Wireless Communications and Mobile Computing

Privacy-Preserving Techniques in Deep Learning for Mobile Computing

Lead Guest Editor: Lihua Yin
Guest Editors: Yanhui Guo and Ben Niu

Privacy-Preserving Techniques in Deep Learning for Mobile Computing

Wireless Communications and Mobile Computing

Privacy-Preserving Techniques in Deep Learning for Mobile Computing

Lead Guest Editor: Lihua Yin
Guest Editors: Yanhui Guo and Ben Niu

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2375702, 12 pages
https://doi.org/10.1155/2022/2375702

Research Article

Machine Learning Methods for Intrusive Detection of Wormhole Attack in Mobile Ad Hoc Network (MANET)

Masoud Abdan and Seyed Amin Hosseini Seno

Department of Computer Engineering, Ferdowsi University of Mashhad, Iran

Correspondence should be addressed to Masoud Abdan; abdan@mail.um.ac.ir

Received 28 April 2021; Revised 3 July 2021; Accepted 16 December 2021; Published 31 January 2022

Academic Editor: Lihua Yin

A wormhole attack is a type of attack on the network layer that reflects routing protocols. The classification is performed with several methods of machine learning consisting of K-nearest neighbor (KNN), support vector machine (SVM), decision tree (DT), linear discrimination analysis (LDA), naive Bayes (NB), and convolutional neural network (CNN). Moreover, we used nodes' properties for feature extraction, especially nodes' speed, in the MANET. We have collected 3997 distinct (normal 3781 and malicious 216) samples that comprise normal and malicious nodes. The classification results show that the accuracy of the KNN, SVM, DT, LDA, NB, and CNN methods are 97.1%, 98.2%, 98.9%, 95.2%, 94.7%, and 96.4%, respectively. Based on our findings, the DT method's accuracy is 98.9% and higher than other ways. In the next priority, SVM, KNN, CNN, LDA, and NB indicate high accuracy, respectively.

1. Introduction

A MANET (mobile ad hoc network) is a series of wirelessly interconnected, self-arranged nodes. Each mobile ad hoc network node functions as a router to transmit the packet to the destination node from the source node. Remote ad hoc networks are enormous and commonly used networks. Each movable node is a node that is self-managed, and there is no central mobile network management node. Based on their need, the mobile nodes have permission to go somewhere. It makes it possible for the nodes to join or exit the network [1] quickly. There is no restriction to the capacity of nodes for communication. If the relationship is formed and the nodes are outside the network radio range, data loss can occur. MANET is commonly used in numerous fields, such as science, rescue operations, and military. Cyberattacks are also growing due to improved connectivity across networks [2]. Because of shared channel illumination, unconfident operating environment, restricted resource mobility, rapidly evolving device topology, resource-limited [3], ad hoc wireless mobile networks are susceptible to many security threats.

Detection based on irregularities accepts interference based on a system's everyday actions. The method of enumerating standard system output is demanding because system activity varies from time to time [4]. The anomaly procedure figures out fresh or unexplained attacks with high false positive rates. Signature-based IDS is characterized by searching for unique patterns such as byte sequences in network traffic as an attack detection method [5]. It merely recognizes proven attacks and fails to identify new attacks for which there is no trend. In MANET, safe connectivity is challenging due to the lack of fixed infrastructure, complex topology, etc. Detection of intrusion is a notion that holds up the balance by methods of cryptography and access management. It is displayed to resolve the attack that has happened or is in progress as automatic detection and root of warning. In various IDS such as host intrusion detection systems (HIDS), application-based IDS, and network intrusion detection systems, the notion of ID is stored (NIDS). Since they are passive, the IDS do not take protective action, and they only discover intrusion that triggers an alarm [6]. A wormhole attack is a sort of network layer assault that mimics routing mechanisms. Two or more malicious nodes

detect a wormhole threat using a private channel named the tunnel. The wormhole tunnel would then continue to capture and relay the same data packets to some other location. A malicious node receives a control packet on one side of the tunnel. It transfers through a private channel to another interesting node at the other end and rebroadcasts the packet locally. The path for communication between the source and target is preferred via the private channel due to better prediction, e.g., fewer hops or less time, relative to packets exchanged through other routes [7]. One component that was developed in the late 1950s by artificial intelligence was ML. Over time, it has developed and evolved into algorithms that could be machine-based and efficient enough in medical, engineering, and computer sciences to solve different concerns, such as sorting, clustering, regression, and optimization [8–11] and medical image processing [12–17]. ML architectures learn dynamically without human participation and take action accordingly. It builds a model by automatically, effectively, and correctly manipulating complex data. To have a general approach to improving device performance, ML can benefit from a generalized structure. It has many applications in scientific fields such as manual information entry, automatic spam detection, medical diagnostics, image recognition, data clearing, and noise reduction [9, 18], etc. The latest findings indicate that in WSNs, ML has been implemented to address several problems. Using ML in WSNs increases the efficacy of the system and prevents complex problems, such as reprogramming, manually accessing vast volumes of data, and extracting valuable data from data. In gathering vast quantities of data and producing useful data, ML methods are often beneficial [19, 20]. There are many applications of ML methods for identification and classifications such as unsupervised approach [21–23], power electric usage [24–27], and gas consumption analysis [28–31]. KNN's core idea is to look at your area, suppose the test dataset is comparable to them, and deduce the result. We find k neighbors and predict using KNN. In KNN, no prior experience is required. During the test, k neighbors with the shortest distance will be classified. With a few hyperparameters, it is simple to do. However, the drawbacks are that k should be carefully chosen, that high computing costs will be incurred during runtime if the sample size is enormous, and that correct scaling will be required to ensure that all features are treated equally. KNN differs from other models in that it involves a lot of real-time processing compared to others [31]. Compared to other techniques, naïve Bayes is significantly quicker than KNN due to KNN's real-time execution compared to other methods. SVM also handles outlier's superior to KNN. KNN outperforms SVM when the training data is significantly more significant than the number of features. When there are many characteristics and little training data, SVM beats KNN. The DT algorithm is a tree-based method for solving regression and classification issues. An inverted tree is constructed to generate the result, with branches branching off from a homogeneous probability distributed root node to extremely heterogeneous leaf nodes. The significant benefits are that data does not need to be preprocessed or distributed.

