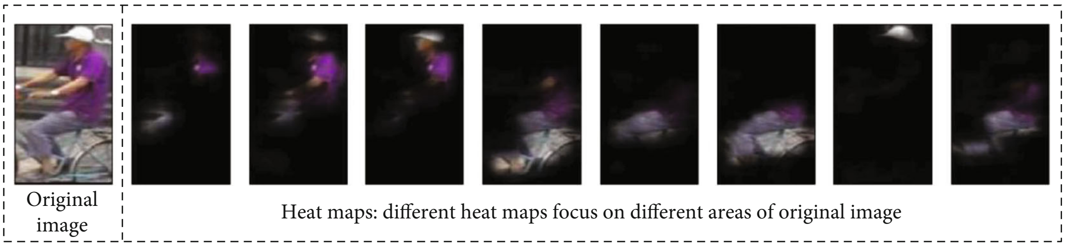

Original image
Heat maps: different heat maps focus on different areas of original image

Figure 2: Schematic diagram of heat map generated by Grad-CAM.

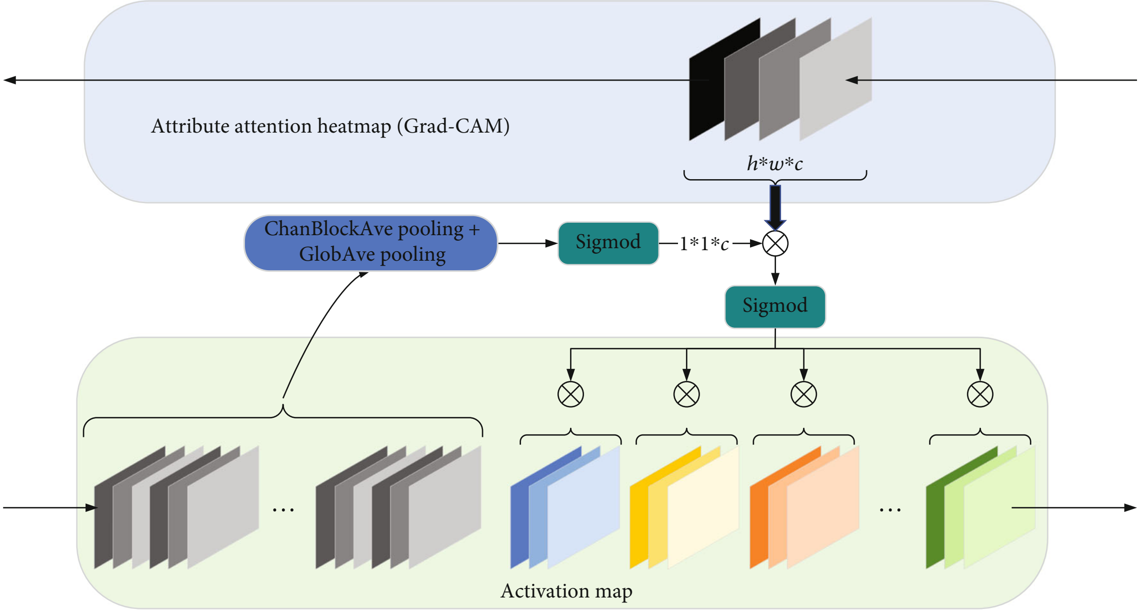

Attribute attention heatmap (Grad-CAM)
h*w*c
ChanBlockAve pooling + GlobAve pooling
Sigmod
1*1*c
Sigmod
...
...
Activation map

Figure 3: ASAM structure diagram.

network unchanged, as shown in Figure 3. In particular, in each ASAM, the first half of the channel features of the activation map that keeps the corresponding location of the Re-ID network is kept unchanged in order to maintain a certain amount of global information and avoid information loss that may be caused by the attention mechanism. In addition, the first half of the channel features in the activation map was used to learn channel attention, because the contributions of different attributes to the Re-ID task are generally different.

Firstly, the attention parameter β_c of each channel of AHH is calculated, as shown in Equation (3). In order to avoid additional parameters, we use channel block average pooling and global average pooling when we calculate β_c.

$$\beta_c = \mathrm{Sigmoid}\left(\frac{1}{hw}\sum_{k=1}^{c_{\mathrm{attr}}/c_{\mathrm{activ}}}\sum_{j=1}^{w}\sum_{i=1}^{h} x_{ijk}^{c}\right), \quad (3)$$

where (i, j) represents the element coordinate on the current feature map; x_{ijk}^{c} represents the pixel value on this coordinate; w and h, respectively, represent the width and height of the current feature map; $c_{\mathrm{attr}}/c_{\mathrm{activ}}$ represents the number of channels of each block feature map; c_{activ} represents half of the number of channels of the activation map in the corresponding position of the Re-ID network; and c_{attr} represents the number of channels for AAH.

Table 1: Comparison of ablation performance.

Methods	Market1501		DukeMTMC-reID	
	Rank-1 (%)	mAP (%)	Rank-1 (%)	mAP (%)
IDonly (baseline)	92.6	81.3	82.3	72.2
ID+attr	92.4	82.1	82.4	72.4
ID+attrAttention (our SAN-GAL)	94.4	83.9	85.8	74.1

Then, the attention parameters $(1 * 1 * c)$ of all channels are multiplied by each channel of AHH. After activation by Sigmoid function, weighted spatial attention parameter $\alpha_c\,(h * w * c)$ is obtained, which can be represented as

$$\alpha_c = \mathrm{Sigmoid}\left(\beta_c \otimes L_{\mathrm{Grad\text{-}CAM}}^{k}\right). \quad (4)$$

Finally, after the spatial attention parameter passes through the Sigmoid function again, the spatial attention parameter is dotted with the last half channel features of the corresponding location activation maps of the Re-ID

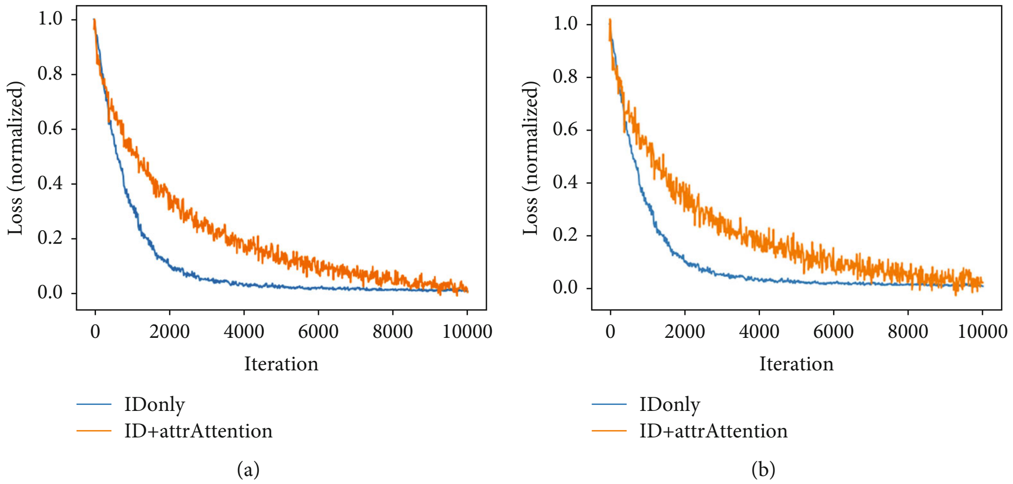


Figure 4: Comparison of training loss of the two network: (a) Market1501; (b) DukeMTMC-reID.

network in blocks to get the final spatial attention (i.e., the feature map contains salient attribute information).

4. Experiments

First of all, we introduced the dataset containing evaluation protocol, loss function, and training details used in Section 4.1; then, we designed a series of ablation experiments in Section 4.2 to verify the effectiveness of our scheme design; finally, in Section 4.3, we compared our model with the advanced Re-ID algorithm to illustrate the superiority of our algorithm.

4.1. The Experimental Details

4.1.1. Dataset

(1) DukeMTMC-reID. DukeMTMC-reID is a person Re-ID subset of DukeMTMC dataset and provides manually annotated bounding boxes. DukeMTMC-reID contains 16,522 training images of 702 of these pedestrian IDs, 2,228 query images from another 702 pedestrian IDs, and 17,661 images' gallery of 702 pedestrian IDs.

(2) Market1501. It was captured on the campus of Tsinghua University in the summer with 6 cameras. The dataset contains a total of 32,688 images with 1,501 pedestrian IDs. Among them, the training set contains 12,936 images of 751 pedestrian IDs, the query set contains 3,368 images of 750 pedestrians, and the test set includes 16,384 images of 750 pedestrians, all of whom have appeared in at least two cameras.

We adopt the Cumulative Matching Characteristics (CMC) at Rank-1 and the mean Average Precision (mAP) as the evaluation indicators to test the performance of different Re-ID methods on these datasets. The mAP is the mean of Average Precision (AP) for each query image. Rank-1 is the probability that the top image in the search results is the target.

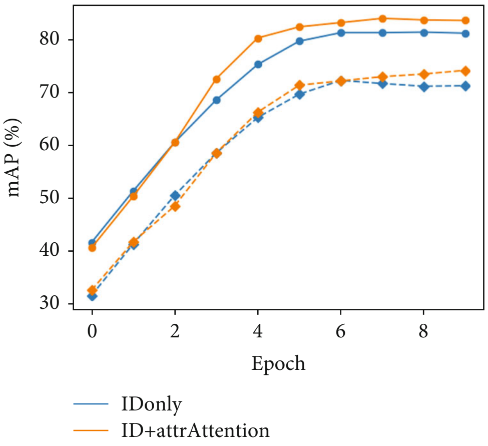


Figure 5: Comparison of mAP change trends of the two networks during training.

4.1.2. Loss Function. We set both attribute loss and identification loss to cross-entropy loss (in our code, it is the combination of Softmax function and cross-entropy function), and distance loss as triplet loss.

4.1.3. Training Details. In order to ensure the consistency of the experimental results, the experimental process is carried out in the same software and hardware environment. The experimental platform is based on 64-bit Ubuntu18.04 operating system, the device memory is 32 G, the CPU is Intel® Xeon E5-2678V3 CPU @2.5 GHz, and the training is conducted on the NVIDIA GTX1080TI single GPU platform, the CUDA version is 10.2, and the experimental framework is based on the PyTorch 1.6.0 version.

We set the size of the input pedestrian images as 256 ∗ 128, and use data augmentation methods such as random erasing and image expansion during the training process. Stochastic Gradient Descent (SGD) is selected by the network parameter optimization algorithm. In particular, we increase the learning rate from 3*e*-5 to 1*e*-3 in the first training epochs and then decrease it to 5*e*-4, 1*e*-4, and 3*e*-5 in the

Table 2: Performance comparison between our SAN-GAL and other classic Re-ID methods.

Methods	Market1501		DukeMTMC-reID	
	Rank-1 (%)	mAP (%)	Rank-1 (%)	mAP (%)
SVDNet (CVPR17) [24]	82.3	62.1	76.7	56.8
PAN (TCSVT18) [25]	82.2	63.4	71.6	51.5
DuATM (CVPR18) [26]	91.4	76.6	81.8	64.6
PCB (ECCV18) [27]	92.3	77.4	81.8	66.1
SPReID (CVPR18) [28]	92.5	81.3	84.4	70.1
VPM (CVPR19) [29]	93.0	80.8	83.6	72.6
AANet (CVPR19) [29]	93.9	83.4	87.7	74.3
IANet (CVPR19) [30]	94.4	83.1	87.1	73.4
SAN-GAL (ours)	94.4	83.9	85.8	74.1
BFE (ICCV19) [31]	95.3	86.2	88.9	75.9

5th, 7th, and 15th training epochs, respectively, which ends after a total of 20 training epochs.

4.2. Ablation Experiments. In order to fully verify the effectiveness of our proposed module and method, we conduct three ablation experiments on Market1501 and DukeMTMC-reID. The specific experimental differences are as follows: firstly, only the pedestrian global ID information is used to train the Re-ID network as the baseline; then, the pedestrian attribute classification network is added on the basis of baseline. Finally, on the basis of the above steps, the ASAM is introduced to form the final design scheme. The experimental results are shown in Table 1.

As shown in Table 1, the Rank-1 and mAP values of IDonly method on the Market1501 dataset are 92.6% and 81.3%, and the Rank-1 and mAP values on the DukeMTMC-reID dataset are 82.3% and 72.2%. After the introduction of attribute information, compared with IDonly method, the mAP of ID+attr method on Market1501 and DukeMTMC-reID increases by 0.8% and 0.2%, but the Rank-1 in Market1501 decreases by 0.2%, and the Rank-1 in DukeMTMC-reID only increases by 0.1%. It can be seen that simply adding attribute classification network cannot bring better effect to the Re-ID task. After introducing our attention mechanism on the basis of ID+attr method, the improvement is significant, especially the accuracy improvement of Rank-1 on the two datasets is 2% and 3.4%, respectively, which fully proves the effectiveness of our ASAM and the overall design.

During the training process, the ordinates of the loss value of IDonly and ID+attrAttention are normalized as shown in Figure 4 (only part of iteration is intercepted). It can be seen that in the Re-ID network with attribute attention mechanism, although loss declines slowly in the initial stage, it is still in a downward trend when IDonly method converges early and eventually achieves loss value lower than that of IDonly method. The change trend comparison of mAP shown in Figure 5, it also supports the enhancement effect of ID+attrAttention method on the Re-ID task.

4.3. Comparison of Algorithms. To demonstrate the superiority of our SAN-GAL in the overall design, we select some representative algorithms in the Re-ID field for comparison, and the selection principles are as follows:

(1) All of them are all based on convolutional neural networks

(2) All of them are representative algorithms in different Re-ID genres

(3) Experiments were carried out on Market1501 and DukeMTMC-reID datasets and evaluated by Rank-1 and mAP

As shown in Table 2, our SAN-GAL algorithm outperforms most of the algorithms on both Market1501 and DukeMTMC-reID datasets. Among all the algorithms listed, compared with PAN in 2018 TCSVT, the Rank-1 and mAP of our SAN-GAL on Market1501 are improved by 12.2% and 20.5%, respectively, and the Rank-1 and mAP of our SAN-GAL on DukeMTMC-reID are improved by 14.2% and 22.6%, respectively. Compared with the VPM in 2019, SAN-GAL is 3.1% higher than its mAP on Market1501. However, compared with BFE, there is still a 0.9% gap in Rank-1 on Market1501. In particular, our SAN-GAL does not add any feature enhancement module or special training technique other than the introduction of a specific attribute attention mechanism. Therefore, our SAN-GAL is an algorithm with both performance and potential.

5. Conclusion

In order to overcome the limitation of global pedestrian features in cross-device scenarios, we proposed SAN-GAL. Different from the previous approach of simply increasing the branch of attribute binary classification network, our SAN-GAL network is guided by attribute labels. Firstly, by generating AAH, we can accurately locate the fine-grained attributes of pedestrians. Then, on the basis of AAH, ASAM is constructed to integrate the global ID information and local attribute information to enhance the discrimination. In particular, our dual-trace network does not need additional attribute region annotation on the dataset, so it has better flexibility and adaptability. By testing on Market1501 and DukeMTMC-reID datasets, the effectiveness and superiority of our scheme design are proved. In the future, we want to expand and apply these ideas to other computer vision tasks, such as Person Search (PS).

Data Availability

Previously reported DukeMTMC-reID and Market1501 data were used to support this study and are available at 10.1109/ICCV.2017.405 and 10.1109/ICCV.2015.133, respectively. These prior studies (and datasets) are cited, respectively, at relevant places within the text as references [19, 20].

Conflicts of Interest

Center for Public Security Information and Equipment Integration Technology agreed to publish this paper, and all authors declare that they have no conflict of interest.

Acknowledgments

This research was supported by the Center for Public Security Information and Equipment Integration Technology of UESTC (University of Electronic Science and Technology of China). Especially, thanks to the computing platform provided by the laboratory.

References

[1] Z. Cai and Z. Xu, “A private and efficient mechanism for data uploading in smart cyber-physical systems,” *IEEE Transactions on Network Science and Engineering (TNSE)*, vol. 7, no. 2, pp. 766–775, 2020.

[2] Z. Xu and Z. Cai, “Privacy-preserved data sharing towards multiple parties in industrial IoTs,” *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 38, no. 5, pp. 968–979, 2020.

[3] Z. Cai and Z. He, “Trading private range counting over big IoT data,” in *The 39th IEEE International Conference on Distributed Computing Systems (ICDCS 2019)*, Dallas, TX, USA, 2019.

[4] W. S. Zheng, S. Gong, and T. Xiang, “Reidentification by relative distance comparison,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 3, pp. 653–668, 2013.

[5] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, San Diego, CA, USA, 2005.

[6] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, Kerkyra, Greece, 1999.

[7] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Deep metric learning for person re-identification,” in *2014 22nd international conference on pattern recognition*, pp. 34–39, Stockholm, Sweden, 2014.

[8] W. Li, R. Zhao, T. Xiao, and X. Wang, “Deepreid: deep filter pairing neural network for person re-identification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 152–159, Columbus, USA, 2014.

[9] A. Hermans, L. Beyer, and B. Leibe, “In defense of the triplet loss for person re-identification,” 2017, https://arxiv.org/abs/1703.07737.

[10] W. Chen, X. Chen, J. Zhang, and K. Huang, “Beyond triplet loss: a deep quadruplet network for person re-identification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 403–412, Honolulu, USA, 2017.

[11] D. Chen, D. Xu, H. Li, N. Sebe, and X. Wang, “Group consistent similarity learning via deep CRF for person re-identification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8649–8658, Salt Lake City, USA, 2018.

[12] M. Geng, Y. Wang, T. Xiang, and Y. Tian, “Deep transfer learning for person re-identification,” 2016, https://arxiv.org/abs/1611.05244.

[13] P. Fang, J. Zhou, S. K. Roy, L. Petersson, and M. Harandi, “Bilinear attention networks for person retrieval,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8030–8039, Seoul, Korea (South), 2019.

[14] Y. Fu, X. Wang, Y. Wei, and T. Huang, “STA: Spatial-temporal attention for large-scale video-based person re-identification,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 8287–8294, Hawaii, USA, 2019.

[15] S. Li, S. Bak, P. Carr, and X. Wang, “Diversity regularized spatiotemporal attention for video-based person re-identification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 369–378, Salt Lake City, USA, 2018.

[16] Y. Lin, L. Zheng, Z. Zheng et al., “Improving person re-identification by attribute and identity learning,” *Pattern Recognition*, vol. 95, pp. 151–161, 2019.

[17] Z. Cai, Z. He, X. Guan, and Y. Li, “Collective data-sanitization for preventing sensitive information inference attacks in social networks,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[18] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, “Generative adversarial networks,” *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.

[19] Z. Zheng, L. Zheng, and Y. Yang, “Unlabeled samples generated by GAN improve the person re-identification baseline in vitro,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3774–3782, Venice, Italy, 2017.

[20] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, “Scalable person re-identification: a benchmark,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1116–1124, Santiago, Chile, 2015.

[21] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, Venice, Italy, 2017.

[22] X. Liu, H. Zhao, M. Tian et al., “Hydraplus-net: attentive deep features for pedestrian analysis,” in *Proceedings of the IEEE international conference on computer vision*, pp. 350–359, Venice, Italy, 2017.