Furthermore, DTs can offer a clear rationale for the prediction. However, when training complex datasets, the tree may become quite complex. DTs are better at dealing with categorical data and colinearity than SVM [8, 31]. The fundamental purpose of this paper is to suggest the technique of detecting a wormhole threat base on machine learning methods. The classification is performed with several ways of machine learning consisting of K-nearest neighbor (KNN), support vector machine (SVM), decision tree (DT), linear discrimination analysis (LDA), naive Bayes (NB), and convolutional neural network (CNN). Moreover, we used nodes' properties for feature extraction, especially nodes' speed, in the MANET. The results are illustrated based on performance criteria in the form of a confusion matrix and ROC curve.

2. Literature Review

Wireless networks are very vulnerable to threats, and the lines of communication are open to hackers. In MANETs, the monitoring of attackers can be accomplished by program modules that track malicious network operations automatically. We ought to consider specific thoughts when developing an intruder identification method for MANETs [32]. For MANETs, the intruder detection systems will act separately from their wired counterparts. When developing intruder detection systems for MANETs, some problems need to be tackled. Unsupervised UOSDA method monitoring systems deploy node-level agents to track and record any unusual activities [33]. In determining the position of agents when the nodes are mobile, the most significant challenge lies. Similarly, the nodes hosting the intruder detection agents require higher bandwidth, battery capacity, and processing power. In MANETs [34], however, these services are restricted. An NP-complete challenge is increasing the attacker detection rate with minimal resources, and multiple writers have suggested algorithms to provide the closest solutions. For MANETS [35], there are many intruder detection architectures available. As in wired networks, many attacks can occur, some of which in MANETs are more destructive. The standard techniques for detecting attack traffic are inadequate due to the features of these networks. Intrusion detection systems (IDSs) are based on various detection techniques, but anomalies' detection is one of the most important. Besides, if these IDSs are centralized, IDSs based on previous attack signatures are less effective. Artin et al. [36] have used a novel machine learning technique that predicts the traffic based on climate condition. A two-level monitoring method for detecting malicious nodes in MANETs is proposed by Amouri et al. Dedicated sniffers operating in promiscuous mode are installed at the first stage. Each sniffer uses a decision tree-based classifier that produces quantities that we apply to every reporting time correctly categorized instances.

In another study, the classified instances were transmitted to the algorithmically operated supernode. It determines the amounts related to the cumulative fluctuation measure of the classified samples obtained for each node being evaluated. The outcome approach has also been extended to

wireless sensor networks and is a feasible IDS scheme for those networks [37]. Abasi et al. presented a novel method for the simulation and modeling of the control system in the power electronics of a 72 pulse [20]. Abasi et al. have designed a new artificial intelligence to solve unit commitment problem in the wind farms' presence [27].

Abd-El-Azim et al. suggested MANET's streamlined fuzzy-based intrusion detection method with an automation mechanism employing an adaptive neurofuzzy inference system to generate a fuzzy system (ANFIS). The next move was to configure the FIS and then use the genetic algorithm (GA) to optimize this initialized framework. The network increased with an average of 36 percent in the existence of only blackhole attacks [38]. Some other methods are fixed-time [39] and finite-time [40] fuzzy method and output-feedback decentralized neural network and fuzzy multiple attribute decision-making [41]. Sharifi et al. have modeled a sensitivity analysis for predicting NOx emotion and compared it with other methods [42]. The intrusion detection device for the jamming attack was suggested by Soni and Sudhakar. The jamming attacker slowly inserted the packets into the network and, depending on the time example, the number of these packets is quickly improved. Its unwelcome flooding actions recognize the IDS as the attacker nodes, and the attacker's infection is detected. The suggested scheme continuously tracked all nodes' actions in the network, and the malicious node's behaviors were different from normal nodes and did not behave like a regular node [28]. Abasi et al. have analyzed a model classification for finding in GUPFC-compensated double-circuit transmission lines [26]. Also, in another research, Nezhadnaeini et al. have applied an optimal allocation of distributed generation using a new search optimizer algorithm in system of unbalanced loads [43]. Abasi et al. have studied a new dynamic and static technique for parallel transmission lines [25]. In the presence of the reputed packet dropping nodes in a MANET network, Sultana et al. analyzed the current IDS output. Whenever the packets obtain more than their handling capacities, the reputed intermediate nodes lose the packets, recognized as intermediate bottleneck nodes. The network simulator, NS-2, measured the efficiency. The findings have shown that the negligence by IDS algorithms of the reputed packet falling nodes is a significant problem and harms network performance [44] (see Table 1).

3. Methods and Materials

3.1. Wormhole Attack. One of MANET's most significant security attacks is the wormhole threat. More MANET routing protocols (DSR), AODV, OLSR, DSDV, etc. can be damaged. A wormhole attack is detected by at least two malicious nodes using a private channel called a tunnel. At this stage, the wormhole tunnel will then start to collect the data packets and pass them to some other location [62]. A malicious node receives a control packet on one side of the tunnel. It transfers to another interesting node via a private channel at the other end, retransmitting the packet locally. The path for communication between source and destination is chosen via the private channel due to improved metrics, such as fewer hops or less time than packets sent over other routes usually. Typically, the assault operates in two steps. The wormhole nodes are interested in several paths in the first step. In the second point, the packets start using these malicious nodes. These nodes can complicate the functionality of the network in a variety of ways [63]. For malicious purposes, wormhole nodes may drop, alter, or send data to an outsider. Different forms of attack may be done through this allow, for example, DOS attack, Eavesdropping, and development. A wormhole attack can cut down the whole routing network in MANET. MANET describes how to run MANET in the wormhole attack in Figure 1.

3.2. Support Vector Machine (SVM). SVM is a supervised technical group of ML that best classifies each observation from a given dataset using a hyperplane. SVM can deal with both linear and nonlinear questions and is more useful in large datasets. To address different problems such as routing [64], localization [65], fault diagnosis [66], congestion control [67], and communication issues [68], SVM is added to WSNs.

3.3. K-Nearest Neighbor (KNN). The most popular example-based approach to solve regression and classification problems is the K-nearest neighbor (KNN). The distance between the sample given and the model being measured is mainly defined by KNN. The different distances are known in KNN, such as the Hamming distance, Euclidean distance, Manhattan distance, and Chebyshev distance function. The missing samples from the featured room are detected by this method, and the measurements are reduced. KNN was introduced in WSN applications by data aggregation and anomaly detection.

3.4. Deep Learning. DL is a type of machine learning that belongs to the ANN family with a multilayer understanding [69]. It has application in some studies such as transport and routing networks [70], health care, such as detection and segmentation [71]. Also, it imitates the human brain's communication and information processing mechanisms and procedures the data for object identification, language translation, speech recognition, and decision making. In WSNs, DL is used to tackle many problems, such as abnormality and fault detection, energy harvesting, data efficiency calculation, and routing [72]. In the design of data safety, classification, and prediction activities, the security applications of deep learning models such as intrusion detection systems (IDS), malware detection, and spam filtering have become important. Based on intelligence, these various activities are structured to construct a paradigm that generally classifies and discriminates between "normal" and "malicious" samples, such as attacks and standard packets. With the exponential growth in deep learning models [73], the sophistication of attack strategy tools is enhanced.

3.5. Naïve Bayesian Learning. Bayesian learning is a mathematical technique that seeks the connection among the data by learning conditional dependency with various statistical approaches. To evaluate posterior likelihoods, Bayesian learning takes previous functions of probability and new knowledge. If $Y1, Y2, Y3 \cdots Yn$ represents a series of inputs and returns a mark θ, the likelihood of $p(\theta)$ must be

Table 1: The summary of researches based on ID detection in MANETs.

Author	Year	Method	Results
Shastri et al. [45]	2016	Hop-based analysis technique	Capable of detecting both hidden and revealed attacks
Mudgal and Gupta. [46]	2016	AODV technique	The approach is that the overhead routing is significantly minimized
Artin et al. [36]	2017	The online CEP learning engine	In MANET, to identify attack traffic in an online way
Sui et al. [37]	2018	Two-level detection scheme	The malicious nodes are isolated from the usual nodes easily and effectively
Abdel-Azim et al. [38]	2018	Adaptive neurofuzzy inference system	Blackhole and grayhole detection in the MANET system. The blackhole attack has a more significant impact on the network than the grayhole attack, based on performance
Jhanjhi et al. [47]	2019	Machine learning	The usage of ML methods in the internet of things proposes a rank and wormhole attack detection system
Cheng et al. [48]	2019	PPVF-RSU-CSP	An effort is made to combine cloud storage with VANET, and a PPVF is proposed for cloud-assisted VANET
Prasad et al. [49]	2019	Machine learning	The accuracy is 93.12% for wormhole attack detection in ad hoc
Jhanjhi et al. [50]	2020	Machine learning	Suggesting a rank and wormhole attack prevention hybrid RPL protocol using machine learning
Wang et al. [51]	2020	FD-WCFRSNPS	The suggested FD-WCFRSNPS is efficient and effective, according on the findings of five different tests
Singh et al. [52]	2020	Artificial neural network	Detecting wormhole attack in wireless sensor network
Srilakshmi et al. [53]	2021	Hybrid reactive search and bat algorithm	To evaluate the lifespan of the node, the attack detection rate and node energy are estimated
Goyal et al. [54]	2021	CDMA-based security	Underwater wireless sensor networks wormhole attack. Compared to current methods, the proposed approach also increases energy efficiency
Wang et al. [55]	2021	Approaches to service selection that is quick and dependable	Our suggested scheme offers higher dependability and a lower time cost than previous alternatives, according to the findings
Ni et al. [56]	2021	Fault detection method relies on TDMA	When compared to the chain-TDMA approach, the suggested methodology reduces resource utilization by 87.95-90.42 percent
Amutha et al. [57]	2021	Clustering techniques	A brief analysis of wireless sensor network clustering focused on three distinct types, as classical, optimization, and machine learning techniques is presented
Jiang et al. [58]	2021	System for adaptive cosite fading channels	The analytical modeling and tests confirm that the theoretical argument is valid and useful
Ahmadi et al. [59]	2021	New KATP network for adaptation	Tests on a variety of typical samples as well as clinical data back up your model's state-of-the-art efficiency
Tami and Lim [60]	2021	Ensemble learning	In terms of their Matthews coefficients, accuracies, false positive rates, and the area under ROC metrics, the value of success among classification algorithms is statistically studied
Chen et al. [61]	2021	RPPTD	A truth-finding scheme that is both robust and privacy preserving
Sultana et al. [44]	2021	Considering bottleneck intermediate node	The findings reveal that the negligence by IDS algorithms of the reputed packet falling nodes is a significant issue and hurts network efficiency

amplified. Bayesian learning approaches have resolved many problems in WSNs, such as routing [74], data location [75], aggregation [76], fault prediction, connectivity, and coverage problems [77].

3.6. Decision Trees (DT). DT is similar to supervised ML algorithms that use arrays of it and then other rules to improve readability [78]. There are two kinds of trees in DT. The leaf node is one, and the decision nodes are another. DT forecasts a class or goal based on the judgment rules and generates a training model derived from training results. Decision trees offer many advantages, such as transparency, less complexity, and rigorous decision-making analysis. Decision trees are used to resolve different WSN problems, including connectivity, data aggregation, and mobile devices.

3.7. Convolutional Neural Network. CNNs are widely utilised for deep learning and the most well-known types of neural networks, mainly in large datasets such as photos and videos. Cortex neurobiology has resulted in a multilayer neural network design. It is made up of both convolutional

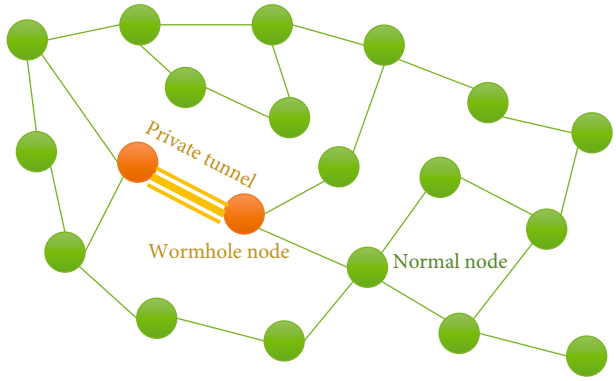

Figure 1: The diagram of the wormhole attack.

and fully linked layers. Subsampling layers can occur between these two levels. They achieve the best of DNNs with complexity in well scaling and multidimensional locally correlated input data. Therefore, the immediate implementation of CNN takes place in dataset where relatively numerous nodes and factors require to be trained.