[23] 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, USA, 2016.

[24] Y. Sun, L. Zheng, W. Deng, and S. Wang, “Svdnet for pedestrian retrieval,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3800–3808, Venice, Italy, 2017.

[25] Z. Zheng, L. Zheng, and Y. Yang, "Pedestrian alignment network for large-scale person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 3037–3045, 2018.

[26] J. Si, H. Zhang, C. G. Li et al., "Dual attention matching network for context-aware feature sequence based person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5363–5372, Salt Lake City, USA, 2018.

[27] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, "Beyond part models: person retrieval with refined part pooling (and a strong convolutional baseline)," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 480–496, Munich, Germany, 2018.

[28] M. M. Kalayeh, E. Basaran, M. Gökmen, M. E. Kamasak, and M. Shah, "Human semantic parsing for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1062–1071, Salt Lake City, USA, 2018.

[29] C. P. Tay, S. Roy, and K. H. Yap, "Aanet: attribute attention network for person re-identifications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7134–7143, Long Beach, USA, 2019.

[30] R. Hou, B. Ma, H. Chang, X. Gu, S. Shan, and X. Chen, "Interaction-and-aggregation network for person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9317–9326, 2019.

[31] Z. Dai, M. Chen, X. Gu, S. Zhu, and P. Tan, "Batch dropblock network for person re-identification and beyond," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3691–3701, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8261808, 9 pages
https://doi.org/10.1155/2021/8261808

Research Article

Proactive Flexible Interval Intermittent Jamming for WAVE-Based Vehicular Networks

Hao Li,[1] Xiaoshuang Xing,[2] Anqi Bi,[2] and Jin Qian[3]

[1] *School of Engineering and Applied Science, The George Washington University, Washington, DC 20052, USA*
[2] *Department of Computer Science and Engineering, Changshu Institute of Technology, Changshu, Jiangsu, China*
[3] *College of Computer Science and Technology, Taizhou University, Taizhou, Jiangsu, China*

Correspondence should be addressed to Anqi Bi; anqi_b@cslg.edu.cn

Received 14 April 2021; Accepted 1 August 2021; Published 15 August 2021

Academic Editor: Yan Huang

In this paper, we deal with the eavesdropping issue in Wireless Access in Vehicular Environments- (WAVE-) based vehicular networks. A proactive flexible interval intermittent jamming (FIJ) approach is proposed which predicts the time length T of the physical layer packet to be transmitted by the legitimate user and designs flexible jamming interval (JI) and jamming-free interval (JF) based on the predicted T. Our design prevents eavesdroppers from overhearing the information with low energy cost since the jamming signal is transmitted only within JI. Numerical analysis and simulation study validate the performance of our proactive FIJ, in terms of jamming energy cost and overhearing defense, by comparing with the existing intermittent jamming (IJ) and FIJ.

1. Introduction

A WAVE- (Wireless Access in Vehicular Environments-) based vehicular network has been considered a promising way to improve safety and driving experience with vehicular level information exchange playing the most critical role. However, wireless communication is vulnerable to eavesdropping threats due to its broadcasting nature. The information exchanged among vehicles, including vehicle identities, locations, and speeds, is exposed to eavesdropping attackers. To protect this private information from leakage, reliable eavesdropping defense mechanisms must be designed.

Friendly jamming is an effective approach to defend against eavesdropping [1–5]. Continuous jamming (CJ), which requires the friendly jammer to keep sending jamming signals during the whole transmission of the legitimate transmitter, has been extensively studied in literature. Eavesdroppers are disabled via CJ in the cost of large energy consumption. In recent work [6], the authors argued that it is unnecessary to jam the whole transmission. Partially jamming the transmission of a data packet is capable of preventing eavesdroppers from getting sensitive information. Therefore, they proposed an intermittent jamming (IJ) scheme where the friendly jammer sends the jamming signal only in the jamming interval (JI) and keeps silent in the jamming-free interval (JF). This scheme can keep the information safe with low energy cost. However, the length of JI and JF was fixed in their design (as shown in Figure 1) without considering the length of the packet transmitted by the legitimate transmitter. This fixed design has drawbacks in the following aspects. When the length of the transmitted packet is short, unnecessary energy will be consumed during a long JI. On the other hand, a combination of JI and JF will occur repeatedly for a long packet. The jammer should switch between JI and JF frequently, and energy will be wasted due to the switching loss. Therefore, the length of the transmitted packet should be considered when designing the length of JI and JF to achieve better energy efficiency.

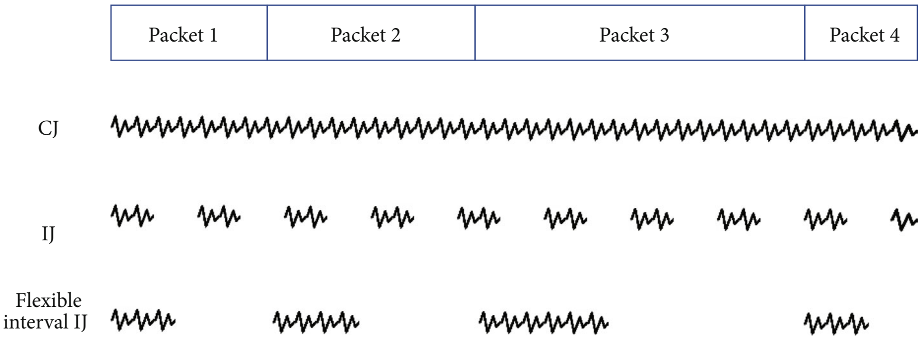


Figure 1: Continuous jamming, intermittent jamming, and flexible interval intermittent jamming.

In this paper, we try to design a flexible interval IJ (FIJ) scheme by setting the length of JI and JF according to the time length of the transmitted packet. For a specific physical packet with time length T, we will find the time duration within which the core information is transmitted and set this duration as JI. As for T, its actual value cannot be obtained before the physical packet is generated. However, if the jammer obtains the value of T after the packet has been generated and decides the length of JI and JF accordingly, nonnegligible time delay will be introduced before starting the jamming process. This way, the jamming signal may not be able to be transmitted synchronously with the physical packet leading to degraded jamming performance. To deal with this problem, this paper will predict the value of T and proactive FIJ will be enabled to achieve better jamming performance. As a summary, the contributions of this paper are as follows.

(i) The physical packet structure in WAVE-based vehicular networks is analyzed. For a specific physical packet with time length T, the time length of the "Application Data," which is generated in the application layer and contains the core information to be transmitted, is obtained

(ii) An FIJ scheme is proposed where the length of JI depends on the value of T such that the friendly jammer disables the eavesdropper with less energy cost

(iii) Support vector regression (SVR) is applied to learn the characteristics of the time length of N historical physical packets and predict the time length of the future physical packet. Proactive FIJ is enabled by designing the length of JI according to the predicted time length of the next physical packet

The paper is organized as follows. Section 2 discusses the related works. The considered system model is illustrated, and the problem is formulated in Section 3. The FIJ scheme is designed in Section 4, and the value of T is predicted based on SVR in Section 5. Performance investigation is conducted in Section 6. Finally, the paper is concluded in Section 7.

2. Related Works

From the application layer to the link layer, the security threat has long been under concern [7–11]. The multimedia streaming scheme proposed in [12] deals with the security issues in the application layer. Authentication schemes are designed to ensure the confidentiality of communication in the transport layer [13–15]. The secured routing protocol proposed in [16, 17] provides a safe transmission in the network layer. [18] detects possible denial of service ahead of confirmation time in the link layer.

According to the IEEE 802.11p standard, driving-related information, including identity, location, speed, and direction, is transmitted through vehicle to vehicle (V2V) communication and vehicle to infrastructure (V2I) communication. This sensitive information is transmitted on the air and is exposed to eavesdropping attack in the physical layer due to the natural characteristics of wireless communication. By eavesdropping this information, a malicious user may track the driving information and analyze the driving route of legitimate users [19]. Therefore, it is necessary to tackle the eavesdropping attack in the physical layer for secure sensitive information transmission.

Friendly jamming has been widely applied to defend against eavesdropping attacks. It can help to improve the security of vehicle localization [20], location verification [21], and secure communication [22]. In most existing friendly jamming schemes, friendly jammers keep sending jamming signals. These schemes are known as CJ which consumes a large amount of energy. In order to reduce power consumption, [23] proposes temporary jamming to provide information security when encryption is limited. A later research [6] advances an IJ scheme where the friendly jammer sends the jamming signal only in the JI and keeps silent in the JF. The IJ scheme greatly decreases the power consumption while providing information security via achieving a high package error rate (PER) at the eavesdropper. However, this scheme fixes the length of JI and JF without considering the length of the packet transmitted by the legitimate transmitter. For a short physical packet, unnecessary energy will be consumed during a long JI. On the other hand, a combination of JI and JF will occur repeatedly for a long packet. Energy will be wasted during the frequent change between JI and JF. In order to further reduce the energy cost of the IJ scheme, this paper will design flexible JI and JF depending on the length T of the transmitted packet.

In order to predict the time length T of the physical packet to be transmitted in the next time, machine learning will be applied. Typical machine learning algorithms include linear regression, logistic regression, ridge regression, and support vector regression [24–29]. Linear regression [24] uses least square methods as cost function and optimizes the target model by Newton iteration. However, linear regression may obtain local optimum solution for some applications. Logistic regression [25] is based on the probabilistic mechanism, which determines parameters by maximum likelihood estimation. However, logistic regression is a linear model in essence and may not be suitable for the vibrating samples. By adding an additional degree of deviation to the regression estimate, ridge regression can effectively reduce the variance [27]. Nevertheless, this model requires samples involved to be multidimensional. In our work, the time length of the historically transmitted physical packets will be taken as the samples. They are one-dimensional vibrating samples. Therefore, neither logistic regression nor ridge regression fits our application. On the other hand, support vector regression (SVR) [28] maps samples into the high-dimensional feature space by nonlinear change. Thus, the performance of SVR is independent of the sample dimension. Besides, SVR shows effective fitting ability for vibrating samples. Therefore, we will utilize the SVR model to learn the characteristics of the time length of N historical physical packets and predict the time length of the physical packet to be transmitted.

3. Problem Formulation

We are under a general vehicle communication scenario in a vehicular network under the WAVE protocol. As shown in Figure 2, the legitimate user U_A is sending its driving information to U_B. Meanwhile, there is an eavesdropper U_E trying to overhear the packets being sent. A cooperative jammer U_J located near U_A is sending jamming signals with power P_J to degrade the packets received by eavesdropper U_E.

For a physical packet with time length T, U_J sends jamming signals in the JI with length T_J and keeps silence in the JF with length T_F. Here, $T_J \leq T$, $T_F \leq T$, and $T_J + T_F = T$. Let W_J indicate the energy cost of the cooperative jammer, B_J indicate the bit error rate (BER) of U_E during JI, B_F indicate the BER of U_E during JF, and B_E indicate U_E's average BER within T. It can be derived that

$$W_J = T_J \cdot P_J, \tag{1}$$

$$B_E = \frac{T_J}{T} \cdot B_J + \frac{T_F}{T} \cdot B_F. \tag{2}$$

The closed-form expressions of the BERs for different modulation schemes have been given in [30]. It can be found that BER is always a decreasing function of the signal to noise plus interference ratio (SNIR), denoted by γ_b. During JF, no jamming signal is transmitted by the jammer. Therefore, $\gamma_b^{\mathrm{JF}} = E_b/N_0$ when calculating B_F. Here, E_b is the received signal energy per bit and N_0 is the power spectral density of the noise. On the other hand, the receiving performance of U_E is degraged by the jammer during JI. Therefore, $\gamma_b^{\mathrm{JI}} = E_b/(N_0 + \phi_J)$ when calculating B_J. Here, $\phi_J = P_J|h_{JE}|^2/B$ is the received jamming signal power spectral density with $|h_{JE}|^2$ indicating the channel gain from U_J to U_E and B being the channel bandwidth. Obviously, $\gamma_b^{JI} \leq \gamma_b^{JF}$ and $B_J \geq B_F$. Therefore, B_E is an increasing function of T_J. According to (1), it can be found that W_J is also an increasing function of T_J. Recall that we want to disable the eavesdropping of U_E with low energy cost; we need to decide a proper T_J that can ensure a high enough BER at U_E while achieving a W_J as low as possible.

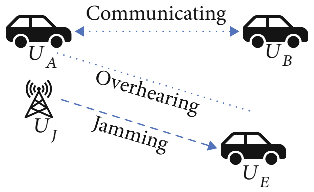


FIGURE 2: General communication scenario.

4. Design of Flexible Interval IJ Scheme

In order to obtain a high enough B_E while maintaining a low W_J, the jammer should transmit jamming signals only during the transmission time of the most significant part of the physical packet. Figure 3 shows the component of a physical packet. Intuitively, the "Application Data," which is generated in the application layer, contains the core information to be transmitted by U_A to U_B. Therefore, "Application Data" is the most significant part of the physical packet. If the jammer can identify the time duration within which the "Application Data" is transmitted and sends jamming signals only during this time, U_E's eavesdropping will be disabled and U_J's energy cost will be reduced. Therefore, the main challenge to be solved in our design is to identify the time duration within which the "Application Data" is transmitted.

According to [31], a physical packet is consisting of a 16 μs PLCP preamble, a 4 μs Signal Field, and a variable-length Data Field. The Data Field is constructed by 16 bits of the PLCP Header, the WSMP-T-Header, the WSMP-N-Header, the LLC Header, the MAC Header, 32-bit FCS, 6-bit tail, and variable-length Application Data. Moreover, n bits pad bits are also added in the Data Field to make the length of the Data Field divisible by N_{DBPS}. Therefore, n takes a value between 0 and $N_{\mathrm{DBPS}} - 1$. The value of N_{DBPS} depends on the modulation schemes and the coding rates. Typical values of N_{DBPS} in WAVE-based vehicular networks are listed in Table 1.

When the Data Field is constructed, it will be divided into symbols. Each symbol consists of N_{DBPS} bits and is 4 μs long in time. According to [6], the minimum length of the WSMP-T-Header, the WSMP-N-Header, the LLC Header, and the MAC Header is 2 bytes, 2 bytes, 2 bytes, and 24 bytes, respectively. There are a total of 30 bytes, which are 240 bits, in the physical packet before the Application Data in the Data Field. In the time domain, the time length of these 240 bits will be $t_1 = 240/N_{\mathrm{DBPS}} \times 4\,\mu$s. As mentioned before, there are 6-bit tail, 32-bit FCS, and 0 to $N_{\mathrm{DBPS}} - 1$ bits pad bits after the Application Data. These are totally 38 to $37 + N_{\mathrm{DBPS}}$ bits,

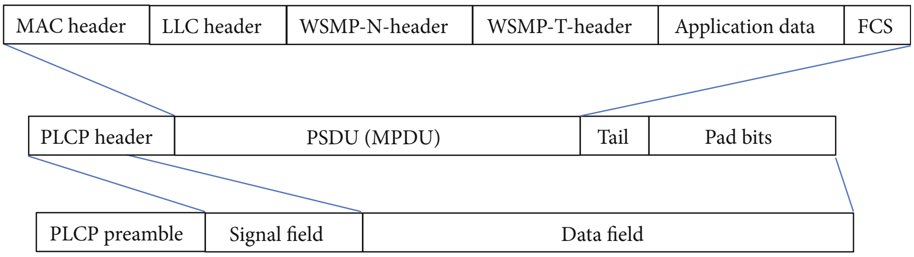


Figure 3: Physical packet structure.

Table 1: Values of N_{DBPS} for different modulation schemes and coding rates.

Modulation	Coding rate	N_{DBPS} (bits)	Modulation	Coding rate	N_{DBPS} (bits)
BPSK	1/2	24	16-QAM	1/2	96
BPSK	3/4	36	16-QAM	3/4	144
QPSK	1/2	48	64-QAM	2/3	192
QPSK	3/4	72	64-QAM	3/4	216

and the time length of these bits is denoted by t_2. t_2 takes value from $38/N_{\text{DBPS}} \times 4\,\mu\text{s}$ to $(37 + N_{\text{DBPS}})/N_{\text{DBPS}} \times 4\,\mu\text{s}$. The PLCP preamble, the Signal Field, and the headers are transmitted before the Application Data. The time length before transmitting the Application Data in the physical packet, which is denoted by T_F^1, can be calculated as $T_F^1 = 16\,\mu\text{s} + 4\,\mu\text{s} + t_1$. On the other hand, the FCS, the tail bits, and the pad bits are transmitted after the Application Data. Therefore, the time length after transmitting the Application Data in the physical packet, which is denoted by T_F^2, can be calculated as $T_F^2 = t_2$. Then, for a physical packet of length T, the flexible interval IJ scheme will be designed as shown in Figure 4. According to the value of N_{DBPS} given in Table 1, the value of T_F^1, T_F^2 can be easily obtained. For example, $T_F^1 = 60\,\mu\text{s}$ and $6.3\,\mu\text{s} \le T_F^2 \le 10.17\,\mu\text{s}$ when the physical packet is BPSK modulated with the coding rate being 1/2. Then, we have $T_J = T - T_F = T - T_F^1 - T_F^2$. Theoretically, the best antieavesdropping performance can be achieved when T_F^2 takes the lower bound value, which is $T_F^2 = 6.3\,\mu\text{s}$ in the aforementioned example, while most energy can be saved when T_F^2 takes the upper bound value, that is, $T_F^2 = 10.17\,\mu\text{s}$ in the example.

5. Predicting the Time Length of the Physical Packet Based on SVR

This section is aimed at obtaining the time length T of the physical packet to be transmitted. As discussed in Section 1, the jamming performance will be degraded if the jammer tries to obtain the value of T after the physical packet has been generated. To solve this problem, we learn the characteristics of the time length of N historical physical packets and predict the time length of the physical packet to be transmitted (that is, the $(N+1)$-th physical packet) via machine learning. Then, proactive FIJ will be enabled by designing the length of JI and JF according to the predicted result, and the jamming signal will be able to be transmitted synchronously with the physical packet to ensure the jamming performance.

Let $\{(x_1, t_1), (x_2, t_2), \cdots, (x_N, t_N)\}$ denote N historical records, called as samples, regarding the time length of the physical packets. Here, $x_i = i$, $1 \le i \le N$, is the index of the physical packet that has been transmitted with the x_N-th physical packet being the most recently transmitted one. t_i is the time length in μs of the x_i-th physical packet. Then, we utilize the SVR model [28] to learn the characteristics of the time length of N historical physical packets by finding the hyperplane that fits the N samples. To simplify the calculation, we first scale the time length values of the samples. Let y_i denote the scaled value of t_i, then

$$y_i = 10 \times \frac{t_i}{t_{\max}}, \tag{3}$$

with $t_{\max} = \max\{t_1, t_2, \cdots, t_N\}$. The scaled value y_i will be distributed within $[0, 10]$. SVR define the function of the fitting hyperplane as

$$f(x) = w^T x + b. \tag{4}$$

Here, $x = (x_1, x_2, \cdots, x_N)^T$, $w = (w_1, w_2, \cdots, w_N)^T$, and $b = (b_1, b_2, \cdots, b_N)^T$. The SVR model is aimed at minimizing the maximum margin between $y = (y_1, y_2, \cdots, y_N)^T$ and $f(x)$. According to [32], the target function of the SVR model can be defined as

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right) \tag{5}$$

$$\begin{aligned} \text{subject to} \quad & y_i - (w_i x_i + b_i) \le \varepsilon + \xi_i, i = 1, 2, \cdots, N \\ & (w_i x_i + b) - y_i \le \varepsilon + \xi_i^*, i = 1, 2, \cdots, N \\ & \xi \ge 0, \xi_i^* \ge 0, i = 1, 2, \cdots, N. \end{aligned} \tag{6}$$

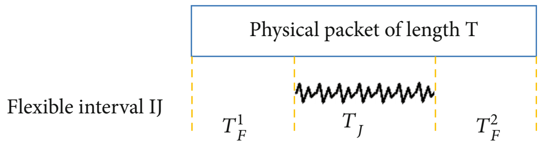


Figure 4: Flexible interval IJ scheme for a physical packet of length T.