3.8. Proposed Process. Our method is helpful in the identification of malicious material. This wormhole attack mitigation is introduced in an ad hoc network of natural and malicious output file monitoring nodes. Initially, with their procedures, we describe the sum of normal nodes and malignant nodes. In this scheme, a tunnel between the malicious nodes and the message or packet is established. These are transmitted only over the tunnel. When the malicious node is neighboring to the traditional central node, the message is sent without using the data itself (see Figure 2).

We follow data from each moving node at that stage and accept a message that aids in data collection. The execution of the system can be expanded by specifying the essential role. At that point, to construct a dataset that was marked with the support of an outstanding hub address, we selected eight significant features. Therefore, six standard machine learning classifiers specifically organize ordinary and malicious data from study samples into two categories apply. Device efficiency is measured based on multiple mathematical criteria and compared to the new techniques.

3.9. Performance Analysis of Classification. Accuracy (ACC), precision (P), and sensitivity or recall (R) metrics are used for assessment purposes. Four separate parameters are applied true positive (TP), true negative (TN), false positive (FP), and false negative (FN) to measure these metrics. Accuracy is the proportion, over the volume of data, of the correctly classified number of documents. Precision means the relevant percentage of the performance. On the other hand, recall corresponds to the rate correctly classified by the total functional outcome algorithm. The ratio of the number of abnormal records correctly flagged as an anomaly against the total number of anomaly records is also referred to as detection rate (DR) and true positive rate (TPR). When the total number breaks the anomaly of standard forms, the false positive rate (FPR) is the percentage of the wrongly flagged ordinary record number as follows:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}, \tag{1}$$

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{2}$$

$$\mathrm{DR} = \mathrm{TPR} = R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{3}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{4}$$

4. Results and Discussion

4.1. Simulation of Wormhole Attack. With a finite number of nodes, we have simulated wormhole attacks in the MATLAB 2019b set. It generates a topology consisting of the node, computer, channel, and protocol. Different network programs transfer packets over a network in this simulation process. Packets are either generated or approved and processed, and the simulation model execution reaches the primary role and is processed until the termination state. The original location of nodes and contact nodes against their adjacent nodes is seen in Figure 3.

This simulation was done in an ad hoc network environment with 48 regular nodes and two malicious nodes. Topology room $1000 \times 1000\,\mathrm{m}^2$, spontaneous node activity, and the 250-meter radio range of a node are the simulation environment's experimental parameters (1000 for wormhole nodes). Regarding Figure 3, the normal nodes are indicated with red circles, and wormhole nodes are illustrated with black triangles. Moreover, the initial connection is shown with blue lines between nodes.

4.2. Feature Extraction Results. The selection of features is one of the central principles of machine learning that directly influences its performance. Unrelated or partly related functions may adversely impact the output of the device. The output file includes complete node information in which only any of the data for a given application is informative. Whenever irrelevant or less informative features that do not lead to classification are omitted, it may pick similar features for the dataset. There are many benefits of feature selection, such as decreasing overfitting, reducing training time, and improving accuracy. We have chosen eight essential features that optimize the system's performance. Table 2 includes the characteristics of the MANET presented. Such attributes are either continuous or discrete. We use the specific node address to mark samples and presume that malicious nodes often yield malicious samples.

We have gathered 3997 different samples containing normal and malicious samples (normal 3781 and malicious 216). It builds a dataset that is compiled and tagged with eight chosen attributes. It is a high-volume dataset for wormhole attack detection created in an ad hoc network context.

4.3. Results of Classification. The results of classification with several methods of machine learning consisting of K-nearest neighbor (KNN), support vector machine (SVM), decision tree (DT), linear discrimination analysis (LDA), naive Bayes (NB), and convolutional neural network (CNN) are

Figure 2: Conceptual diagram of the detection process.

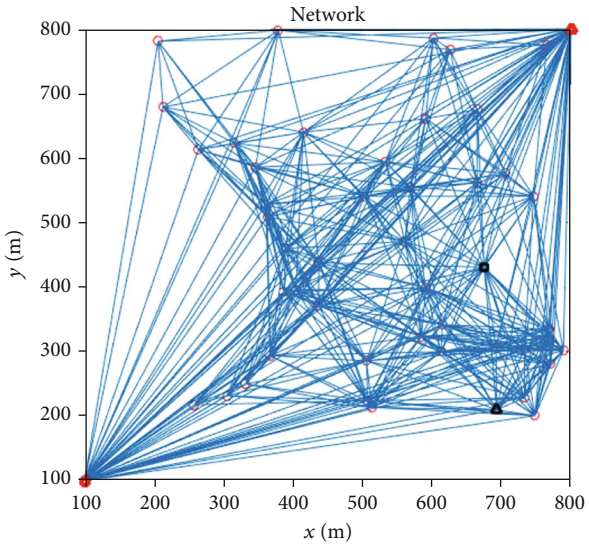

Figure 3: The position of MANET nodes.

Table 2: The selected feature for diagnosis of wormhole attacks in MANET.

No.	Features
1	Number of nodes
2	Maximum speed
3	Minimum speed
4	Average speed
5	The standard deviation of speed
6	Faster's direction
7	Distance to the destination
8	Sum of distances

illustrated in Figure 4. Regarding the confusion matrix of Figure 4, the green arrays show the true values, and red elements indicate false ones. For binary evaluation, the target class is usually considered a positive class. For this paper, our main objective is to find wormhole nodes between normal nodes. Therefore, the class of wormhole is regarded as a positive class. Base on the confusing matrix of Figure 4 from true values, the upper cell shows the true negative, and the lower one is true positive. Respectively, from red cells, the upper one is false negative, and the lower one is false positive class. The classification is performed based on two classes, including normal and malicious nodes.

Vertical gray cells represent accuracy and negative predictive values, while horizontal gray cells represent sensitivity and specificity. For example, in SVM results, from 216 wormhole nodes, 158 (73.1%) are diagnosed correctly. However, 58 (26.9%) are misdiagnosed as normal nodes. In other words, the sensitivity of the SVM method is 73.1%. On the other hand, the SVM method can diagnose the normal node with 99.6% specificity. It means that from 3781 normal nodes, only 15 (0.4%) are misdiagnosed. Moreover, in the DT classifier, 87.7% (precision) are in a true state from all detected wormhole nodes. On the other hand, the precision of the DT classifier is 87.7%. The total accuracy value that comprises DT is the value in the confusion matrix's lower-right corner cell. This value equals 98.9%. To conclude, the results show that the accuracy of the KNN, SVM, DT, LDA, NB, and CNN methods are 97.1%, 98.2%, 98.9%, 95.2%, 94.7%, and 96.4%, respectively. Furthermore, the classifier's overall error value is displayed in red writing in the lower-right corner. We calculated that DT outperforms all other classical classifiers in terms of accuracy.

Table 3 shows the deep convolutional neural network that was employed in this work. For every 3997 nodes in each layer, there are 8 features. As a result, the input matrix is 8×1. We also employed two convolutional layers with ten filters of 2×2 size and stride [1] with zero paddings, as well as two convolutions with ten filters of 2×2 size and stride [1]. We also utilised the Tanh and ReLU routines to activate the layers. Then, with 384 and 2 cells, respectively, two completely linked layers are employed. The SoftMax layer is then used to calculate likelihood and activate the final levels. The classification layer is then utilised, which is based on cross-entropy and takes mutually exclusive classifications into account. The categorization procedure' outcomes are depicted in Figure 5. The procedure is carried out on a core i7 Intel processor with a clock speed of 3 GHz and 12 GB of RAM. The training procedure is repeated 3000 times. Figure 5 shows the accuracy and loss rate of the training procedure for a deeper understanding of machine learning techniques, and Figure 6 shows the ROC curve based on classifier. In the ROC curve, the horizontal axis represents the false positive rate, while the vertical axis represents the true positive rate. To put it another way, the graph is shown with wormhole nodes as the positive class. The area under the curve of the ROC curve, often known as AUC, is an important criterion for classifier performance assessment.

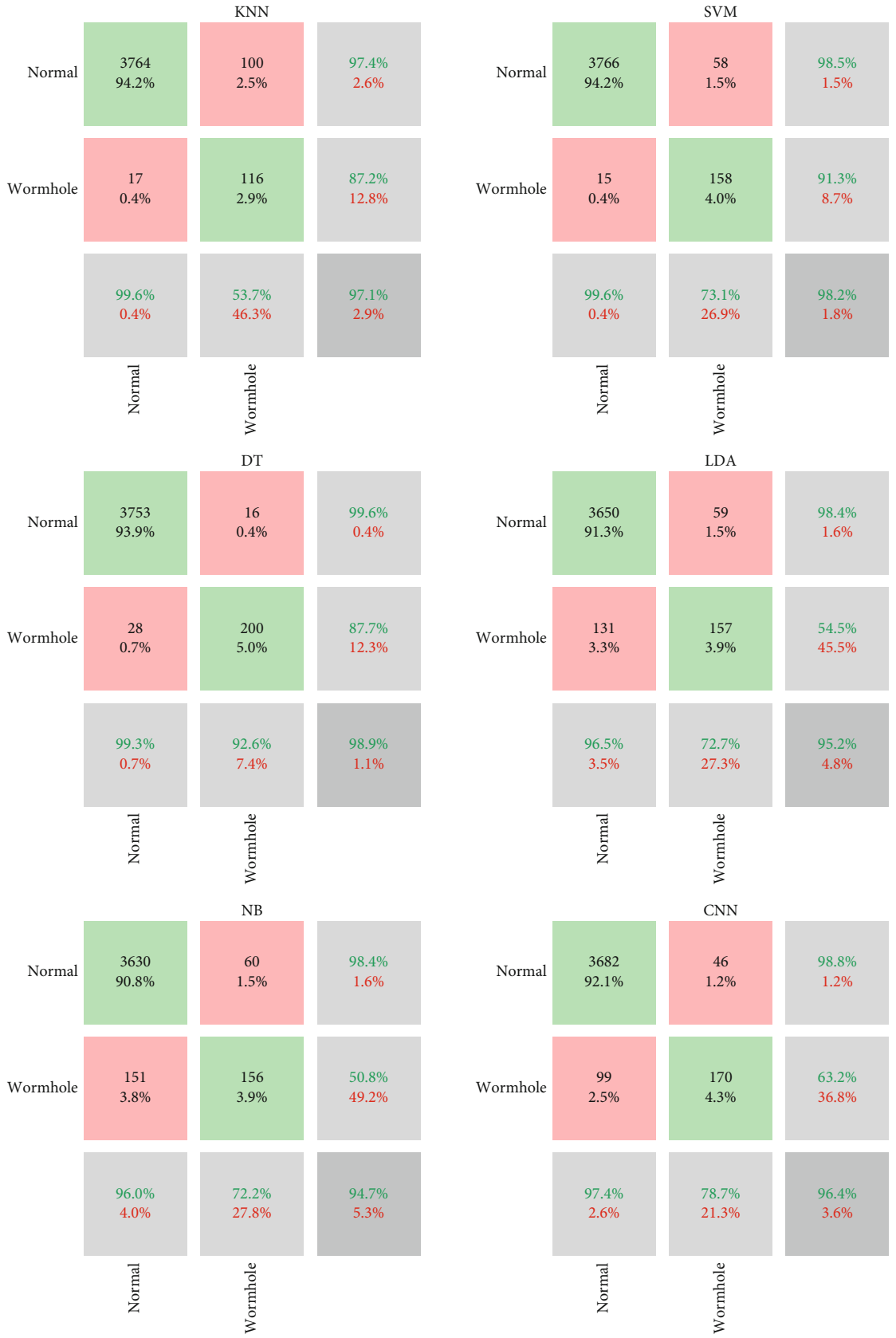

Figure 4: The confusion matrix of the utilized machine learning methods.

Table 3: The architecture of the presented CNN method.