Here, $C > 0$ is the constant regularization parameter and ξ and ξ^* are slack variables whose values are close to 0. Slack variables are introduced according to soft margin loss theory to cope with infeasible constraints of the optimization problem. The Lagrangian function of (5) is given in

$$\begin{aligned} L(w, b, \alpha, \xi, \eta) = {} & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right) \\ & + \sum_{i=1}^{N}\alpha_i\left(w^T x_i + b - y_i - \varepsilon_i - \xi\right) \\ & - \sum_{i=1}^{N}\alpha_i^*\left(w^T x_i + b - y_i - \varepsilon_i - \xi^*\right) \\ & - \sum_{i=1}^{N}\left(\eta_i\xi_i + \eta_i^*\xi_i^*\right). \end{aligned} \tag{7}$$

Here, L is the Lagrangian and $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$ are Lagrange multipliers (also referred to as dual variables) that should satisfy positivity constraints, i.e., $\alpha_i \geq 0, \alpha_i^* \geq 0, \eta_i \geq 0$, and $\eta_i^* \geq 0$. In this condition, the target function of the SVR model can be transferred to its dual problem. By optimizing the dual variables in Lagrangian function, the original target function (5) would be solved as well. Specifically, according to the SVR framework and Karush-Kuhn-Tucker (KKT) conditions, we optimize the minimum of the partial derivatives of L with respect to the variables (w, b, ξ_i, ξ_i^*), namely,

$$\frac{\partial L}{\partial w} = 0 \longrightarrow w = \sum_{i=1}^{N}\left(\alpha_i - \alpha_i^*\right)x_i, \tag{8}$$

$$\frac{\partial L}{\partial b} = 0 \longrightarrow \sum_{i=1}^{N}\left(\alpha_i - \alpha_i^*\right) = 0, \tag{9}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \longrightarrow C - \alpha_i - \eta_i = 0, \tag{10}$$

$$\frac{\partial L}{\partial \xi_i^*} = 0 \longrightarrow C - \alpha_i^* - \eta_i^* = 0. \tag{11}$$

Substituting (8), (9), (10), and (11) into (7), the dual optimization problem can be yielded, and the problem converts to minimizing the objective function as follows:

$$\begin{aligned} \min_{\alpha_i, \alpha_i^*} & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\alpha_i - \alpha_i^*\right)\left(\alpha_i - \alpha_i^*\right)\left(x_i^T x_j\right) \\ & + \sum_{i=1}^{N} y_i\left(\alpha_i - \alpha_i^*\right) + \varepsilon\left(\alpha_i + \alpha_i^*\right). \end{aligned} \tag{12}$$

Obviously, (12) is the dual form of the target function and a typical quadratic programming problem. The problem could be easily solved by several mathematic frameworks, such as SMO, and obtained the corresponding α_i, α_i^*. In the involved experiments, we directly apply the toolkit in MATLAB. Then, the hyperplane function (4) becomes

$$f(x) = w^T x + b = \sum_{i=1}^{N}\alpha_i y_i x_i^T x + b. \tag{13}$$

The SVR model usually takes linear function, polynomial function, Radial Basis Function (RBF), or sigmoid function as kernel function. In our work, considering that t_i is one-dimensional and vibrates greatly, smooth kernel function is applicable. Besides, [33] has proved that RBF with proper δ could smoothly fit any curve compared with other kinds of kernel functions. Accordingly, we choose RBF given in (14) as kernel function when training the SVR model [33]:

$$K\left(x_i, x\right) = \exp\left(-\frac{\|x_i - x\|^2}{2\delta^2}\right). \tag{14}$$

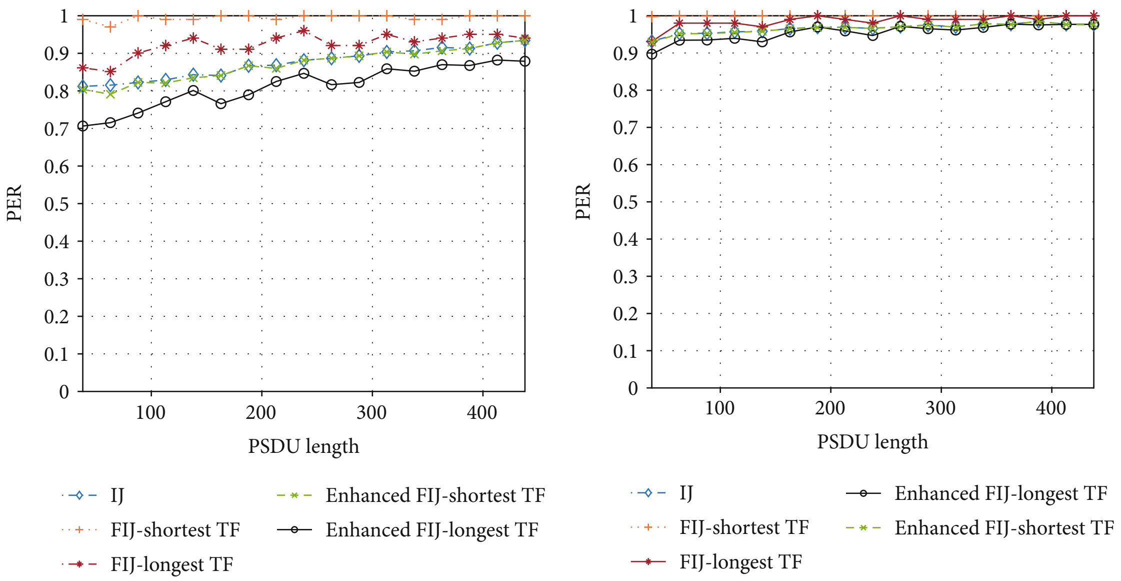
That is, $x_i^T x$ in (13) should be replaced by $K(x_i, x)$ as shown in (14).

After getting the hyperplane function given in (13), we can predict the time length T of the x_{N+1}-th physical packet by substituting $x = N + 1$ into (13) and conducting the reverse conversion of (3). That is, $T = f(N + 1) * t_{\max}/10$.

6. Numerical Analysis and Simulation Study

In this section, we first investigate the performance of FIJ for securing the transmission of physical packets with a known time length. Then, SVR is applied to find the hyperplane that fits the samples of 500 historically transmitted physical packets and predict the time length of the physical packet to be transmitted. Proactive FIJ is conducted based on the prediction result, and the performance is studied.

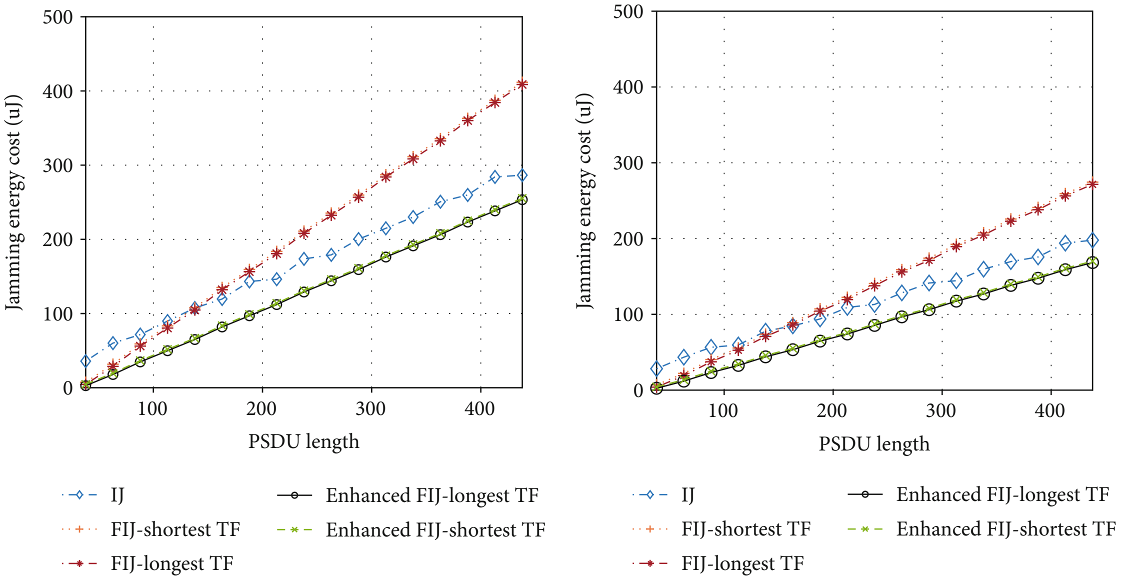
6.1. Performance Investigation of FIJ for Securing the Transmission of Physical Packets with Known T. This subsection compares the performance of the FIJ scheme with the IJ scheme proposed in [6]. Besides, the performance of our design when T_F^2 takes the lower bound value (referred to as FIJ-shortest TF in the following) and the upper bound value (referred to as FIJ-longest TF in the following) is also investigated. The simulation is performed in MATLAB 2018b using the WLAN toolbox. We use function "wlanNonHTConfig" to generate non-HT packets transmitted in the WAVE-based vehicular network. The channel bandwidth is set to be 10 MHz, and we are using the default sampling rate for



(a) BPSK, $r = 1/2$

(b) BPSK, $r = 3/4$

Figure 5: Packet error rate comparison with different PSDU lengths.



(a) BPSK, $r = 1/2$

(b) BPSK, $r = 3/4$

Figure 6: Jamming energy cost comparison with different PSDU lengths.

10 MHz. We set the delay profile as "Urban NLOS" because most of the V2V communication happens in an urban area and does not have a line of sight. BPSK modulation is used, and the coding rate r is set to be 1/2 and 3/4.

The performance comparison is conducted from two aspects. To validate the antieavesdropping performance of our design, the packet error rate (PER) of U_E, which is the ratio of the number of physical packets not successfully decoded by U_E to the number of the physical packets sent by the transmitter U_A, is adopted. The function "V2VPERSimulator" from MATLAB is utilized to simulate the PER. The energy cost for sending jamming signals referred to as the jamming energy cost in the following is used to investigate the energy efficiency of our design.

According to [6], the optimal transmission power of U_J is set to be $P_J = 760$ mW for BPSK modulation with coding rate r being 1/2. The corresponding T_J and T_F are 47.12 μs and 28.88 μs, respectively, in the IJ scheme. While for BPSK modulation with $r = 3/4$, the IJ scheme is set as $P_J = 760$ mW, $T_J = 37.2\,\mu$s, and $T_F = 22.8\,\mu$s. The setting of the IJ scheme is fixed regardless of the length of the transmitted physical packet. On the other hand, the length of T_J and $T_F = T_F^1 + T_F^2$ in our design is flexible which can be calculated as given

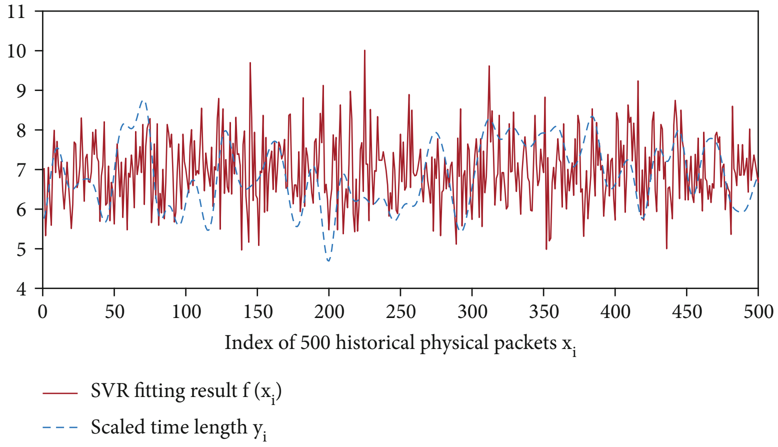


Figure 7: The training result of 500 historical physical packets based on SVR model.

in Section 4. U_J's transmission power in our flexible interval IJ scheme is set to be the same as that in the IJ scheme, which is $P_J = 760$ mW.

We change the length of the PSDU from 38 octets to 438 octets resulting in the time length of the physical packets changing from 76 μs to 608 μs for BPSK modulation with $r = 1/2$ and from 60 μs to 412 μs for BPSK modulation with $r = 3/4$. The PER of U_E is shown in Figure 5. It can be found that U_E's PER increases with the increasing of the PSDU length for schemes other than FIJ-shortest TF. With the increase of the PSDU length, more information bits are enclosed in a physical packet. The probability of information bits within a physical packet being incorrectly decoded will increase resulting in an increased PER. For the FIJ-shortest TF scheme, U_E's SNR keeps low since U_J sends jamming signals during the whole transmission time of the "Application Data." Therefore, U_E's PER is always close to 100% regardless of the PSDU length. Small performance fluctuations occur for the FIJ-longest TF scheme. In the FIJ-longest TF scheme, the length of T_F^2 is fixed to be $(37 + N_{\text{DBPS}})/N_{\text{DBPS}} \times 4\,\mu$s by assuming that there are always $N_{\text{DBPS}} - 1$ pad bits in the physical packet. However, the length of the pad bits varies with the PSDU length leading to insufficient jamming of the "Application Data" for some PSDU length and thus performance fluctuations on U_E's PER. Moreover, one can see that a higher coding rate r causes a higher PER. A higher r implies more information bits, and less redundant bits are enclosed in a physical packet, which means that more information is transmitted in a physical packet and the transmission efficiency is improved. However, the redundant bits play an important role in error correction, and less redundant bits can decrease U_E's error correction capability and lead to a higher PER.

The results regarding the jamming energy cost are given in Figure 6. We found that our FIJ scheme consumes less energy when the physical packet is short (for example, when the PSDU is 100 bytes long). While for long physical packets, the IJ scheme performs better in terms of energy cost. This is because the length of T_J and T_F is fixed in IJ. In other words, T_J/T is fixed for any PSDU length (i.e., any physical packet length). In the flexible interval IJ scheme, the length of $T_F = T_F^1 + T_F^2$ is fixed, while the length of $T_J = T - T_F$ increases with the length of the physical packet. Therefore, T_J/T increases with the increasing of the PSDU length leading to more jamming energy cost compared with the IJ scheme proposed in [6].

In order to further improve the jamming energy cost of the flexible interval IJ scheme, we conduct enhanced-FIJ in our simulation study. The enhanced-FIJ is designed by taking the same T_F^1 and T_F^2 as that of the FIJ scheme. While for the "Application Data" transmitted within T_J, the IJ scheme proposed in [6] is applied. That is, T_J is further divided into subjamming intervals and subjamming-free intervals according to the IJ scheme proposed in [6]. The performance of enhanced FIJ-shortest TF and enhanced FIJ-longest TF is shown by green dashed lines and black solid lines in Figures 5 and 6. We found that enhanced FIJ-shortest TF can achieve PER performance almost the same as the IJ scheme while saving 10% energy.

6.2. *Performance Investigation of Proactive FIJ.*

In this subsection, we first generate 600 samples (x_i, t_i) with $x_i = i$, $1 \le i \le 600$. $68\,\mu\text{s} \le t_i \le 3140\,\mu\text{s}$ is generated as follows. (1) Generate a random number following a lognormal distribution with the mean μ being 2 and the standard deviation σ being 0.5. (2) The generated random number first times 785 then is rounded down to a multiple of 4 to match the pattern of the time length of physical packets. (3) If the result is not within the section of proper time length (68 μs-3140 μs), repeat the process until the result falls into the section. (4) Repeat the process until the value of $t_1, t_2, \cdots, t_{600}$ is generated. Then, we train the SVR model with the scaled first 500 samples, that is, $(1, y_1), (2, y_2), \cdots, (500, y_{500})$, to find the hyperplane that fits these 500 samples.

Taking the RBF kernel function into consideration, two parameters need to be set in the SVR model, namely, regularization parameter C and Gaussian kernel parameter δ. We use the grid search technique to find the optimal values. Specifically, the optimal range of regularization parameter C is $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 5, 10, 10^2, 10^3\}$, and the range of

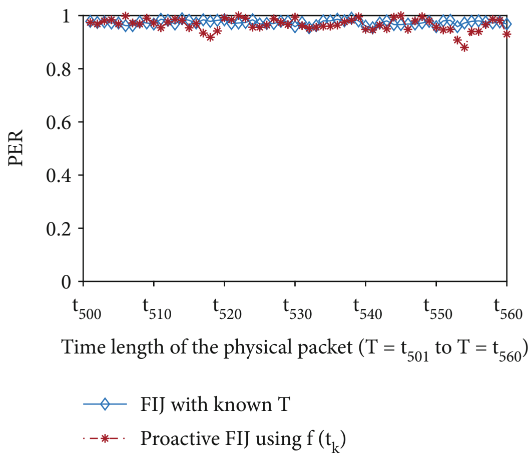


Figure 8: Performance comparison between proactive FIJ and FIJ with known T.

δ is $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$. In our training experiment, we set the constant regularization parameter $C = 5$, the slack variables ξ, ξ^* close to 0, and δ in the Gaussian kernel equal to 100. The training result is shown in Figure 7. Based on the trained SVR model, we predict the value of $f(501)$ then design the length of JI, denoted by T_J^{501}, according to Section 4 with $T = T^{501} = f(501) * t_{\max}/10$ (the reverse conversion of (3)). Sequentially, we train the SVR model with samples $(2, y_2), (3, y_3), \cdots, (501, y_{501})$ and predict the value of $f(502)$; train the SVR model with samples $(3, y_3), (4, y_4), \cdots, (502, y_{502})$ and predict the value of $f(503), \cdots$; and train the SVR model with samples $(60, y_{60})$, $(61, y_{61}), \cdots, (559, y_{559})$ and predict the value of $f(560)$. Then, we will get $T_J^{502}, T_J^{503}, \cdots, T_J^{560}$. After that, we transmit 500 packets for each time length t_k, $501 \le k \le 560$, and jam their transmission according to the obtained T_J^k to observe the PER. The results are given in Figure 8. It can be found that proactive FIJ based on SVR can lead to similar PER compared with the FIJ scheme derived from known time length. Taking an average of the results for $T = t_{501}$ to $T = t_{560}$, the average PER by using proactive FIJ is 96.59%. It is 0.77% less than the average PER achieved by using FIJ with known time length. Proactive FIJ based on SVR can effectively secure the transmission of the physical packets in WAVE-based vehicular networks.

7. Conclusion

In conclusion, FIJ provides a way to save more energy than existing IJ when dealing with eavesdropping attacks in WAVE-based vehicular networks. Proactive FIJ leads to no processing delay for deciding the length of JI and JF thanks to its capability of predicting T. Simulation results confirm that our design is capable of defending eavesdropping attacks while enhancing the performance in energy saving.

Data Availability

No publicly archived dataset has been applied.

Disclosure

Part of the content is included in a conference paper entitled "Flexible Interval Intermittent Jamming against Eavesdropping in WAVE Based Vehicular Networks," which has been accepted by the 9th International Conference on Computational Data and Social Networks (CSoNet 2020). Compared with the conference version, this manuscript enables proactive FIJ by predicting the time length of physical packets based on SVR. Organization and presentation have been improved, and evaluation has been conducted to validate the performance of proactive FIJ. This work is part of the first author Hao Li's Ph.D. dissertation entitled "Enhancing Physical Layer Security in Wireless Communications Using Secrecy Extraction and Friendly Jamming."

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Hao Li and Xiaoshuang Xing contributed equally to this study and share first authorship.

Acknowledgments

This work was supported by the Youth Program of National Natural Science Foundation of China (grant numbers 61802274 and 61602062) and Natural science fund for colleges and universities in Jiangsu Province (grant number 18KJB510044).

References

[1] Q. Gao, Y. Huo, L. Ma et al., "Joint design of jammer selection and beamforming for securing MIMO cooperative cognitive radio networks," *IET Communications*, vol. 11, no. 8, pp. 1264–1274, 2017.

[2] P. Siyari, M. Krunz, and D. N. Nguyen, "Distributed power control in single-stream MIMO wiretap interference networks with full-duplex jamming receivers," *IEEE Transactions on Signal Processing*, vol. 67, no. 3, pp. 594–608, 2019.

[3] Y. Li, R. Zhang, J. Zhang, S. Gao, and L. Yang, "Cooperative jamming for secure UAV communications with partial eavesdropper information," *IEEE Access*, vol. 7, pp. 94593–94603, 2019.

[4] Z. Mobini, M. Mohammadi, and C. Tellambura, "Wireless-powered full-duplex relay and friendly jamming for secure cooperative communications," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 621–634, 2018.

[5] Y. Huo, X. Fan, L. Ma, X. Cheng, Z. Tian, and D. Chen, "Secure communications in tiered 5G wireless networks with cooperative jamming," *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3265–3280, 2019.

[6] X. Xing, G. Sun, J. Qian, D. Yu, and X. Cheng, *Intermittent Jamming for Eavesdropping Defense in Wave Based Vehicular Networks*, Submitted to IEEE Transactions on Wireless Communications, 2020.

[7] J. Wang, Z. Cai, and J. Yu, "Achieving personalized k-anonymity-based content privacy for autonomous vehicles in

CPS," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4242–4251, 2020.

[8] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[9] M. N. Mejri, J. Ben-Othman, and M. Hamdi, "Survey on VANET security challenges and possible cryptographic solutions," *Vehicular Communications*, vol. 1, no. 2, pp. 53–66, 2014.

[10] B. Mokhtar and M. Azab, "Survey on security issues in vehicular ad hoc networks," *Alexandria Engineering Journal*, vol. 54, no. 4, pp. 1115–1126, 2015.

[11] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "VANet security challenges and solutions: a survey," *Vehicular Communications*, vol. 7, pp. 7–20, 2017.

[12] U. Challita, A. Ferdowsi, M. Chen, and W. Saad, "Machine learning for wireless connectivity and security of cellular-connected UAVs," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 28–35, 2019.

[13] J. Cui, J. Zhang, H. Zhong, and Y. Xu, "SPACF: a secure privacy-preserving authentication scheme for VANET with cuckoo filter," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10283–10295, 2017.

[14] Z. Wei, J. Li, X. Wang, and C.-Z. Gao, "A lightweight privacy-preserving protocol for VANETs based on secure outsourcing computing," *IEEE Access*, vol. 7, 2019.

[15] X. Wang, S. Li, S. Zhao, and Z. Xia, "A VANET privacy protection scheme based on fair blind signature and secret sharing algorithm," *Automatika*, vol. 58, no. 3, pp. 287–294, 2017.

[16] L. Feng, Y. Xiu-Ping, and W. Jie, "Security transmission routing protocol for MIMO-VANET," in *Proceedings of 2014 International Conference on Cloud Computing and Internet of Things*, pp. 152–156, Changchun, China, 2014.

[17] S. DasGupta, R. Chaki, and S. Choudhury, "SBRPV: security based routing protocol for vehicular ad hoc networks," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, pp. 745–750, Samsun, Turkey, 2019.

[18] R. Fotohi, Y. Ebazadeh, and M. S. Geshlag, "A new approach for improvement security against DoS attacks in vehicular ad-hoc network," 2020, http://arxiv.org/abs/2002.10333.

[19] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[20] B. Deka, R. M. Gerdes, M. Li, and K. Heaslip, "Friendly jamming for secure localization in vehicular transportation," in *International Conference on Security and Privacy in Communication Networks*, pp. 212–221, Springer, 2014.

[21] T. Tithi, B. Deka, R. M. Gerdes, C. Winstead, M. Li, and K. Heaslip, "Analysis of friendly jamming for secure location verification of vehicles for intelligent highways," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7437–7449, 2018.

[22] H. Lee, S. Eom, J. Park, and I. Lee, "UAV-aided secure communications with cooperative jamming," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9385–9392, 2018.

[23] Y. Allouche, E. M. Arkin, Y. Cassuto et al., "Secure communication through jammers jointly optimized in geography and time," *Pervasive and Mobile Computing*, vol. 41, pp. 83–105, 2017.

[24] S.-M. Huang and J.-F. Yang, "Linear discriminant regression classification for face recognition," *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 91–94, 2013.

[25] N. Prabhakaran and M. S. Sudhakar, "Fuzzy curvilinear path optimization using fuzzy regression analysis for mid vehicle collision detection and avoidance system analyzed on NGSIM I-80 dataset (real-road scenarios)," *Neural Computing and Applications*, vol. 31, no. 5, pp. 1405–1423, 2019.

[26] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[27] J. Kruppa, A. Ziegler, and I. R. König, "Risk estimation and risk prediction using machine-learning methods," *Human Genetics*, vol. 131, no. 10, pp. 1639–1654, 2012.

[28] V. Vapnik and A. Lerner, "Pattern recognition using generalized portrait method," *Automation and Remote Control*, vol. 24, pp. 774–780, 1963.

[29] K. Li, G. Lu, G. Luo, and Z. Cai, "Seed-free graph de-anonymiztiation with adversarial learning," in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pp. 745–754, 2020.

[30] A. Goldsmith, *Wireless Communications*, Cambridge University Press, Cambridge, UK, 2004.

[31] IEEE Computer Society LAN/MAN Standards Committee, "IEEE standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications," in *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, 2012.

[32] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, 2004.

[33] I. Guyon, B. Boser, and V. Vapnik, "Automatic capacity tuning of very large VCdimension classifiers," in *Advances in Neural Information Processing Systems 5*, pp. 147–155, Morgan Kaufmann, 1993.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6383646, 10 pages
https://doi.org/10.1155/2021/6383646

Research Article

Robust Visual Relationship Detection towards Sparse Images in Internet-of-Things

Yang He,[1] **Guiduo Duan**,[1,2] **Guangchun Luo**,[2,3] **and Xin Liu**[3]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2]*Trusted Cloud Computing and Big Data Key Laboratory of Sichuan Province, Chengdu 610000, China*
[3]*School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*

Correspondence should be addressed to Guiduo Duan; duanguiduo@163.com

Received 21 April 2021; Accepted 5 July 2021; Published 20 July 2021

Academic Editor: Yan Huang

Visual relationship can capture essential information for images, like the interactions between pairs of objects. Such relationships have become one prominent component of knowledge within sparse image data collected by multimedia sensing devices. Both the latent information and potential privacy can be included in the relationships. However, due to the high combinatorial complexity in modeling all potential relation triplets, previous studies on visual relationship detection have used the mixed visual and semantic features separately for each object, which is incapable for sparse data in IoT systems. Therefore, this paper proposes a new deep learning model for visual relationship detection, which is a novel attempt for cooperating computational intelligence (CI) methods with IoTs. The model imports the knowledge graph and adopts features for both entities and connections among them as extra information. It maps the visual features extracted from images into the knowledge-based embedding vector space, so as to benefit from information in the background knowledge domain and alleviate the impacts of data sparsity. This is the first time that visual features are projected and combined with prior knowledge for visual relationship detection. Moreover, the complexity of the network is reduced by avoiding the learning of redundant features from images. Finally, we show the superiority of our model by evaluating on two datasets.

1. Introduction

Visual relationship detection tries to simultaneously detect objects for an image and classify the predicate between each pair of these objects [1]. It has been considered as a bridge to semantically connect the low-level visual information [2–7] and high-level semantic information [8–11]. Generally, visual relationships indicate types of relations between objects in images and are usually represented by triplets (subject, predicate, and object), where the predicate can be a verb (person, ride, bicycle), spatial (cat, on, table), preposition (person, with, shirt), and comparative (elephant, taller, person) [1, 12]. The detection of these interactions can uncover diverse knowledge from images and significantly benefits the functionalities of IoT systems. Moreover, potential disclosure of sensitive information [13] can also be inferred with the autonomous relationship detection and provides guidelines for secure multimedia IoT data processing [14–16].

The early studies of visual relationship detection mainly rely on pure visual features capturing the complex visual variance of images [17, 18], suffering from the lack of diverse information for predicate classification. Considering the sparsity of IoT data, both the scale of the image dataset and the details within these images will be constrained. Sensing devices will be conservative on data publication [19, 20], especially when the image data contain abundant semantic information. Meanwhile, images maybe masked or obfuscated before publication due to privacy concerns [21]. Both constraints caused by sparsity of images have aggravated the difficulties for visual relationship detection, and purely visual-feature-based methods are not qualified.

Recently, additional sources of information, such as prior knowledge and semantic information, are incorporated into visual relationship detection [1, 22–24], as extra background information can be utilized to supply and refine the detection. Generally, two essential tasks are considered during the incorporation of additional source of information: (1) How to apply the semantic associations among relationships [12, 25, 26] to refine the detection. For example, the relationship (person, ride, horse) is semantically similar to (person, ride, elephant) as the horse and elephant both belong to animals, even though horse and elephant are quite different in images. In this case, visual relationship detection models should be able to infer (person, ride, elephant) base on examples of (person, ride, horse). (2) How to alleviate the huge semantic space of possible relationships. Assume the category of objects to be N and the predicates to be K. Then, the number of possible relationships is $O(N^2K)$ as a relationship is composed of two objects [27]. Therefore, the size of semantic space in relationship detection increases by orders of magnitude, while many of relationships appear rarely in images. Visual relationship detection models should learn all relationship classes sufficiently.

Towards these tasks, extensive studies have been conducted. They mainly consider how to incorporate the additional source of information into the relationship detection. Initially, Lu et al. [1] introduced the additional language priors from semantic word embeddings to fine-tune the likelihood of a predicted relationship. Subsequently, Zhuang et al. [28] integrated the language representations of the subjects and objects as “context” to derive a better classification result for visual relationship detection. Then, Plummer et al. [8] use a large collection of linguistic and visual cues for the relationship detection in images, which contain attribute information and spatial relations between pairs of entities connected by verbs or prepositions. Furthermore, instead of using the pretrained and fixed language representations directly, Zhang et al. [29] tried to fine-tune the subject and object representations jointly and employ the interaction between visual branches to predict the relationship.

Although these methods achieve significant success, they still tend to focus on the word-level semantics [30] as the additional sources of information and lack in adopting the sophisticated knowledge and deep relations among objects. As for such kind of external knowledge, the knowledge graph is treated as a typical category of structural information providing abundant clues on relations between entities. It has been recently applied for many areas including computer vision and achieves dramatical improvements. Generally, a knowledge graph is a multirelational graph composed of entities (nodes) and relations (different types of edges). Each edge is a kind of relation in the form of triplets (head entity, relation, tail entity), indicating that two entities are connected by a specific relation. This type of additional information can provide more semantic association between objects and relations in an image and could be used for more rational reasoning to improve visual relationship detection. However, its application for visual relationship detection has not yet been properly considered, and neither of the abovementioned tasks is investigated.

To take advantage of this type of information, this paper designs a deep neural network for visual relationship detection by considering the knowledge graph as an additional source of information. The input of the model includes the images and an external knowledge graph, and the outputs are the relationships in images. The proposed model includes a visual module extracting the visual features of images, a knowledge module introducing the additional prior knowledge via the knowledge graph embedding [31], and a mapping module combining the visual features with prior knowledge. Finally, a new loss function based on the triplet loss [32] is designed in the mapping module to tune the projection of visual features into the knowledge space.

The proposed model uses the vector translation of the knowledge space for the first time, to capture the valuable structured information between objects and relations. By this mean, the structured semantic association in a knowledge graph can help improve the relationship detection. The proposed model also learns the objects and predicates and fuses them together to predict the relationship triplets [1]. This method can alleviate the impact of a huge semantic space of possible relationships, by reducing the space from $O(N^2K)$ to $O(N+K)$. Furthermore, the model achieves a reduced scale of parameters compared with state-of-the-art works [31], as it does not request the learning of visual features of predicates. The performance of the model is validated on two relation datasets: visual relationship detection (VRD) [1] with 5,000 images and 6,672 unique relations and visual genome (VG) [12] with 99,658 images and 19,237 unique relations. According to the comparison with several baselines, our model shows the superiority in visual relationship detection. In summary, the main contribution of this paper includes

(1) We propose a novel framework for introducing the prior knowledge in visual relationship detection

(2) Our model incorporates the priors in knowledge graph embedding for the first time to capture the valuable structured information between objects and relations

(3) Our model reduces the parameters for extracting the visual features of predicates and designs a loss for combining the visual feature with the prior knowledge

(4) Extensive evaluation shows that our model outperforms several strong baselines in visual relationship detection

This paper is organized as follows. The related works are introduced in Section 2. The proposed model is described in Section 3. The model is validated in VRD and VG datasets and compared with other methods in Section 4. The conclusion is described in Section 5.

2. Related Work

During the past years, there have been a number of studies in visual relationship detection. The earlier works regard visual relationships as an adminicle to improve the performance for other tasks, such as object detection [33, 34], image retrieval [12, 35, 36], and action recognition [37]. They focus on the specific types of relationships, such as spatial relationships

[2, 38], positional relationships [2, 35, 39], and actions (e.g., the interaction between objects) [40–42].

Lu et al. [1] first formalized the visual relationship as the (subject, predicate, object) triplet, defined the visual relationship detection task, and proposed a method by leveraging the language prior to model the more general correlation between objects. Afterwards, more studies on visual relationship detection have been developed, which can be divided into two categories: joint model and separate model.

For the joint model, it detects (subject, predicate, object) simultaneously by considering the relationship triplets as an integrated body [17, 22, 42–44], e.g., (person, ride, horse) and (person, ride, elephant) are of different classes. Vip-CNN [18] considers each visual relationship as a phrase with three components and formulates the visual relationship detection as three interconnected recognition problems. Plummer et al. [8] learned a Canonical Correlation Analysis (CCA) model on top of different combinations of the subject, object, and union regions and train a RankSVM to learn the visual relationship. However, it requires extremely large training data, because all possible combinations of predicates and entities (subject, object) are treated as independent classes. As a result, the general approaches usually pose the problem as a classification task in limited classes.

For a separate model, it first detects subjects and objects and then recognizes the possible interactions among them [1, 39, 45–47]. VtransE [48] uses the object detection output of a Faster R-CNN network and extracts features from every pair of objects to learn the visual translation embedding for relationship detection. Zhang et al. [29] embed the objects and relations of relationship triplets separately to the independent semantic spaces and then implicitly learn the connections between them via visual feature fusion and semantic meaning preservation in the embedding space.

The method proposed recently by Zhang et al. [29] is the most related one to ours. Compared with this work, instead of the word-level semantic embeddings, our work incorporates the knowledge graph and embeds it in a knowledge space as the additional sources of information. Due to the use of TransE [31] as the knowledge graph embedding, our work barely needs to model the large visual variance of relations in images.

Finally, our method adopts the additional semantic information to guide the visual recognition. This is consistent with the trend of using language information for visual recognition. For example, the language information has also been incorporated into visual question answering [49–52], few-shot learning [53–56], and image-sentence similarity task [57–60].

3. Method

3.1. Overview. The goal of the proposed model is to detect visual relationships from images which requires having discriminative power among a set of relationship categories. However, since object categories are often semantically associated, it is critical for a model to preserve semantic similarities, so as to benefit both frequent and rarely seen relationship categories.

The overview of the proposed model is shown in Figure 1. It consists of three modules, namely, visual module, knowledge module, and mapping module. The visual module detects a set of objects in images and extracts the visual features of the objects. The knowledge module consists of a knowledge graph, which is embedded in a low dimension vector space, so it can be used as the additional source of information. The mapping module considers the image and additional source of information comprehensively, which maps the visual features to the knowledge space for relationship detection. For any valid relationships, they are represented by the triplets (subject, predicate, object) in low dimension vectors **s**, **p**, and **o**, respectively.

Note: in this paper, we use “relation” to refer to “predicate” in previous works and “relationship” to refer to the (subject, predicate, object) triplet. The detailed descriptions of notations can be found in Table 1.

3.2. Visual Module. The design of the visual module is based on the intuition that a relationship exists when its objects exist, but not vice versa. Therefore, to detect the visual relationships from images, the first step is to detect the objects and corresponding visual features in images.

In the visual module, the object detection is based on a Faster-RCNN [61] network with the VGG-16 [62] architecture, composed of a Region Proposal Network (RPN) and a classification layer. In the Faster-RCNN network, convolution does not change the size of the input image.

$$\text{output}_{\text{size}} = \frac{\text{input}_{\text{size}} - \text{kernel}_{\text{size}} + 2\text{pad}}{\text{stride}} + 1. \qquad (1)$$

After that, the Feature Extraction Layer is proposed to extract $\mathbf{x_s}$ and $\mathbf{x_o}$, when suppose $\mathbf{x}_s, \mathbf{x}_o \in \mathbb{R}^M$ are the M -dimensional visual features of the subject and object, respectively. The visual features $\mathbf{x_s}$ and $\mathbf{x_o}$ are obtained by concating the vector from the last convolution feature map in the Faster-RCNN network and the bounding box parameterization in [63].

3.3. Knowledge Module. A knowledge graph is represented by $G(V, E)$, while V is the set of nodes, which represents the entities (subjects, objects), and E is the set of edges, which represents the connections between entities. Hence, the relations between the subject and object can be represented by the connections between the entities in the knowledge graph, mainly describing real world entities and their interrelations organized in a graph. Compared with the word-level external information, this type of additional information can capture a more semantic association between objects and relations and be used for rational reasoning to improve the results of visual relationship detection.