No	Layer	Properties
1	Input feature	$8 \times 1 \times 1$ matrix
2	Convolution layer	10 (2×2) convolutions, stride [1]
	Tanh	
2	Convolution layer	10 (2×2) convolutions, stride [1]
3	ReLU	$F(x) = \max(0, x)$
4	Fully connected	384 fully connected layers
6	Fully connected	Two fully connected layers
7	SoftMax	$\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}, i = 1, \cdots, K \quad x = (x_1, \cdots, x_K)$
8	Classification output	For multiclass classifier with class labels, the cross-entropy loss is used

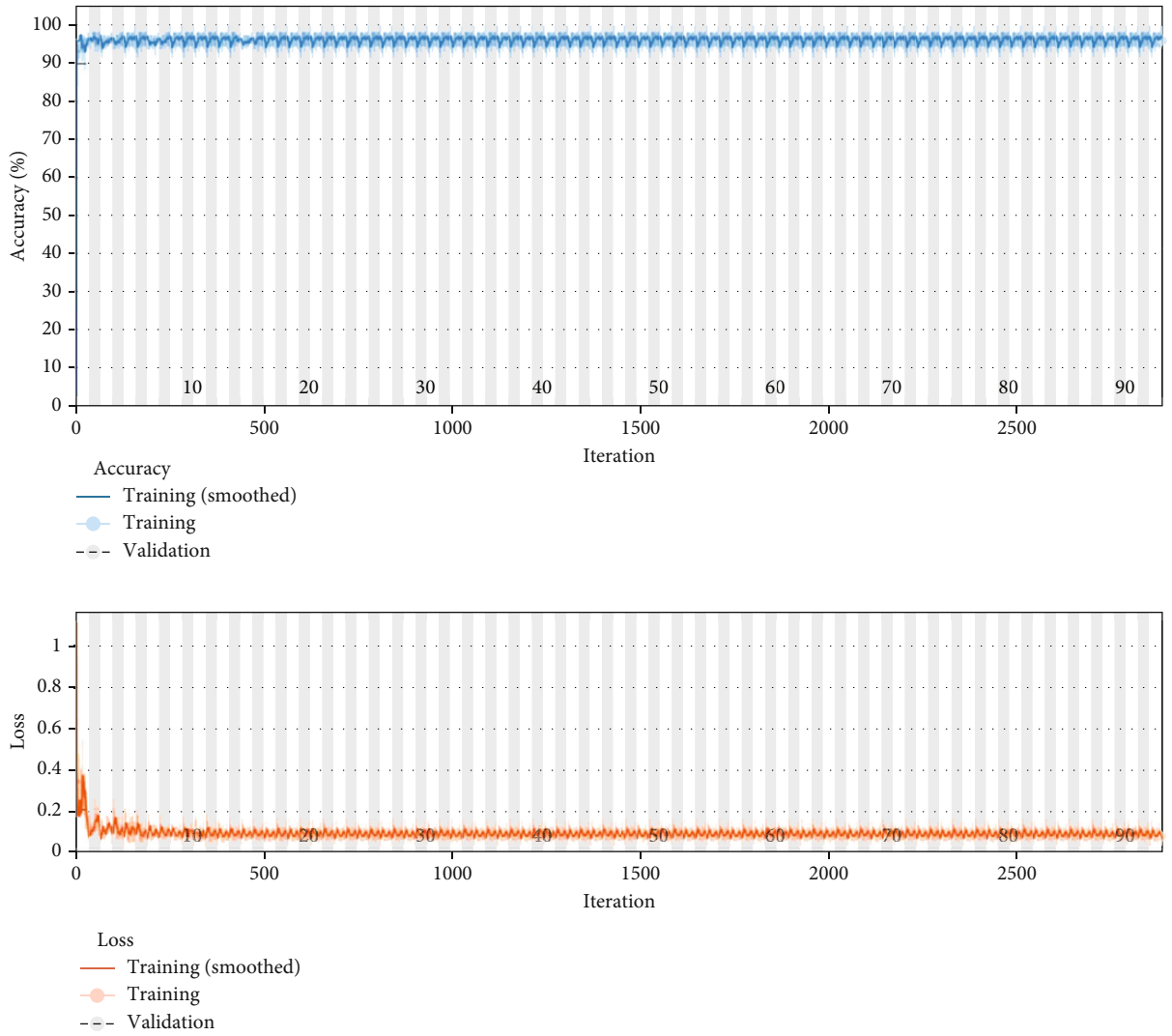

Figure 5: The accuracy and the loss value for CNN architecture.

The DT classifier has a higher AUC than the other approaches, as can be observed.

Table 4 shows the results of the evaluation of machine learning approaches. The sensitivity of the DT technique exceeds other methods, according to the findings. The sensitivity refers to the method's ability to detect wormhole nodes in MANET. As a result, the size of it signified the classifiers' capability. In other words, the DT classifier has a higher sensitivity than other approaches. The accuracy also reveals the method's capability for producing outcomes or its

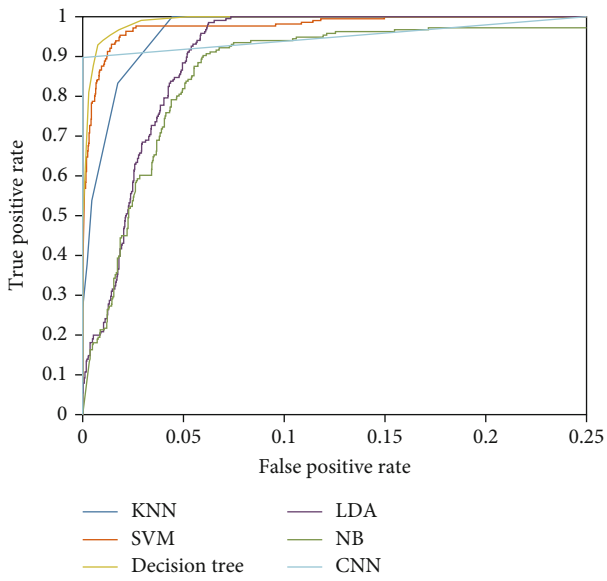

Figure 6: The ROC curve of different classifiers used for wormhole detection.

Table 4: The comparison of methods of diagnosis employed in this article.

	KNN	SVM	DT	LDA	NB	CNN
Sensitivity	53.7%	73.1%	92.6%	72.7%	72.2%	78.7%
Specificity	99.6%	99.6%	99.3%	96.5%	96.0%	97.4%
Precision	87.2%	91.3%	87.7%	54.5%	50.8%	63.2%
AUC	99.1%	99.4%	99.74%	97.5%	95.9%	96.3%
Accuracy	97.1%	98.2%	98.9%	95.2%	94.7%	96.4%

dependability. The SVM approach, for example, has a precision of 91.3 percent. It means that, from all nodes that the SVM recognized as wormhole nodes, 91.3% are the positive test of the real wormhole. The specificity also shows that how the classifier detects the normal node. The higher specificity is belonging to KNN and SVM approaches. Finally, the higher AUC value has resulted from the DT method. To summarise the findings, the DT approach has a 98.9% accuracy rate, which is greater than other methods. SVM, KNN, CNN, LDA, and NB, in order of importance, indicate excellent accuracy.