The knowledge module introduces jointly a knowledge graph and projects it into an embedding space, to activate the rich prior knowledge in tuning the relationship detection. Translation embedding (TransE) [31] is a remarkable model that represents a valid relationship (subject, predicate, object) in the knowledge graph in low dimension

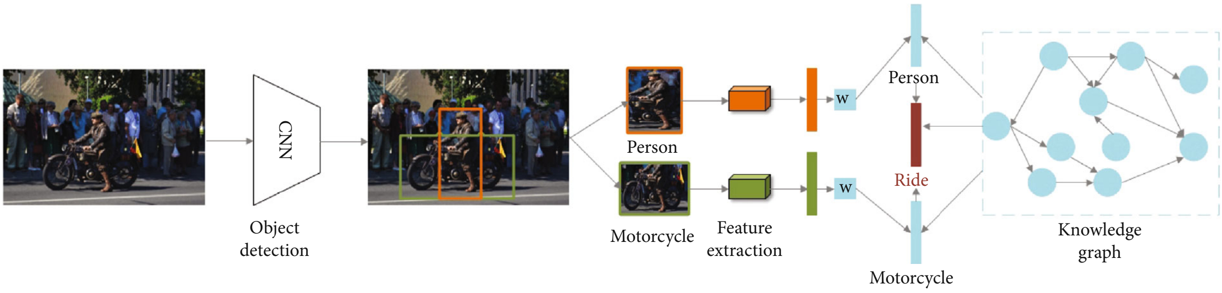


Figure 1: The overview of our visual relationship detection model. It consists of visual module, knowledge module, and mapping module. Visual module uses the CNN to detect a set of objects in images and extracts the visual features of the objects. Knowledge module consists of a knowledge graph, which is embedded in a low dimension vector space. Mapping module maps the visual features to the knowledge space for relationship detection.

vectors $\mathbf{s}$, $\mathbf{p}$, and $\mathbf{o}$, and $\mathbf{s}, \mathbf{p}, \mathbf{o} \in \mathbb{R}^r$, respectively. The relation is represented as a translation in the vector space:

$$\mathbf{y}_s + \mathbf{y}_p \approx \mathbf{y}_o, \tag{2}$$

when the relationship triplet holds, and $\mathbf{y}_s + \mathbf{y}_p \not\approx \mathbf{y}_o$ otherwise.

Since TransE offers a simple but effective method for representing the complex relationships in large knowledge graphs, it is adopted into the knowledge module for representing prior knowledge in the knowledge space. To learn such embeddings for the knowledge graph, we suppose a training set S of triplets (s, p, o) composed of two entities s, $o \in E$ (the set of entities) and a relation $p \in L$ (the set of relations). Since the relation is represented as a translation in the vector space, the energy of a triplet is defined by $d(y_s + y_p, y_o)$, which regard the squared Euclidean distance as a dissimilarity function:

$$d\left(y_s + y_p, y_o\right) = \|y_s\|_2^2 + \left\|y_p\right\|_2^2 + \|y_o\|_2^2 - 2\left(y_s^T y_o + y_p^T (y_o - y_s)\right). \tag{3}$$

To project the knowledge graph to knowledge space, we minimize a margin-based ranking criterion over the training set:

$$\mathscr{L} = \sum_{(s,p,o)\in S} \sum_{\left(s',p,o'\right)\in S'_{(s,p,o)}} \left[\gamma + d\left(y_s + y_p, y_o\right) - d\left(y'_s + y_p, y'_o\right)\right]_+, \tag{4}$$

where $[x]_+$ denotes the positive part of x, $\gamma > 0$ is a margin hyperparameter, and

$$S'_{(s,p,o)} = \left\{\left(s', p, o\right) \mid s' \in E\right\} \cup \left\{\left(s, p, o'\right) \mid o' \in E\right\}. \tag{5}$$

In the knowledge graph embedding, the loss function, constructed according to Equation (4), has lower values of the energy for training triplets than for wrong triplets, so the embeddings for the knowledge graph have the ability to distinguish wrong triplets. As for the wrong triplets, it is constructed according to Equation (5), which is composed of training triplets with either the subject or object replaced by a random entity (but not both at the same time).

3.4. Mapping Module. To consider the image visual feature and extra knowledge feature comprehensively, the mapping module is adopted to learn the joint visual and knowledge embedding. In the mapping module, there is a projection matrix $\mathbf{W} \in \mathbb{R}^{r\times M}$ from the feature space to the knowledge embedding space:

$$\mathbf{y}'_s = \mathbf{W}\mathbf{x}_s, \tag{6}$$

$$\mathbf{y}'_o = \mathbf{W}\mathbf{x}_o, \tag{7}$$

$$\mathbf{y}'_o - \mathbf{y}'_s \approx \mathbf{y}_p, \tag{8}$$

where $\mathbf{y}'_s$ and $\mathbf{y}'_o$ are the vector representations after the projection of $\mathbf{x}_s$ and $\mathbf{x}_o$. To guarantee that the corresponding entities are close to each other during the projection process, a modified triplet loss is employed, where the triplet loss [32] can encourage matched entities from the two modalities to be closer than the mismatched ones by a fixed margin. To this end, two sets of entity triplets for each positive visual-knowledge pair are denoted by $(\mathbf{y}'_E, \mathbf{y}_E)$:

$$\mathrm{tri}_{y'_E} = \left\{\mathbf{y}'_E, \mathbf{y}_E, \mathbf{y}'^{-}_E\right\}, \tag{9}$$

$$\mathrm{tri}_{y_E} = \left\{\mathbf{y}'_E, \mathbf{y}_E, \mathbf{y}^{-}_E\right\}, \tag{10}$$

where $s, o \in E$ and the set $\mathrm{tri}_{y'_E}$ and tri_{y_E} correspond to triplets with negatives from the visual mapping and knowledge space, respectively. If the superscripts $s, o \in E$ are omitted for clarity, the triplet loss $\mathscr{L}^{Tr}$ is the summation of two losses $\mathscr{L}^{Tr}_{y'}$ and $\mathscr{L}^{Tr}_{y}$:

$$\mathscr{L}^{Tr}_{y'} = \sum_{i=1}^{N} \max\left[0, \mathrm{sim}\left(\mathbf{y}'^{-}_{i}, \mathbf{y}_i\right) - \mathrm{sim}\left(\mathbf{y}'_i, \mathbf{y}_i\right) + m\right], \tag{11}$$

$$\mathcal{L}_y^{Tr} = \sum_{i=1}^{N} \max\left[0, \mathrm{sim}\left(\mathbf{y}_i', \mathbf{y}_i^-\right) - \mathrm{sim}\left(\mathbf{y}_i', \mathbf{y}_i\right) + m\right], \quad (12)$$

$$\mathcal{L}^{Tr} = \mathcal{L}_{y'}^{Tr} + \mathcal{L}_y^{Tr}, \quad (13)$$

where $\mathcal{L}_{y'}^{Tr}$ guarantees that entities in knowledge space can be close to the corresponding entities in the visual mapping space, $\mathcal{L}_y^{Tr}$ guarantees that the entities in visual mapping space can be close to the corresponding entities in knowledge space, N is the number of entities, m is the margin between the distances of positive and negative pairs, and sim() is a similarity function, which is the cosine similarity function:

$$\mathrm{sim}\left(\mathbf{y}_i', \mathbf{y}_i\right) = \frac{y_i \cdot y_i'}{\|y_i\| \times \|y_i'\|}. \quad (14)$$

4. Experiments

Datasets: the *visual relationship detection (VRD)* [1] dataset contains 5,000 images with 100 object categories and 70 relations. In total, VRD contains 37,993 relationship annotations with 6,672 unique relationships and 24.25 relationships per object category. We follow the same train/test split as in previous works [1] to get 4,000 training images and 1,000 test images. To demonstrate that the proposed method can work reasonably well on a dataset with small relationship space, experiments in terms of visual relationship detection task are performed in the VRD dataset.

The *visual genome (VG)* [12] dataset is the latest release version (VG v1.4) that contains 108,077 images with 21 relationships on average per image. Each relationship is of the form (subject, relation, object) with annotated subject and object bounding boxes. Since the VG dataset is annotated by crowd workers, the objects and relations are noisy. Therefore, we clean it by removing nonalphabet characters and stop words and use the autocorrect library to correct spelling. Finally, the data is split into 86,462 training images and 21,615 testing images. The statistics for datasets can be found in Table 2.

Knowledge graph: in order to take advantage of the effective background knowledge, the knowledge graph for visual relationship detection is constructed according to the processed image label information and the public knowledge graph, WordNet [64]. To build the accurate knowledge graph, the annotation noise in the dataset should be removed. Firstly, duplicate words are deleted, such as "apple apple" and "dog dog." Secondly, phrases with the same meaning are merged, such as "surfboard" and "surf board." Specifically, the one with more occurrences in the dataset is selected and replaces other phrases with identical meaning. Then, we build the knowledge graph by using the object-object relationship annotations in the dataset.

However, this kind of knowledge graph lacks some common sense information. For instance, it can be helpful to know that a horse is a kind of animal. But if images of horse labels miss the "animal" label, our constructed knowledge graph will also lack in this common sense. Thus, it is necessary to combine our constructed knowledge graph with the semantic knowledge graph, WordNet. First, we collect the new nodes in WordNet which directly connect to the nodes in the constructed knowledge graph. Then, we add these new nodes to our knowledge graph. Finally, we take all of the WordNet edges between these nodes and add them to the knowledge graph.

Table 1: Notations used in this paper.

Notations	Descriptions
$\mathbb{R}^M$	m-dimensional Euclidean space
x, **x**, **X**	Scalar, vector, and matrix, respectively
$\mathbf{x}_s$, $\mathbf{x}_o$	Feature of subject and object in image, respectively
$\mathbf{y}_s$, $\mathbf{y}_o$	Knowledge embedding of subject and object, respectively
$\mathbf{y}_p$	Knowledge embedding of predicate
d	Dissimilarity function
S	Set of relation triplets
E	Set of entities
R	Set of relations
sim	Similarity function

Table 2: Statistics for the datasets.

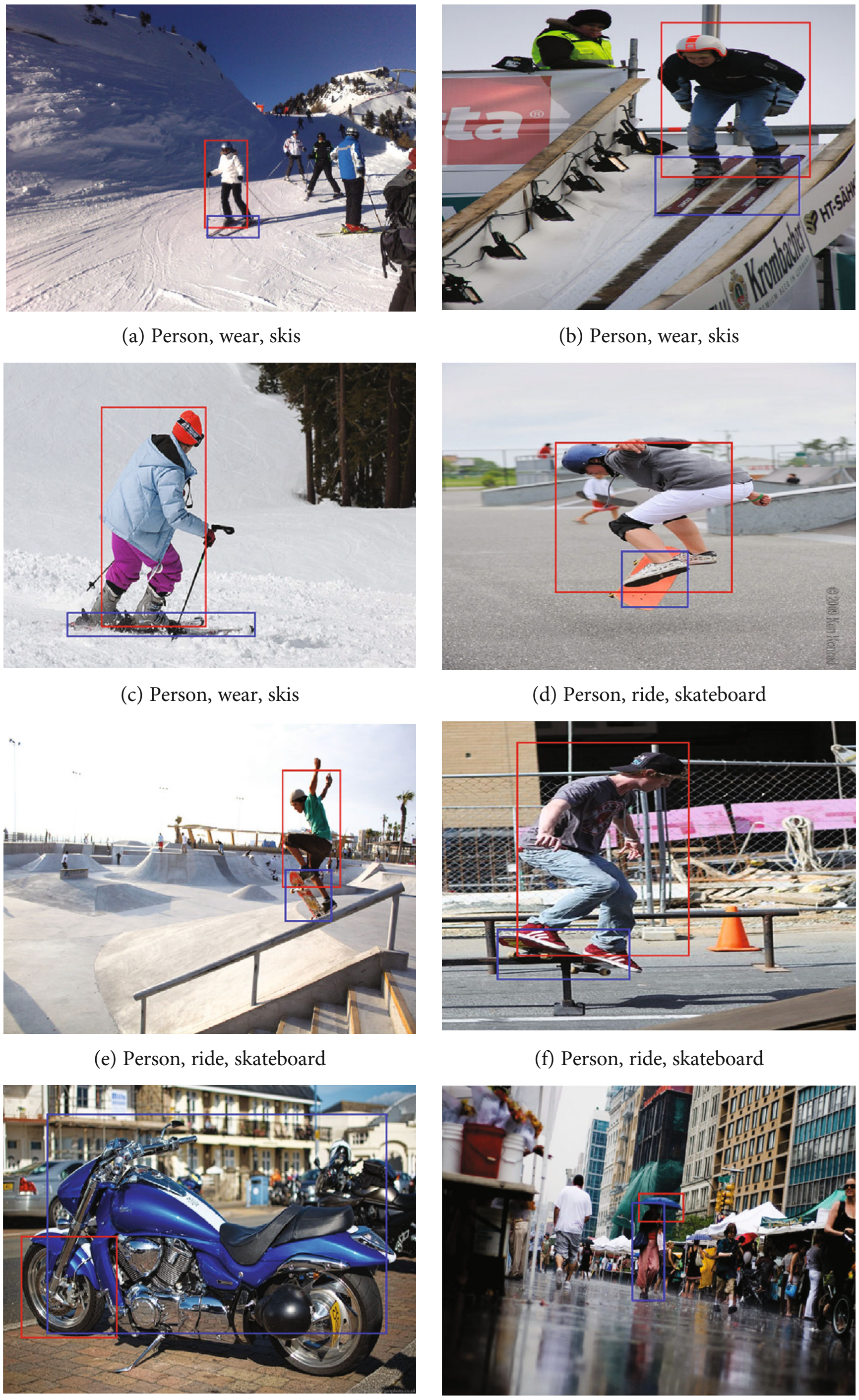
Datasets	Images	Object types	Predicate types	Relationship types
VRD [1]	5,000	100	70	6,672
VG [12]	108,077	200	100	19,237

Table 3: Results on the VRD dataset.

Dataset			VRD			
Task	Phrase det.		Relationship det.		Predicate det.	
Metric	R@50	R@100	R@50	R@100	R@50	R@100
Lu's-V [1]	2.24	2.61	1.58	1.85	7.11	7.11
Lu's-VLK [1]	16.17	17.03	13.86	14.70	47.87	47.87
VtransE [48]	19.42	22.42	14.07	15.20	46.99	46.99
Ours	23.67	25.01	16.56	18.13	48.64	48.64

Table 4: Results on the VG dataset.

Dataset			VG			
Task	Phrase det.		Relationship det.		Predicate det.	
Metric	R@50	R@100	R@50	R@100	R@50	R@100
Lu's-V [1]	—	—	—	—	—	—
Lu's-VLK [1]	—	—	—	—	—	—
VtransE [48]	9.46	10.45	5.52	6.04	61.45	61.70
Ours	9.59	10.52	5.63	6.16	62.52	62.73



(a) Person, wear, skis

(b) Person, wear, skis

(c) Person, wear, skis

(d) Person, ride, skateboard

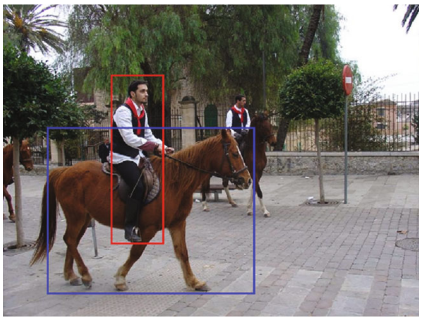
(e) Person, ride, skateboard

(f) Person, ride, skateboard

(g) Wheel, on, motorcycle

(h) Umbrella, cover, person

FIGURE 2: Continued.



(i) Person, ride, horse

Figure 2: Qualitative examples of relationship detection. The red rectangles are identified subjects, the blue rectangles are identified objects, and the captions below are identified visual relationships.

Detecting a visual relationship involves classifying both the objects, predicting the predicate, and localizing both the objects. To study the model's performance for visual relationship detection, the visual relationship detection is measured in three tasks: (1) predicate detection: predict a set of possible predicates between pairs of objects, under the given ground truth object boxes and labels; (2) phrase detection: output a label (subject, predicate, object) and localize the entire relationship as one bounding box; and (3) relationship detection: output a set of (subject, predicate, object) and localize both subject and object in the image simultaneously.

Metrics: Recall@50 (R@50) and Recall@100 (R@100) are adopted as evaluation metrics for detection. R@K computes the fraction of times a correct relationship is predicted in the top *K* confident relationship predictions in an image. Note that precision and average precision (AP) are also widely used metrics, but they are not proper as visual relationships are labeled incompletely and they will penalize the detection if we do not have that particular ground truth.

Compared methods: we compare our model with three representative models. The three visual relationship detection models are as follows: (1) Lu's-V (V-only in [1]): it is a two-stage separate model that first uses R-CNN [63] for object detection and then adopts a large-margin JointBox model for predicate classification; (2) Lu's-VLK (V+L+K in [1]): a two-stage separate model that combines Lu's-V and word2vec language priors [65]; (3) VtransE [48]: a fully convolutional visual architecture that draws upon the idea of knowledge embedding for predicate classification.

4.1. *Comparison on VRD.*

The proposed model is first validated on the small VRD dataset with comparison to the similar methods using the metrics proposed above in Table 3. From the quantitative results, it can be found that the proposed model outperforms other methods in all tasks. Specifically, our proposed model improves performance by 4.25% in the phrase detection task and improves performance by 2.93% in the relationship detection task. These improvements validate the assumption that visual relationships might be helpful for object detection, which can be owed to the incorporation of the knowledge graph as an additional source of information. In addition, the improvement in predicate detection shows that the incorporation of the knowledge graph can provide more meaningful information than the word-level text.

4.2. *Comparison on VG.*

The results of the proposed model on the VG dataset are presented in Table 4. Since VG is a relatively large and newer dataset, some representative models have not been validated on it. In addition, some methods have no public codes, and we can only mark the performance of these methods as a blank in Table 4. Even though the variety of possible relationships becomes more diverse, our proposed model still outperforms other methods in all tasks. Specifically, our proposed model improves performance by 1.07% in the predication detection task. Since the predicate detection isolates the factor of subject/object localization accuracy by using ground truth subject/object boxes and labels, it focuses more on the relationship recognition ability of a model. Therefore, the improvement of our model in this task shows that the incorporation of the knowledge graph is essentially effective for visual relationship detection. Besides, the performance of our proposed model has been improved to some extent, but it is not obvious in phrase detection task and relationship detection task. It may be due to the noise annotations in the large-scale VG dataset and the limited quality of the constructed knowledge graph.

4.3. *Case Study.*

The VRD and VG datasets have densely annotated relationships for images with a wide range of types. From the qualitative results in Figure 2, it shows that our model can clearly detect a wide variety of visual relationship categories. Specifically, in Figures 2(a)–2(c) are the same interactive relationships (person, wear, skis). Figures 2(d)–2(f) are the same positional relationships (person, ride, skateboard). It shows that our model can detect different types of identical relationship, even though their visual representations are quite divergent. Moreover, there are more categories of relationships, such as Figure 2(g) (wheel, on, motorcycle), Figure 2(h) (umbrella, cover, person), and Figure 2(i) (person, ride, horse). It shows that the proposed model can be able to cover all kinds of relationships in (subject, predicate, object), where the predicate can be a verb, spatial, and preposition.

5. Conclusion

The visual relationship detection has been treated as a critical task in enhancing the functionalities of IoTs with CI tools. Considering the sparsity of multimedia IoT data, this work investigates the improvement of visual relationship detection with the knowledge graph as the additional structural semantic information. We proposed a new model for visual relationship detection incorporating the knowledge graph. In the proposed model, the Faster-RCNN and TransE models are used for feature learning from the image and knowledge graph, respectively. A third module is proposed to combine the two parts at the level of low dimensional vectors. Furthermore, a corresponding loss function is designed for the whole network. We validate the effectiveness of the proposed model on several datasets, both on the classification and detection task, and demonstrate the superiority of our approach over other similar methods. The proposed model can be applied for both the knowledge discovery and security analysis for sparse multimedia IoT data. Our future work includes the combination of other techniques like graph neural networks for visual relationship detection, as well as the privacy preservation towards these multimedia IoT data.

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 research work was partly supported by the Sichuan Science and Technology Program (2019YFG0507 and 2020YFG0328) and the National Natural Science Foundation of China (NSFC) (U19A2059). The work was also supported in part by the Young Scientists Fund of the National Natural Science Foundation of China under Grant No. 61802050.

References

[1] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9905*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., pp. 852–869, Springer, Cham, 2016.

[2] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object categorization using co-occurrence, location and appearance," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Anchorage, AK, USA, June 2008.

[3] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," in *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9905*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., pp. 21–37, Springer, Cham, 2016.

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

[5] Z. Xiong, W. Li, Q. Han, and Z. Cai, "Privacy-preserving autodriving: a GAN-based approach to protect vehicular camera data," in *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 668–677, Beijing, China, November 2019.

[6] X. Fan, M. Dai, C. Liu et al., "Effect of image noise on the classification of skin lesions using deep convolutional neural networks," *Tsinghua Science and Technology*, vol. 25, no. 3, pp. 425–434, 2020.

[7] G. Li, Y. Zhao, L. Zhang, X. Wang, Y. Zhang, and F. Guo, "Entropy-based global and local weight adaptive image segmentation models," *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 149–160, 2020.

[8] B. A. Plummer, A. Mallya, C. M. Cervantes, J. Hockenmaier, and S. Lazebnik, "Phrase localization and visual relationship detection with comprehensive image-language cues," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1928–1937, Venice, Italy, October 2017.

[9] H. Izadinia, F. Sadeghi, and A. Farhadi, "Incorporating scene context and object layout into appearance modeling," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 232–239, Columbus, OH, USA, June 2014.

[10] M. Elhoseiny, A. Elgammal, and B. Saleh, "Write a classifier: predicting visual classifiers from unstructured text," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2539–2553, 2017.

[11] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[12] R. Krishna, Y. Zhu, O. Groth et al., "Visual genome: connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[13] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart internet of things systems: a consideration from a privacy perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[14] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.

[15] J. Pang, Y. Huang, Z. Xie, J. Li, and Z. Cai, "Collaborative city digital twin for the covid-19 pandemic: a federated learning solution," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 759–771, 2021.

[16] J. Pang, Y. Huang, Z. Xie, Q. Han, and Z. Cai, "Realizing the heterogeneity: a self-organized federated learning framework for IoT," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3088–3098, 2021.

[17] M. A. Sadeghi and A. Farhadi, "Recognition using visual phrases," in *CVPR 2011*, pp. 1745–1752, Colorado Springs, CO, USA, June 2011.

[18] Y. Li, W. Ouyang, X. Wang, and X. Tang, "ViP-CNN: visual phrase guided convolutional neural network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1347–1356, Honolulu, HI, USA, July 2017.

[19] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2018.

[20] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on*

Distributed Computing Systems (ICDCS), pp. 144–153, Dallas, TX, USA, July 2019.

[21] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, *Generative adversarial networks: a survey towards private and secure applications*, ACM Computing Surveys (CSUR), 2021.

[22] Y. Atzmon, J. Berant, V. Kezami, A. Globerson, and G. Chechik, "Learning to generalize to new compositions in image understanding," 2016, http://arxiv.org/abs/1608.07639.

[23] A. Farhadi, M. Hejrati, M. A. Sadeghi et al., "Every picture tells a story: generating sentences from images," in *Computer Vision – ECCV 2010. ECCV 2010. Lecture Notes in Computer Science, vol 6314*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., pp. 15–29, Springer, Berlin, Heidelberg, 2010.

[24] Y. Wu, X. Zhang, Y. Bian et al., "Second-order random walk-based proximity measures in graph analysis: formulations and algorithms," *The VLDB Journal*, vol. 27, no. 1, pp. 127–152, 2018.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Miami, FL, USA, June 2009.

[26] J. Deng, N. Ding, Y. Jia et al., "Large-scale object classification using label relation graphs," in *Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8689*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., pp. 48–64, Springer, Cham, 2014.

[27] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[28] B. Zhuang, L. Liu, C. Shen, and I. Reid, "Towards context-aware interaction recognition for visual relationship detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 589–598, Venice, Italy, October 2017.

[29] J. Zhang, Y. Kalantidis, M. Rohrbach, M. Paluri, A. Elgammal, and M. Elhoseiny, "Large-scale visual relationship understanding," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9185–9194, 2019.

[30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, http://arxiv.org/abs/1301.3781.

[31] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in neural information processing systems*, pp. 2787–2795, NIPS, 2013.

[32] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: a unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, Boston, MA, USA, June 2015.

[33] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 129–136, San Francisco, CA, USA, June 2010.

[34] M. P. Kumar and D. Koller, "Efficiently selecting regions for scene understanding," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3217–3224, San Francisco, CA, USA, June 2010.

[35] J. Johnson, R. Krishna, M. Stark et al., "Image retrieval using scene graphs," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3668–3678, Boston, MA, USA, June 2015.

[36] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. D. Manning, "Generating semantically precise scene graphs from textual descriptions for improved image retrieval," in *Proceedings of the Fourth Workshop on Vision and Language*, pp. 70–80, Lisbon, Portugal, 2015.

[37] A. Gupta, A. Kembhavi, and L. S. Davis, "Observing human-object interactions: using spatial and functional compatibility for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1775–1789, 2009.

[38] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *International Journal of Computer Vision*, vol. 80, no. 3, pp. 300–316, 2008.

[39] A. Gupta and L. S. Davis, "Beyond nouns: exploiting prepositions and comparative adjectives for learning visual classifiers," in *Computer Vision – ECCV 2008. ECCV 2008. Lecture Notes in Computer Science, vol 5302*, D. Forsyth, P. Torr, and A. Zisserman, Eds., pp. 16–29, Springer, Berlin, Heidelberg, 2008.

[40] B. Yao and L. Fei-Fei, "Grouplet: a structured image representation for recognizing human and object interactions," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 9–16, San Francisco, CA, USA, June 2010.

[41] G. Gkioxari, R. Girshick, and J. Malik, "Contextual action recognition with R∗ CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1080–1088, Santiago, Chile, December 2015.

[42] V. Ramanathan, C. Li, J. Deng et al., "Learning semantic relationships for better action retrieval in images," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1100–1109, Boston, MA, USA, June 2015.

[43] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[44] H. Zhao, X. Puig, B. Zhou, S. Fidler, and A. Torralba, "Open vocabulary scene parsing," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2002–2010, Venice, Italy, October 2017.

[45] C. Desai, D. Ramanan, and C. C. Fowlkes, "Discriminative models for multi-class object layout," *International Journal of Computer Vision*, vol. 95, no. 1, pp. 1–12, 2011.

[46] F. Sadeghi, S. K. Kumar Divvala, and A. Farhadi, "Viske: visual knowledge extraction and question answering by visual verification of relation phrases," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1456–1464, Boston, MA, USA, June 2015.

[47] B. Dai, Y. Zhang, and D. Lin, "Detecting visual relationships with deep relational networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3076–3086, Honolulu, HI, USA, July 2017.

[48] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua, "Visual translation embedding network for visual relation detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5532–5540, Honolulu, HI, USA, July 2017.

[49] J. Donahue, L. Anne Hendricks, S. Guadarrama et al., "Long-term recurrent convolutional networks for visual recognition and description," in *2015 IEEE Conference on Computer Vision*

and Pattern Recognition (CVPR), pp. 2625–2634, Boston, MA, USA, June 2015.

[50] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3128–3137, Boston, MA, USA, June 2015.

[51] S. Antol, A. Agrawal, J. Lu et al., "Vqa: visual question answering," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2425–2433, Abu Dhabi, UAE, October 2015.

[52] P. Wang, Q. Wu, C. Shen, A. Dick, and A. van den Hengel, "FVQA: fact-based visual question answering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 2413–2427, 2018.

[53] A. Frome, G. S. Corrado, J. Shlens et al., "Devise: a deep visual-semantic embedding model," in *Advances in neural information processing systems*, pp. 2121–2129, NIPS, 2013.

[54] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, and K. Kavukcuoglu, "Matching networks for one shot learning," in *30th Conference on Neural Information Processing Systems (NIPS 2016)*, pp. 3630–3638, Barcelona, Spain, 2016.

[55] M. Norouzi, T. Mikolov, S. Bengio et al., "Zero-shot learning by convex combination of semantic embeddings," 2013, http://arxiv.org/abs/1312.5650.

[56] M. Elhoseiny, S. Cohen, W. Chang, B. Price, and A. Elgammal, "Sherlock: scalable fact learning in images," *Thirty-First AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[57] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," 2014, http://arxiv.org/abs/1411.2539.

[58] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun, "Order-embeddings of images and language," 2015, http://arxiv.org/abs/1511.06361.

[59] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, "A multi-view embedding space for modeling internet images, tags, and their semantics," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 210–233, 2014.

[60] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "VSE++: improving visual-semantic embeddings with hard negatives," 2017, http://arxiv.org/abs/1707.05612.

[61] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[62] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, http://arxiv.org/abs/1409.1556.

[63] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[64] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Machine Learning*, vol. 94, no. 2, pp. 233–259, 2014.

[65] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Curran Associates Inc.*, pp. 3111–3119, Red Hook, NY, USA, 2013.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6030144, 11 pages
https://doi.org/10.1155/2021/6030144

Research Article

A Road Network Enhanced Gate Recurrent Unit Model for Gather Prediction in Smart Cities

Mingchao Yuan,[1] **Ling Tian**,[1,2] **Ke Yan**,[1,2] **and Xu Zheng**[1,2]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2]*Trusted Cloud Computing and Big Data Key Laboratory of Sichuan Province, Chengdu 610000, China*

Correspondence should be addressed to Ling Tian; lingtian@uestc.edu.cn

Received 20 April 2021; Accepted 1 July 2021; Published 15 July 2021

Academic Editor: Yan Huo

Gather prediction is an indispensable part of smart city projects. The city government can respond in advance based on gather predictions and greatly reduce the loss and risks caused by vicious gatherings. Compared with other trajectory prediction tasks (i.e., the recommendation of point of interest), gather prediction pay more attention to real-time trajectory data and requests stronger spatial-temporal dependence. At the same time, gather prediction is more focused on scenes with multiple types of trajectories. And the existing methods majorly rely on the trajectory data and ignore the great influence of geographical environment (i.e., road network structure). Therefore, this paper transforms the gather prediction into the trajectory prediction task with strong real-time condition in a certain city and conducts the gathering situations by predicting users' aggregated movements in next minutes or hours. A novel Spatiotemporal Gate Recurrent Unit (STGRU) model is proposed, where spatiotemporal gates and road network gate are introduced to capture the spatiotemporal relationships between trajectories. Compared with existing methods, we improve the performance of the model by adding road network structure and external knowledges, as well as time and distance gates to reduce model parameters. The proposed STGRU is evaluated on three real-world trajectory datasets, and the experimental results demonstrate the effectiveness of the proposed model.

1. Introduction

In recent years, various problems caused by the gathering of people are one of the main reasons hindering urban construction and development. Crowd gatherings are prone to various accidents, such as stampede, fighting, and wounding, which place high demands on the city's management and control capabilities. Therefore, gather prediction can greatly help the city government react in advance and significantly reduce the losses and risks caused by vicious gatherings [1, 2]. As an indispensable part of smart transportation, gather prediction mainly leverages the trajectory data collected by various Internet of Things sensing devices [3–6] (such as mobile phones [7], cars, and other GPS devices [8]). These trajectory data include multiple types of patterns such as walking, driving, and public transportation. Firstly, the city government needs to predict the gathering situation to take preventive measures in advance and combine the geographical features such as rivers and buildings to divide the regions. Then, the city government may process all the trajectory data at the current moment in the city by predicting its location at a certain time in the future. Finally, the gather is obtained through statistical methods or clustering algorithms on trajectory prediction results.

In order to achieve high-accuracy trajectory prediction, current methods majorly focus on modeling the sequential of the trajectory data and the time interval and distance interval between adjacent trajectory points. The main object is to integrate temporal and spatial features to model user behavior patterns. Typical techniques like recurrent neural networks (RNN) [9], Long short-term memory (LSTM) [10], and Gate Recurrent Unit (GRU) [11] have been successfully applied to various types of sequential data modeling and have greatly improved performance. However, none of the above methods consider time intervals and geographic informations [12] in the trajectory data. Some recent works are devoted to extending RNN and LSTM to enable modeling of time and distance intervals between neighbor points. For example, ST-RNN [13] tries to model spatiotemporal context by extending RNN, and HST-LSTM [14] merged the spatiotemporal influence into LSTM. Recently, STGN [15] achieves SOTA by designed two pairs of time and distance gates to model the time interval and distance interval separately.

Nevertheless, trajectory data applied for gather prediction usually suffers the data sparsity [16], due to the uneven sampling interval and distribution of sensing devices. Previous efforts tried to apply spatial-temporal relations to mitigate the problem of data sparsity, but the effect is not obvious. Inspired by Li et al. [17], geographic environment information (such as road network structure) and external knowledge (such as weather information and holiday information) can effectively alleviate the problem of data sparsity. The impact of the geographic environment is essential for the modeling of short-term and long-term behavior patterns of users, and weather and holiday information will affect the overall behavior of users. For example, if the user's continuous trajectories are on the same road segment, it can be judged that the current behavior patterns are similar. Meanwhile, the long-term historical trajectory road network information can well assist in modeling the long-term behavior pattern of users. Furthermore, on weekends, more people are willing to visit more distant areas and stay in a certain location for a long time. All these side information can benefit mitigate the problem of data sparsity and improve the performance of gather prediction.

In order to make full use of external knowledge, this paper proposes a new spatiotemporal gated network by integrating road network structure and external knowledge, named Spatiotemporal Gate Recurrent Unit (STGRU). One pair of time gate and distance gate is designed to capture the short-term behavior pattern by utilize time and distance intervals, and a road network gate is introduced to memorize road network structures to model geographical environment constraints.

The proposed model abstracts the road network structure of the city into a planar graph and extracts the road network structure of a certain track point. And the weather and holiday information are integrated into the track information for input. Moreover, STGRU can model the long-term and short-term behavior patterns of users and reduce the scale of model parameters to a certain extent. Finally, the proposed model processes the trajectory prediction results with statistic methods to achieve gather prediction. Experiments show that considering the road network structure and external knowledge can effectively improve the performance of the model.

Our contributions are summarized below:

(1) Based on the standard Gate Recurrent Unit, we proposed a Spatiotemporal Gate Recurrent Unit (STGRU) model, which integrates road network structure and external knowledge, and reduces the amount of model parameters to a certain extent

(2) We propose an innovative gating mechanism, adding road network gate, which can model the road network structure for learning spatiotemporal relationships between users' trajectories

(3) We evaluate the proposed method on three real-world datasets including population data of Nagoya, Osaka, and Tokyo. The comprehensive comparisons with the state-of-the-art methods show the effectiveness of our model

2. Related Work

2.1. Smart Transportation and Trajectory Data. Smart transportation is a major component of smart city. The main research difficulty is the analysis and decision-making reaction of traffic information.

Traffic information analysis [18] includes traffic flow forecasting and traffic demand forecasting. Traffic flow forecasting [19, 20] and traffic congestion forecasting can help better regulate and control traffic and can effectively alleviate traffic congestion. The taxi demand forecasting method proposed by Geng et al. [21] can help taxi companies to better allocate vehicles. Li et al. [22] proposed a method for forecasting the demand for shared bicycles, which can optimize resource scheduling.

Mining and analysis of trajectory data can assist in traffic planning decisions. Wei et al. [23] used the number of stops and the parking position to analyze the effectiveness of the main line coordination.

2.2. Spatiotemporal Data Modeling. On the other hand, trajectory data is a type of spatiotemporal data, with two dimensions of time and space. Data mining of spatiotemporal data is very difficult, and it is also one of the current research hotspots. The Markov chain-based model [24] is a classic sequence model. And deep learning methods [25] such as RNN, LSTM, and GRU have excellent results in time modeling. The method based on matrix factorization [26] or tensor factorization [27] can model spatial features. CNN [28, 29] and GCN [30] are currently the best spatial modeling methods.

In order to capture the spatiotemporal features, Al-Molegi et al. (2016) proposed STF-RNN [31] to learn different temporal and spatial features. The TGCN [32] proposed by Zhao et al. uses GRU and GCN stacking to model spatiotemporal features. The STGCN [33] innovatively used CNN to model temporal features and achieved good results.

2.3. Gather Prediction. Gather prediction is an indispensable part of smart transportation, which includes many application scenarios, such as hotspot area analysis, passenger flow prediction, and population transfer prediction. Tomaharu et al. [34] proposed a collective graphical model to predict the transition populations between areas. Verma et al. [35] use trajectory data to mine hotspots and realized large-granularity gather prediction. Ni et al. [36] through passenger flow forecasting realized the gather prediction between cities. Kumar et al. [37] used trajectory clustering and similarity analysis for gather prediction. Gather prediction also can be transformed into a multitrajectory prediction task

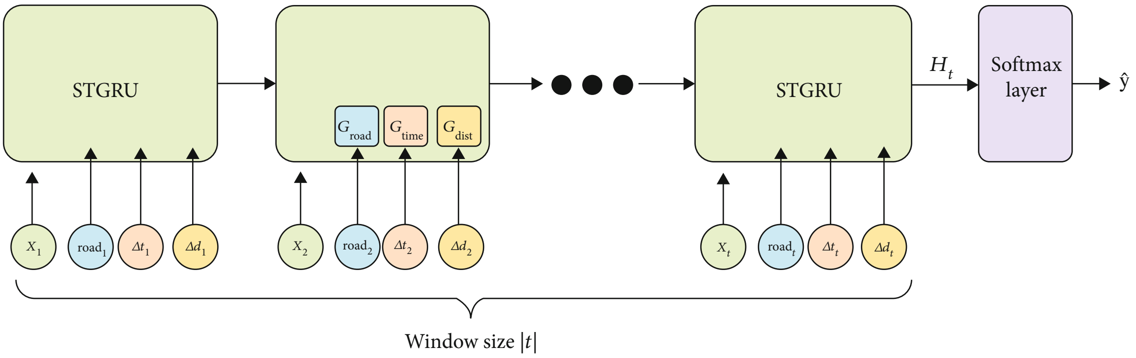


Figure 1: Gather prediction network model based on STGRU.

under the same time and space, which is focus on the time and space characteristics between multiple trajectories.