5. Conclusion

A wormhole attack is a type of attack on the network layer that reflects routing protocols. To detect wormhole attacks using machine learning, a training dataset must train models in any training mode. Training datasets can be obtained from real-time conditions or tests for classification. As a function, the experimental data may be defined as a target value and a descriptive process. This article has obtained 3997 different samples containing normal and malicious samples (normal 3781 and malicious 216). It builds a dataset compiled with eight selected features and labeled. The classification is performed with several methods of machine learning consisting of K-nearest neighbor (KNN), support vector machine (SVM), decision tree (DT), linear discrimination analysis (LDA), naive Bayes (NB), and convolutional neural network (CNN). To conclude, the results show that the accuracy of the KNN, SVM, DT, LDA, NB, and CNN methods are 97.1%, 98.2%, 98.9%, 95.2%, 94.7%, and 96.4%, respectively. Based on the results, the sensitivity of the DT method outperforms other approaches. The higher specificity is belonging to KNN and SVM approaches. Finally, the higher AUC value has resulted from the DT method. To conclude the results, the DT method's accuracy is 98.9% and higher than other methods. In the next priority, SVM, KNN, CNN, LDA, and NB indicate high accuracy, respectively. Our strategy's success encourages us to expand this work to address the limitations and simulation described in a 3D ad hoc network. In the future, authors should extend some methods to diagnosed different types of attacks related to WSN and IoT systems based on artificial intelligence and machine learning method.

Data Availability

We have simulated wormhole attacks data in the MATLAB 2019b set with a finite number of nodes, and it generates a network topology consisting of the protocol of the node, computer, channel, and network.

Disclosure

The funding sources had no involvement in the study design, collection, analysis, or interpretation of data, writing of the manuscript, or submitting the manuscript for publication.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] J. Su and H. Liu, "Protecting flow design for DoS attack and defense at the MAC layer in mobile ad hoc network," in *International Conference on Applied Informatics and Communication*, pp. 233–240, Springer, Berlin, Heidelberg, 2011.

[2] M. Chitkara and M. W. Ahmad, "Review on MANET: characteristics, challenges, imperatives, and routing protocols," *International journal of computer science and mobile computing.*, vol. 3, no. 2, pp. 432–437, 2014.

[3] M. Sookhak, H. Tang, Y. He, and F. R. Yu, "Security and privacy of smart cities: a survey, research issues and challenges," *IEEE Communications Surveys & Tutorials.*, vol. 21, no. 2, pp. 1718–1743, 2019.

[4] B. Mandal, S. Sarkar, S. Bhattacharya, U. Dasgupta, P. Ghosg, and D. Sanki, "A review on cooperative bait based intrusion detection in MANET," SSRN 3515151.2020.

[5] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[6] O. Can, M. O. Unalir, E. Sezer, O. Bursa, and B. Erdogdu, "An ontology-based approach for host intrusion detection systems," in *research Conference on Metadata and Semantics Research*, pp. 80–86, Springer, Cham, 2017.

[7] A. Varmaghani, A. Matin Nazar, M. Ahmadi, A. Sharifi, S. Jafarzadeh Ghoushchi, and Y. Pourasad, "DMTC: optimize energy consumption in dynamic wireless sensor network based on fog computing and fuzzy multiple attribute decision-making," *Wireless Communications and Mobile Computing.*, vol. 2021, pp. 1–14, 2021.

[8] W. Qiao, M. Khishe, and S. Ravakhah, "Underwater targets classification using local wavelet acoustic pattern and multilayer perceptron neural network optimized by modified whale optimization algorithm," *Ocean Engineering*, vol. 219, article 108415, 2021.

[9] A. Ala, F. E. Alsaadi, M. Ahmadi, and S. Mirjalili, "Optimization of an appointment scheduling problem for healthcare systems based on the quality of fairness service using whale optimization algorithm and NSGA-II," *Scientific Reports*, vol. 11, no. 1, pp. 1–19, 2021.

[10] B. Alizadeh, D. Li, Z. Zhang, and A. H. Behzadan, "Feasibility study of urban flood mapping using traffic signs for route optimization," 2021, https://arxiv.org/abs/2109.11712.

[11] M. F. Nezhadnaeini, M. Hajivand, M. Abasi, and S. Mohajerami, "Optimal Allocation of Distributed Generation Units Based on Different Objectives by a Novel Version Group Search Optimizer Algorithm in Unbalance Load System," *Revue roumaine des sciences techniques Série Électrotechnique et Énergétique*, vol. 61, no. 4, pp. 338–342, 2016.

[12] S. Hassantabar, M. Ahmadi, and A. Sharifi, "Diagnosis and detection of infected tissue of COVID-19 patients based on lung X-ray image using convolutional neural network approaches," *Chaos, Solitons & Fractals.*, vol. 140, no. 140, article 110170, 2020.

[13] M. Ahmadi, A. Sharifi, S. Hassantabar, and S. Enayati, "QAIS-DSNN: tumor area segmentation of MRI image with optimized quantum matched-filter technique and deep spiking neural network," *BioMed Research International.*, vol. 2021, pp. 1–16, 2021.

[14] M. Ahmadi, A. Sharifi, M. Jafarian Fard, and N. Soleimani, "Detection of brain lesion location in MRI images using convolutional neural network and robust PCA," *International Journal of Neuroscience.*, vol. 4, 2021.

[15] M. Rezaei, F. Farahanipad, A. Dillhoff, R. Elmasri, and V. Athitsos, "Weakly-supervised hand part segmentation from depth images," in *In The 14th PErvasive Technologies Related to Assistive Environments Conference*, pp. 218–225, 2021.

[16] F. Farahanipad, M. Rezaei, A. Dillhoff, F. Kamangar, and V. Athitsos, "A pipeline for hand 2-D keypoint localization using unpaired image to image translation," in *In The 14th PErvasive Technologies Related to Assistive Environments Conference*, pp. 226–233, 2021.

[17] B. Alizadeh Kharazi and A. H. Behzadan, "Flood depth mapping in street photos with image processing and deep neural networks," *Computers, Environment and Urban Systems*, vol. 88, article 101628, 2021.

[18] M. Ahmadi, F. Dashti Ahangar, N. Astaraki, M. Abbasi, and B. Babaei, "FWNNet: presentation of a new classifier of brain tumor diagnosis based on fuzzy logic and the wavelet-based neural network using machine-learning methods," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 8542637, 13 pages, 2021.