2.4. Trajectory Prediction. Different with other prediction tasks, the main features of trajectory prediction are geographic information and time information. Trajectory prediction can also use position semantics, speed, and direction. Based on the traditional probability, matrix factorization decomposes the matrix with a low-rank matrix to obtain the implicit feature vector of the user and the trajectory. Tensor factorization expands to three dimensions, including user, time information, and spatial information. Kurashima et al.'s [38] sampling is based on the subject and the distance between the user and the historical location. Liu et al. [39] combined location semantics to embed geographic context information. Research has shown that the sequence between consecutive trajectory points plays a vital role in trajectory prediction, and it is more significant in strong real-time trajectory data, because human behavior patterns are sequential. For prediction based on sequential data, the Markov chain model [40] is the most classic. Cheng et al. proposed a tensor-based model, named FPMC-LR [41], by fusing first-order Markov chains and distance constraints. Feng et al. proposed a personalized ranking metric embedding method (PRME) [42], which embeds the state at all times uniformly, and calculate the Euclidean distance between vectors to measure the similarity.

Neural networks are widely used in various tasks because they can learn to model various nonlinear features. The ST-RNN proposed by Liu et al. (2016) is the first method to introduce a deep neural network into trajectory prediction, ST-RNN uses spatiotemporal information to expand RNN, and its effect is improved. STF-RNN replaced the transition matrix with the internal representation of automatically extracted spatiotemporal features, which can more effectively discover useful features related to model human behavior. Zhu et al. [43] considered modeling time intervals to improve performance and equipped LSTM with time gating. Yang et al. [44] used neural network models to model social network structure and user trajectory behavior patterns. HST-LSTM introduces spatiotemporal factors into the gates existing in LSTM to model spatiotemporal features.

A recently proposed STGN considers the spatiotemporal context. Our proposed STGRU has the following differences from STGN. First, STGRU is extended based on GRU, which reduces the amount of parameters and is more suitable for real-time trajectories. STGN adds time and space gates to LSTM, and the amount of parameters is more than twice that of LSTM. Secondly, STGRU is equipped with external knowledge gate to extract the road network structure to enhance the spatiotemporal characteristics and the influence of external knowledge on the overall movement pattern. However, STGN is only based on the trajectory of a single user, and it is difficult to capture the spatiotemporal relationship between users.

3. Method

In this section, we firstly give the definitions of gather prediction and introduce preliminaries for GRU. Then, we propose Spatio-Temporal Gated Recurrent Unit (STGRU), which uses time and distance intervals and road network structure to model short-term and long-term behavior patterns of users.

3.1. Overview. As shown in Figure 1, we perform trajectory prediction by stacking a STGRU layer and a softmax layer, then compare the result of trajectory prediction with the threshold η to obtain the result of gather prediction.

In our proposed Spatio-Temporal Gated Recurrent Unit, three gates are designed to extract spatiotemporal features and model user behavior patterns. The time gate and the distance gate can learn the time interval and distance interval in the trajectory, obtaining users' short-term behavior patterns. The road network gate aims to capture road network structural features which have the impact on short-term and long-term behavioral patterns.

In this paper, we only discuss the meshing method of dividing area, because meshing has the highest applicability. The STGRU model is also applicable to other dividing area methods, and has been echoed in comparative experiments.

3.2. Problem Formulation. Let $\mathbb{U} = \{u_1, u_2, \cdots, u_M\}$ be the set of M users. And according to the side length a, divide the city into a number of grids and number them, where each grid area is a^2. Each grid corresponds to a unique area ID r. For user u, she has a sequence of historical regions visited up to time t_{i-1} represented as $H_i^u = r_{t_1}^u, r_{t_2}^u, \cdots, r_{t_{i-1}}^u$, where $r_{t_i}^u$ means the region user u visits at time t_i.

The goal of gather prediction is to predict the regions where all users are located at time t_i. Specifically, the higher

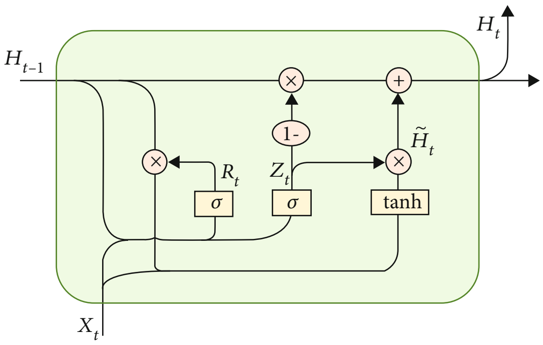


Figure 2: The standard GRU model.

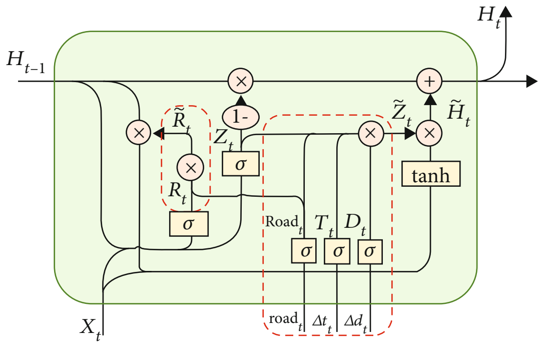


Figure 3: STGRU has time gate, distance gate and road network gate, i.e., T_t, D_t, and Road_t.

the prediction score s^u_{r,t_i} of user u for the region r at time t_i, means the higher probability that the user u would like to located in the region r at time t_i.

According to the prediction scores of all users, predictions of $M \times k$ possible regions can be obtained. The number of people in the region r can be obtained by counting the prediction results. And the gather in the region r can be judged whether there the number is through the threshold η:

$$\text{number}_{r,t_i} = \sum_{u \in U} s^u_{r,t_i},$$
$$r_{\text{state}} = \begin{cases} 1, & \text{number}_{r,t_i} \geq \eta \\ 0, & \text{number}_{r,t_i} < \eta \end{cases}, \tag{1}$$

where r_{state} is the gather state of region r, 1 means there is clustering in the region r, 0 is not, and number_{r,t_i} is the number of people in the region r at time t_i.

3.3. Gated Recurrent Unit. GRU (Cho et al. 2014), a variant of LSTM, can also learn the long dependency problem in RNN networks. The structure of GRU is simpler than that of the LSTM network, while the effect is very good. In order to reduce the amount of parameters to be more in line with real-time data types, our method uses standard GRU as show in Figure 2. Based on the standard LSTM network, GRU combines the forget gate and input gate of LSTM into a single update gate, removes the cell state, and uses the hidden state to transfer information. The basic update formula of GRU is as follows:

$$R_t = \sigma(W_{xr}[H_{t-1}, X_t] + b_r), \tag{2}$$

$$Z_t = \sigma(W_{xz}[H_{t-1}, X_t] + b_z), \tag{3}$$

$$\tilde{H}_t = \tanh\,(W_{xh}X_t + W_{hh}(R_t \odot H_{t-1}) + b_h), \tag{4}$$

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t. \tag{5}$$

Assuming that the number of hidden units is h, the batch input $X_t \in \mathbb{R}^{n\times d}$ at given time step is t, and the hidden state of the previous time step is H_{t-1}. $R_t, Z_t \in \mathbb{R}^{n\times h}$ represents the reset gate and update gate, where $\sigma(\cdot)$ is the logistic sigmoid function. $\tilde{H}_t \in \mathbb{R}^{n\times h}$ represents the candidate hidden state at time step t, where $\tanh\,(\cdot)$ is the double tangent function. $W_{xr}, W_{xz}, W_{xh}, W_{hr}, W_{hz}, W_{hh} \in \mathbb{R}^{d\times h}$ is the weights of gates. b_r, b_z, b_h is corresponding biases. And e represents for the element-wise (Hadamard) product.

The reset gate R_t controls how the hidden state of the previous time step flows into the candidate hidden state of the current time step and captures long-term dependencies in the time series. The update gate Z_t can control how the hidden state should be updated by the candidate hidden state containing the current time step information and capture short-term dependencies in the time series.

3.4. Components. As shown in two dotted red rectangles in Figure 3, STGRU have added time gate, distance gate, and road network gate, which are denoted as T_t, D_t, and Road_t, respectively. T_t and D_t are used to model the influence of time interval and distance interval on trajectory prediction, and Road_t is used to capture the influence of road network structure on behavior patterns. Based on GRU, time gate, distance gate, and road network gate equations are as follows:

$$\begin{aligned} T_t &= \sigma(W_{xt}X_t + \sigma(\Delta t_t W_t) + b_t), \\ D_t &= \sigma(W_{xd}X_t + \sigma(\Delta d_t W_d) + b_d), \\ \text{Road}_t &= \sigma(W_{x\text{road}}X_t + \sigma(\text{road}_t W_{\text{road}}) + b_{\text{road}}). \end{aligned} \tag{6}$$

Combining T_t, D_t, and Road_t, the calculation equation for reset gate and update gate is added as

$$\begin{aligned} \tilde{R}_t &= R_t \odot \text{Road}_t, \\ \tilde{Z}_t &= Z_t \odot T_t \odot D_t \odot \text{Road}_t, \end{aligned} \tag{7}$$

then modify Eqs. (4) and (5) to

$$\begin{aligned} \tilde{H}_t &= \tanh\,\left(W_{xh}X_t + W_{hh}\left(\tilde{R}_t \odot H_{t-1}\right) + b_h\right), \\ H_t &= \tilde{Z}_t \odot H_{t-1} + \left(1 - \tilde{Z}_t\right) \odot \tilde{H}_t, \end{aligned} \tag{8}$$

where Δt_t is the time interval and Δd_t is the distance interval. r_t represents road network information. T_t is equivalent to the time interval input information filter, D_t is equivalent to the distance interval input information filter, and Road_t is used to capture the input information of the road network structure. Calculate the influence of the road network gate on the reset gate and the influence of multiple gate controls on the update gate by adding a new reset gate state $\tilde{R}_t$ and a new update gate state $\tilde{Z}_t$. The influence of the gates determines the influence on the hidden stateH_t.

The candidate hidden state $\tilde{H}_t$ is determined by input information, reset gate, and the hidden state of the previous time step. The second reset gate state $\tilde{R}_t$ is designed to memory the user's long-term road network access information. Road_t is used to memorize the road network information r_t , then transferred to $\tilde{R}_t$, further to $\tilde{H}_t$, and help simulate the long-term behavior pattern of users.

The update gate Z_t can capture short-term dependencies. Therefore, a time gate and a distance gate are designed, combined with the above road network gate to control update gate state. T_t memorizes the Δt_t between the track points, referring to LSTM, and uses element-wise (Hadamard) product to incorporate it into the second update gate state $\tilde{Z}_t$. Similarly, D_t memorizes the Δd_t between the track points, and Road_t memorizes the road network information and integrates it into $\tilde{Z}_t$. Modeling the distance interval can help capture the user's spatial behavior patterns, and modeling the time interval can help capture the user's behavior patterns such as speed and state. Modeling the road network structure can help capture users' short-term behavior constraints and long-term goals, as well as capture the spatial relationships between users.

The method of adapting the model for gather prediction is as follows. First, calculate the time interval and distance interval between track points, and H^u can be converted to

$$[(r_1^u, 0, 0), (r_2^u, t_2^u - t_1^u, d(l_1, l_2)), \cdots, (r_n^u, t_n^u - t_{n-1}^u, d(l_{n-1}, l_n))]. \tag{9}$$

Secondly, add weather, holiday information, and road network information and further transform it into

$$(r_n^u, \text{weather}_n, \text{date}_n, t_n^u - t_{n-1}^u, d(l_{n-1}, l_n), \text{road}_n), \tag{10}$$

where r_t^u contains longitude, latitude, and located region; weather_t contains the highest temperature, lowest temperature, and average temperature; and date_t is marked with 0 or 1 according to whether the date is a holiday. Then, X_t in STGRU is equivalent to $(r_t^u, \text{weather}_t, \text{date}_t)$, Δt_t is equivalent to $t_t^u - t_{t-1}^u$, and Δd_t is equivalent to $d(l_{t-1}, l_t)$, where $d(\cdot, \cdot)$ is the function that computes the distance between two track points. road_t is a vector, which is concatenated by the node where the trajectory point is located in the road network graph and the neighbor nodes. For example, the road segment where the trajectory point i is located represents nod e_i, and the c neighbor nodes of node_i are, respectively, represented as $\text{node}_{i+1}, \text{node}_{i+2}, \cdots, \text{node}_{i+c}$. Then, road_t can be expressed as $\text{road}_t = (\text{node}_i, \text{node}_{i+1}, \cdots, \text{node}_{i+c})$. In addition, in order to extract the behavior pattern of the group, we performed a unified modeling for all users and deleted the user ID.

This paper uses a single-layer network model for comparison experiments, adds softmax layer for output, and uses the loss function of categorical cross entropy:

$$J = -\sum_{i=1}^{K} y_i \log (p_i). \tag{11}$$

Finally, the forecast results are counted in the same time and space. When the statistical value of a certain region r exceeds the set threshold η, it is considered that this region will gather.

3.5. Analysis and Training. STGRU reduces the amount of model parameters. The parameter quantity of a single STGRU unit can be calculated as $3 \times d_h^2 + 6 \times d_h \times d_x + 6 \times d_h$, where d_n is the number of hidden units, and d_x is the number of input units. Similarly, the unit parameter quantity of a LSTM unit can be calculated as $4d_h^2 + 4 \times d_h \times d_x + 4 \times d_h$. The GRU unit has one less gate than the LSTM unit, and the amount of parameters is also reduced accordingly, which can be calculated as $3 \times d_h^2 + 3 \times d_h \times d_x + 3 \times d_h$. The STGN model is the current SOTA method and is an improved model based on LSTM. The unit parameter of STGN can be calculated as $5 \times d_h^2 + 8 \times d_h \times d_x + 10 \times d_h$. The number of parameters of STGRU is slightly more than that of LSTM, which is one-third to one-half less than STGN.

The optimizer use Adam, a variant of Stochastic Gradient Descent (SGD), which comprehensively considers the gradient's first moment estimation (first moment estimation, the mean value of the gradient) and second moment estimation (second moment estimation), and calculates the update step length of parameters. It can automatically adjust the learning rate, and it is very suitable for large-scale data and parameter scenarios.

4. Experiments

In this section, we conduct experiments to evaluate the performance of our proposed model STGRU on three real-world datasets.

4.1. Dataset. We evaluated our proposed method on three SNS-based people flow datasets released by Nightley and Center for Spatial Information Science at the University of Tokyo (CSIS). Each dataset contains trajectories, as detailed follows.

(i) Nagoya People Flow (NPF): NPF dataset contains the trajectories of 2387 users in the Nagoya area in 2013, with a total of 1,068,064 track points. The trajectory time is 6 days, which are July (7/22, 7/28), September (9/16, 9/22), and December (12/24, 12/29). Each trajectory starts from 0:00 to 23:55 of the current day, and the data interval is 5 minutes

(ii) Osaka People Flow (OPF): OPF dataset contains 4924 users in the Osaka area in 2013, and the covering time is July (7/22, 7/28), September (9/16, 9/22), and December (12/24, 12/29). The total number of track points is 2,552,883

(iii) Tokyo People Flow (TPF): TPF dataset is the largest of the three datasets and contains the trajectories of 11536 users amount of 6,883,245 track points in the Osaka area in 2013. The time is July, 2013 (7/1, 7/7), October (10/7, 10/13), and December (12/16, 12/22).

We eliminate user data whose trajectory length was less than 30 in the three data sets and then take 70% of the users as the training set and the remaining 30% as the testing set.

The three datasets do not contain the area information of the track points. This paper divides the area of the track according to 5km × 5km in each area and determines the area where the track points are located according to the latitude and longitude of the track points in the dataset. A sliding window is used to generate samples on both training and test data, and the time interval is randomized within the sliding window to increase the complexity of the trajectory.

For example, if the time of the first track point is 8 : 00 am, the random interval is [7, 12, 18, 24, 30, 31]. Assuming the random number 3, then the time interval between the second track point and the first track point is 3 × 5 min = 15 min, which the second track point times are 8 : 15 am.

4.2. Baseline Methods. We compare our proposed model STGRU with five representative methods for trajectory prediction.

(i) RNN [9]: it passes the state cyclically in its own network; so, it can accept a wider range of time series structure input and widely used for time series prediction tasks

(ii) LSTM [10]: this model is suitable for processing and predicting important events with very long intervals and delays in time series. To a certain extent, the problem of gradient disappearance and gradient explosion of RNN is solved

(iii) GRU [11]: a variant of the LSTM model, which has fewer parameters than LSTM and shows better performance on certain smaller and less frequent datasets

(iv) HST-LSTM [14]: it integrates spatiotemporal influence into the three gates of LSTM. Since there is no session information in the datasets, its ST-LSTM vision is used here

(v) STGN [15]: obtained by enhancing LSTM, introducing two pairs of spatiotemporal gates to capture spatiotemporal relationships

Table 1: Evaluation of prediction results in terms of Recall@K and AUC on three datasets.

(a)

NPF	Recall@1	Recall@5	Recall@10	Recall@20	AUC
LSTM	0.0428	0.1677	0.2894	0.4300	0.6951
GRU	0.0712	0.2372	0.3421	0.4771	0.7778
RNN	0.0809	0.2580	0.3570	0.4865	0.8018
ST-LSTM	0.0621	0.2438	0.3663	0.5050	0.7976
STGN	0.0762	0.2728	0.3734	0.5061	0.8177
STGRU	0.0920	0.2829	0.3891	0.5231	0.8290

(b)

OPF	Recall@1	Recall@5	Recall@10	Recall@20	AUC
LSTM	0.0383	0.1638	0.2634	0.4375	0.7140
GRU	0.0455	0.1979	0.3007	0.4656	0.7611
RNN	0.0588	0.2258	0.3260	0.4881	0.8324
ST-LSTM	0.0502	0.2107	0.3174	0.4890	0.7817
STGN	0.0633	0.1699	0.2833	0.4920	0.8292
STGRU	0.0681	0.2439	0.3464	0.5116	0.8545

(c)

TPF	Recall@1	Recall@5	Recall@10	Recall@20	AUC
LSTM	0.0752	0.3331	0.4849	0.6428	0.8317
GRU	0.0789	0.3319	0.4793	0.6384	0.8451
RNN	0.0856	0.3634	0.5066	0.6561	0.8645
ST-LSTM	0.0864	0.3699	0.5213	0.6682	0.8727
STGN	0.0900	0.3327	0.5103	0.6672	0.8734
STGRU	0.0933	0.3795	0.5263	0.6730	0.8755

4.3. Evaluation Metrics. In order to evaluate the performance of our proposed STGRU model and compare it with the above five baselines, we used two standard metrics area under curve (AUC) and Recall@K. The trajectory prediction task is essentially a multiclassification task, and AUC metrics can better evaluate the classification effect. Recall@K is defined as the ratio of the number of correct predictions to the total number of predictions. First, all possible regions are arranged in descending order according to their probability. Then, the recall score is calculated as the percentage of the number of times the true region is found among the top K most likely regions. In this paper, use $K = 1, 5, 10, 15$, and 20 to illustrate different results of Recall@K. U is the set of users, L_u represents the set of real regions of user u in the testing data, and $P_{K,u}$ denotes the set of top K predicted regions; the calculation formula for Recall@K is:

$$\text{Recall@}K = \frac{1}{|U|}\sum_{u\in U}\frac{|L_u \cap P_{K,u}|}{|L_u|}. \tag{12}$$

4.4. Results and Discussions

4.4.1. Method Comparison. Table 1 shows the performance of our proposed model STGRU and the performance of the six baselines evaluated by Recall@K and AUC on three datasets. The hidden state size is set to 32 in our experiment, the number of epochs is set to 200, and the batch size is set to 512. The

Table 2: The performance with different cell sizes.

Cell size	Recall@1	Recall@5	Recall@10	Recall@20
32	0.0933	0.3795	0.5263	0.6730
64	0.0922	0.3836	0.5253	0.6754
128	0.0942	0.3774	0.5217	0.6736
256	0.0923	0.3761	0.5238	0.6704
512	0.0913	0.3760	0.5190	0.6658

Table 3: The performance with different times and distance gates.

NPF	Recall@1	Recall@5	Recall@10	Recall@20
GRU	0.0712	0.2372	0.3421	0.4771
GRU + D_t	0.0887	0.2763	0.3778	0.5080
GRU + T_t	0.0844	0.2744	0.3773	0.5099
GRU + D_t + T_t	0.0909	0.2801	0.3805	0.5116

sliding window size is set to 10, and the random time interval in the Nagoya dataset and the Osaka dataset is 1 to 3. The random time interval of the Tokyo dataset is 1 to 5, because the data density in the Tokyo data set is higher, and needs to increase the complexity of the data by increasing the random interval. To be fair, all baseline experiments in this paper use the same hyperparameter settings.

From the experimental results, the following observations can be obtained: The STGRU model we proposed is significantly better than the existing state-of-the-art methods in all indicators of the three datasets. The performance gains provided by STGRU over these five counterparts are about 18.1%-110.2%, 5.7%-74.7%, and 3.7%-24.1% in terms of Recall@1 metric in Nagoya, Osaka, and Tokyo datasets, respectively. The results show that the mechanism of modeling the road network structure in STGRU can better model user behavior patterns, modeling short-term temporal and spatial contexts improves the effect on strong real-time data, and is effective for the task of trajectory prediction. That is because the added road network gate is combined with the update gate to integrate the short-term road network characteristics into the model, and the reset gate is combined to integrate the long-term road network characteristics.

In addition, the performance of RNN on the three datasets is better than LSTM. This is because RNN has the characteristics of short-term memory. The closer the time, the greater the weight of track points. Even if the random intervals are added, the obtained samples still have strong real-time performance; so, the performance of RNN is better. Similarly, GRU is superior to LSTM in modeling strong real-time data. The performance of HST-LSTM and STGN is better than the above three models, which proves the importance of spatiotemporal factors to track prediction. Among them, the performance of STGN is better than HST-LSTM, which proves that the method of obtaining spatiotemporal effects through specific gates is more effective than improving on the basis of LSTM gates. The reason may be the increase of the parameters.

In the three datasets, each dataset covers a total of 6 days of trajectory data in an area of Japan, and the time interval between adjacent track points is 5 minutes. Each area can be divided into about 5000 to 10000 regions. Taking the NPF dataset as an example, the number of regions is about 5000. It can be calculated that the size of the spatiotemporal matrix of the dataset is about 5000×3000. However, the number of trajectory points in the NPF dataset is only one million. After removing the repeated spatiotemporal regions, the size of the track point coverage matrix is less than 1% of the size of the spatiotemporal matrix of the dataset. RNN, LSTM, and GRU are directly trained on the spatiotemporal matrix, which will lead to the problems of data sparsity. STGRU adds constraints between track points through time intervals, distance intervals, and road network structure. While the STGRU is being trained, only the local area covered by each sample needs to be considered, which greatly alleviates the problems of data sparsity in the dataset and can better model user behavior patterns compared to the above three models.

4.4.2. Impact of Parameters. In the standard RNN, different cell sizes will lead to different performance. They studied the impact of cell size on STGRU. Observe the impact of different cell sizes on model performance by changing the cell size to 32, 64, 128, 256, and 512. It can be seen from Table 2 that increasing the cell size to a certain extent can improve the performance of the model. Large cell size will increase the training time and result a decline in performance. When the number of model units is determined, the cell size determines the complexity of the model, and a larger cell size may fit the data better.

4.5. Ablation Experiment

4.5.1. Effectiveness of Time and Distance Gates. STGRU has a time gate and a distance gate combined with update gate to capture short-term dependencies. The effectiveness of time and distance gates on modeling time and distance intervals is important. The time gate and distance gate can be closed by set $T_t = 1$ and $D_t = 1$. In order to eliminate the interference of road network gates, the road network gate in STGRU was also closed. There are three sets of experiments, respectively, closing the time gate and the distance gate and closing both two gates at the same time to compare and verify the effectiveness of the time gate and the distance gate.

From Table 3, it can be found that time gate and distance gate have similar importance on the datasets. Compared with GRU, the performance improvement of $GRU + D_t + T_t$ on the four evaluation metrics is 27.67%, 18.04%, 11.22%, and 7.23%, respectively. And the performance difference between $GRU + T_t$ and $GRU + D_t$ is very small, indicating that the distance interval and time interval have similar effects on modeling behavior patterns. And the performance improvement of $GRU + D_t + T_t$ is small, indicating that there is a large degree of overlap in the characteristics of the time interval and the distance interval on the testing dataset.

4.5.2. Effectiveness of Road Network Gates. There is a road network gate in STGRU, which is integrated with the update gate and the reset gate to capture long-term and short-term road network dependencies. The motivation of this group is

Table 4: The performance with different set of road network gates.

NPF	Recall@1	Recall@5	Recall@10	Recall@20
STGRU − R_t	0.0909	0.2801	0.3805	0.5116
STGRU − R_{2t}	0.0867	0.2776	0.3786	0.5101
STGRU − R_{1t}	0.0862	0.2795	0.3796	0.5093
STGRU	0.0920	0.2829	0.3891	0.5231

Table 5: The performance with different window sizes.

Window size	Recall@1	Recall@5	Recall@10	Recall@20
10	0.0933	0.3795	0.5263	0.6730
15	0.0929	0.3897	0.5331	0.6828
20	0.0949	0.3624	0.5174	0.6786
25	0.0905	0.3794	0.5267	0.6831
30	0.1014	0.3611	0.5842	0.6957

to study the role of road network gates in the update gate and reset gate through experiments. The road network gate can be closed by setting $\text{Road}_t = 1$ in $\tilde{R}_t$ and $\tilde{Z}_t$, respectively. The effectiveness of the road network gate in capturing long-term and short-term dependencies can be verified by setting up three sets of experiments, namely, closing all road network gates and closing a single road network gate.

As shown in Table 4, the performance of closing a single road network gate is not as good as closing all road network gates. This may be due to the long-term features and short-term features of the road network structure that need to be used together. The closing of a single road network will cause the road network information to be invalid for the prediction result. At the same time, some parameters are used to model the characteristics of the road network, which will cause the performance of the model to decrease. Therefore, the performance of closing one road network gate alone is almost the same.

4.5.3. Impact of the Sliding Window Size. In our experiment, samples are obtained through a sliding window. The size of the sliding window limits the trajectory length of a single input. In order to compare the performance of our model in different size sliding windows, the sliding windows are set to different lengths to observe the impact, respectively, 10, 15, 20, 25, and 30. In order to ensure the number of samples under a larger sliding window size, it conducts experiments on the Tokyo People Flow dataset, because the dataset has the longest average trajectory length.

The size of the sample length determines the length of the model unit, as well as the parameters of the model. As shown in Table 5, as the length of the sliding window increases and the amount of model parameters increases, the overall performance of the model has a certain improvement. When the sliding window size is set to 30, the model complexity is 3 times that when the sliding window size is 10, the performance improvement of the four metrics increases are 8.68%, -4.85%, 11%, and 3.37%, respectively. Although the increase of the sample length can improve the performance of the model, it is necessary to consider the actually application scenarios of the gather prediction task. The sample length within 10 is more meaningful, and this is also the main reason that the sliding window size is set to 10 in our comparison experiment with the baselines.

Table 6: The performance with different random interval sizes.

Rand interval	Recall@1	Recall@5	Recall@10	Recall@20
3	0.0889	0.3725	0.5148	0.6636
5	0.0933	0.3795	0.5263	0.6730
7	0.0920	0.3835	0.5300	0.6776
9	0.0822	0.3677	0.5196	0.6752
11	0.0747	0.3512	0.5104	0.6702

4.5.4. Impact of the Random Interval Size. Another important parameter is the size of the random interval. Increase the complexity of the trajectory samples by randomly sampling of the time interval between the track points in the sliding window. For comparison, the impact of different random intervals on the complexity of the trajectory sample and the performance of the model sets up different random intervals on the Tokyo People Flow dataset for comparison experiments and sets the random interval sizes to 3, 5, 7, 9, and 11, respectively. Choose the Tokyo People Flow dataset which the continuity in the dataset is strong, and a certain degree of complexity is required to better reflect the purpose of the experiment.

According to Table 6, it can be see that the model performance is the best when the random interval size is 5 and 7, and the random interval size that is too large and too small will cause the model performance to decrease. On the other two datasets, the model performance is better when the random interval size is 3, which is why the three data sets use different random intervals in the comparison experiment with the baselines. Using random intervals can make the sample closer to the data in the real world.

4.6. Case Study. The purpose of verifying the STGRU model is that it can process and predict short trajectory data and long trajectory data, which conducted two sets of experiments with the baseline models. If the user's trajectory data is scarce, it means that it can hardly understand the user's behavior pattern, which requires higher performance of the model. The experiments are based on the Tokyo People Flow dataset, taking data with a track length of less than 30 for calculation, without random intervals, and use recall@k as the evaluation metrics. As shown in Figure 4, STGRU has the best performance on recall@1 and recall@5, which proves that STGRU can better handle sparse data.

In another set of experiments, data with track length greater than 200 was obtained and followed the parameter settings of the comparative experiment. As shown in Figure 5, STGRU is also superior to all baselines on long trajectory data, which proves that STGRU can extract and use long-term features very well, especially the effectiveness of long-term road network features for modeling strong real-time data.

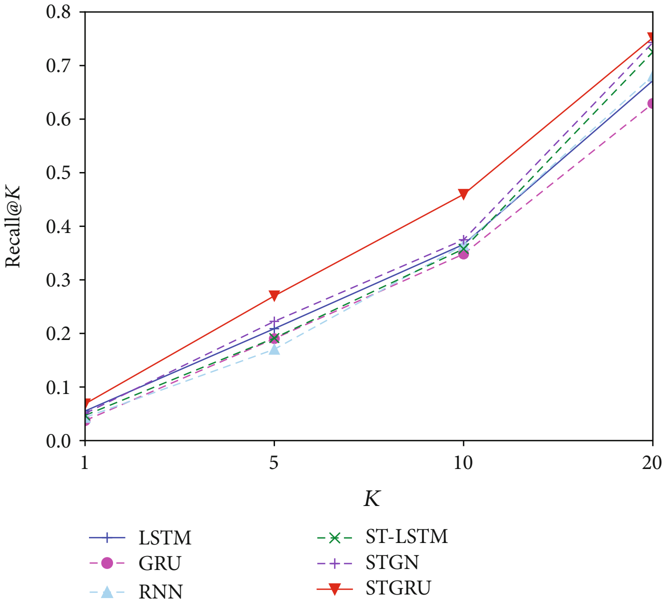


Figure 4: The performance on trajectory data length is less than 30.

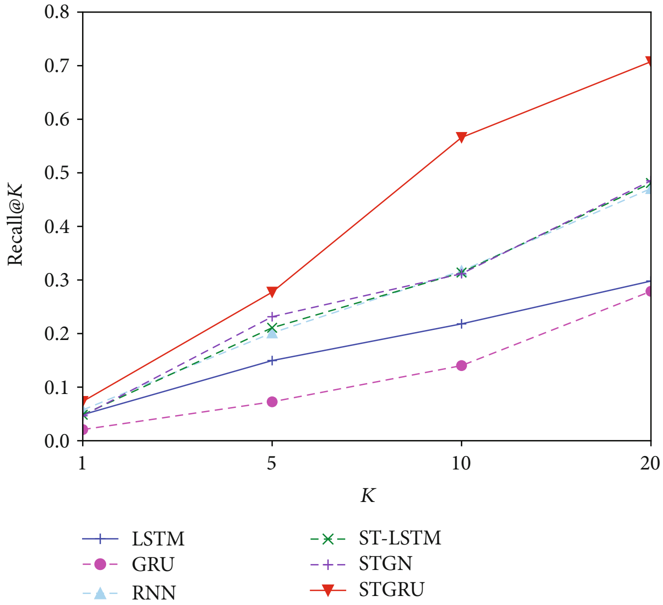


Figure 5: The performance on trajectory data length is greater than 200.

5. Conclusion

In this paper, we propose a Spatio-Temporal Gate Recurrent Unit (STGRU) model by enhancing Gate Recurrent Unit for gather prediction. In STGRU, the time gate and distance gate are introduced to model the time interval and distance interval between consecutive trajectory points, which are essential to describe the short-term behaviors of users, and the road network gate is introduced to model the long-term and short-term road network structure. We believe that the geographical environment represented by the road network structure is very important for both the short-term and long-term behaviors of users. The three gates are combined with the update gate in the GRU to extract the user's short-term behaviors pattern. Only the road net gate and the reset gate in the GRU are combined to extract long-term behaviors patterns of users. Experimental results on three real-world datasets prove the effectiveness of our model, which is better than the latest methods.

In future work, we will further incorporate the structured representation of road network information into the model to further improve the aggregation prediction performance.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (No. 2018YFC0807500), by the National Natural Science Foundation of China under Grant Nos. (U19A2059, 61802050), and by Ministry of Science and Technology of Sichuan Province Program (2018HH0075).

References

[1] J. Pang, Y. Huang, Z. Xie, J. Li, and Z. Cai, "Collaborative city digital twin for the covid-19 pandemic: a federated learning solution," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 759–771, 2021.

[2] M. Sreenivasulu and M. Sridevi, "Comparative study of statistical features to detect the target event during disaster," *Big Data Mining and Analytics*, vol. 3, no. 2, pp. 121–130, 2020.

[3] Z. Cai and Z. He, "Trading private range counting over big iot data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, July 2019.

[4] J. Pang, Y. Huang, Z. Xie, Q. Han, and Z. Cai, "Realizing the Heterogeneity: A Self-Organized Federated Learning Framework for Iot," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3088–3098, 2020.

[5] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart internet of things systems: a consideration from a privacy perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[6] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial iots," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[7] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks: a survey towards private and secure applications," *ACM Computing Surveys (CSUR)*, 2021.

[8] B. Zhao, P. Zhao, and P. Fan, "Epuf: a lightweight double identity verification in iot," *Tsinghua Science and Technology*, vol. 25, no. 5, pp. 625–635, 2020.

[9] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, Makuhari, Chiba, Japan, September 2010.

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

[11] K. Cho, B. van Merrienboer, Ç. Gülçehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014.

[12] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2018.

[13] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: a recurrent model with spatial and temporal contexts," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, Phoenix, AZ, USA, 2016.

[14] D. Kong and F. Wu, "Hst-lstm: a hierarchical spatial-temporal long-short term memory network for location prediction," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 2341–2347, Stockholm, Sweden, 2018.

[15] P. Zhao, A. Luo, Y. Liu et al., "Where to go next: a spatio-temporal gated network for next poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, pp. 5877–5884, 2020.

[16] Q. Hou, M. Han, and Z. Cai, "Survey on data analysis in social media: a practical application aspect," *Big Data Mining and Analytics*, vol. 3, no. 4, pp. 259–279, 2020.

[17] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: data-driven traffic forecasting," in *6th International Conference on Learning Representations, ICLR*, Vancouver, BC, Canada, 2018.

[18] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.

[19] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: a deep learning framework for traffic prediction," *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 5668–5675, 2019.

[20] L. Zhang, N. R. Alharbe, G. Luo, Z. Yao, and Y. Li, "A hybrid forecasting framework based on support vector regression with a modified genetic algorithm and a random forest for traffic flow prediction," *Tsinghua Science and Technology*, vol. 23, no. 4, pp. 479–492, 2018.

[21] X. Geng, Y. Li, L. Wang et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 3656–3663, 2019.

[22] Y. Li, Z. Zhu, D. Kong, M. Xu, and Y. Zhao, "Learning heterogeneous spatial-temporal representation for bike-sharing demand prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 1004–1011, 2019.

[23] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: a reinforcement learning approach for intelligent traffic light control," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2496–2505, London, United Kingdom, 2018.

[24] R. Begleiter, R. El-Yaniv, and G. Yona, "On prediction using variable order markov models," *Journal of Artificial Intelligence Research*, vol. 22, pp. 385–421, 2004.

[25] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[26] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," *Advances in Neural Information Processing Systems*, vol. 20, pp. 1257–1264, 2007.

[27] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with bayesian probabilistic tensor factorization," in *Proceedings of the 2010 SIAM international conference on data mining*, pp. 211–222, Columbus, OH, USA, 2010.

[28] 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.

[29] R. Xin, J. Zhang, and Y. Shao, "Complex network classification with convolutional neural network," *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 447–457, 2020.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France, April 2017http://OpenReview.net.

[31] A. Al-Molegi, M. Jabreel, and B. Ghaleb, "Stf-rnn: space time features-based recurrent neural network for predicting people next location," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7, Athens, Greece, 2016.

[32] L. Zhao, Y. Song, C. Zhang et al., "T-gcn: a temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[33] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3634–3640, Stockholm, Sweden, 2018.

[34] T. Iwata and H. Shimizu, "Neural collective graphical models for estimating spatio-temporal population flow from aggregated data," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3935–3942, 2019.

[35] N. Verma and N. Baliyan, "Pam clustering based taxi hotspot detection for informed driving," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–7, Delhi, India, 2017.

[36] M. Ni, Q. He, and J. Gao, "Forecasting the subway passenger flow under event occurrences with social media," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1623–1632, 2016.

[37] D. Kumar, J. C. Bezdek, S. Rajasegarar, C. Leckie, and M. Palaniswami, "A visual-numeric approach to clustering and anomaly detection for trajectory data," *The Visual Computer*, vol. 33, no. 3, pp. 265–281, 2017.

[38] T. Kurashima, T. Iwata, T. Hoshide, N. Takaya, and K. Fujimura, "Geo topic model: joint modeling of user's activity area and interests for location recommendation," in *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 375–384, Rome Italy, 2013.

[39] Z. Liu, F. Miranda, W. Xiong, J. Yang, Q. Wang, and C. Silva, "Learning geo-contextual embeddings for commuting flow prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 808–816, 2020.

[40] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th International Conference on World Wide Web*, pp. 811–820, North Carolina, USA, 2010.

[41] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: successive point-of-interest recommendation," in *Twenty-Third International Joint Conference on Artificial Intelligence*, Beijing, China, 2013.

[42] S. Feng, X. Li, Y. Zeng, G. Cong, and Y. M. Chee, "Personalized ranking metric embedding for next new poi recommendation," in *IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 2069–2075, Buenos Aires, Argentina, 2015.

[43] Y. Zhu, H. Li, Y. Liao et al., "What to do next: modeling user behaviors by time-lstm," *IJCAI*, vol. 17, pp. 3602–3608, 2017.

[44] C. Yang, M. Sun, W. X. Zhao, Z. Liu, and E. Y. Chang, "A neural network approach to jointly modeling social networks and mobile trajectories," *ACM Transactions on Information Systems (TOIS)*, vol. 35, no. 4, pp. 1–28, 2017.