[19] J. Wang, Y. Gao, X. Yin, F. Li, and H. J. Kim, "An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks," *Wireless Communications and Mobile Computing.*, vol. 2018, article 9472075, pp. 1–9, 2018.

[20] M. Abasi, M. Joorabian, A. Saffarian, and S. G. Seifossadat, "Accurate simulation and modeling of the control system and the power electronics of a 72-pulse VSC-based generalized unified power flow controller (GUPFC)," *Electrical Engineering*, vol. 102, no. 3, pp. 1795–1819, 2020.

[21] F. Liu, G. Zhang, and L. Jie, "Heterogeneous domain adaptation: an unsupervised approach," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 12, pp. 5588–5602, 2020.

[22] W. Qiao, Y. Wang, J. Zhang, W. Tian, Y. Tian, and Q. Yang, "An innovative coupled model in view of wavelet transform for predicting short- term PM10 concentration," *Journal of Environmental Management*, vol. 289, article 112438, 2021.

[23] S. Peng, Q. Chen, and E. Liu, "The role of computational fluid dynamics tools on investigation of pathogen transmission: Prevention and control," *Science of The Total Environment*, vol. 746, p. 142090, 2020.

[24] B. Li, G. Xiao, L. Rongxing, R. Deng, and H. Bao, "On feasibility and limitations of detecting false data injection attacks on power grid state estimation using D-FACTS devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 854–864, 2020.

[25] M. Abasi, M. Joorabian, A. Saffarian, and S. G. Seifossadat, "A novel complete dynamic and static model of 48-pulse VSC-based GUPFC for parallel transmission lines," *International Journal of Industrial Electronics, Control and Optimization*, vol. 3, no. 4, pp. 447–457, 2020.

[26] M. Abasi, A. Saffarian, M. Joorabian, and S. G. Seifossadat, "Fault classification and fault area detection in GUPFC-compensated double- circuit transmission lines based on the analysis of active and reactive powers measured by PMUs," *Measurement*, vol. 169, no. 2, p. 108499, 2021.

[27] M. Abasi, M. F. Nezhadnaeini, M. Karimi, and N. Yousefi, "A novel meta heuristic approach to solve unit commitment problem in the presence of wind farms," *Revue roumaine des sciences techniques Série Électrotechnique et Énergétique*, vol. 60, no. 3, pp. 253–262, 2015.

[28] W. Qiao, W. Liu, and E. Liu, "A combination model based on wavelet transform for predicting the difference between monthly natural gas production and consumption of U.S.," *Energy*, vol. 235, p. 121216, 2021.

[29] E. Liu, D. Li, W. Li et al., "Erosion simulation and improvement scheme of separator blowdown system ——a case study of Changning national shale gas demonstration area," *Journal of Natural Gas Science and Engineering*, vol. 88, p. 103856, 2021.

[30] S. Peng, Y. Zhang, W. Zhao, and E. Liu, "Analysis of the influence of rectifier blockage on the metering performance during shale gas extraction," *Energy & Fuels*, vol. 35, no. 3, pp. 2134–2143, 2021.

[31] S. Peng, R. Chen, B. Yu, M. Xiang, X. Lin, and E. Liu, "Daily natural gas load forecasting based on the combination of long short term memory, local mean decomposition, and wavelet threshold denoising algorithm," *Journal of Natural Gas Science and Engineering*, vol. 104175, 2021.

[32] L. A. Maglaras, "A novel distributed intrusion detection system for vehicular ad hoc networks," *International Journal of Advanced Computer Science and Applications.*, vol. 6, no. 4, pp. 101–106, 2015.

[33] L. Zhong, Z. Fang, F. Liu, B. Yuan, G. Zhang, and L. Jie, "Bridging the theoretical bound and deep algorithms for open set domain adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, pp. 1–15, 2021.

[34] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE communications surveys & tutorials.*, vol. 16, no. 1, pp. 266–282, 2014.

[35] P. Gandotra, R. K. Jha, and S. Jain, "A survey on device-to-device (D2D) communication: architecture and security issues," *Journal of Network and Computer Applications.*, vol. 78, no. 78, pp. 9–29, 2017.

[36] J. Artin, A. Valizadeh, M. Ahmadi, S. A. P. Kumar, and A. Sharifi, "Presentation of a novel method for prediction of traffic with climate condition based on ensemble learning of neural architecture search (NAS) and linear regression," *Complexity*, vol. 2021, 13 pages, 2021.

[37] T. Sui, D. Marelli, X. Sun, and F. Minyue, "Multi-sensor state estimation over lossy channels using coded measurements," *Automatica*, vol. 111, article 108561, 2020.

[38] M. Abdel-Azim, H. E. Salah, and M. E. Eissa, "IDS against black-hole attack for MANET," *IJ Network Security.*, vol. 20, no. 3, pp. 585–592, 2018.

[39] M. Chen, H. Wang, and X. Liu, "Adaptive fuzzy practical fixed-time tracking control of nonlinear systems," *IEEE Transactions on Fuzzy Systems.*, vol. 29, no. 3, pp. 664–673, 2021.

[40] H. Wang, W. Bai, X. Zhao, and P. X. Liu, "Finite-time-prescribed performance-based adaptive fuzzy control for strict-feedback nonlinear systems with dynamic uncertainty and actuator faults," *IEEE transactions on Cybernetics.*, pp. 1–13, 2021.

[41] A. Varmaghani, A. Matin Nazar, M. Ahmadi, A. Sharifi, S. Jafarzadeh Ghoushchi, and Y. Pourasad, "DMTC: optimize energy consumption in dynamic wireless sensor network based on fog computing and fuzzy multiple attribute decision-making," in *Wireless Communications and Mobile Computing*, I. Ali, Ed., 2021.

[42] A. Sharifi, M. Ahmadi, H. Badfar, and M. Hosseini, "Modeling and sensitivity analysis of NOx emissions and mechanical efficiency for diesel engine," *Environmental Science and Pollution Research*, vol. 26, no. 24, pp. 25190–25207, 2019.

[43] M. F. Nezhadnaeini, M. Hajivand, M. Abasi, and S. Mohajerami, "Optimal allocation of distributed generation units based on different objectives by a novel version group search optimizer algorithm in unbalance load system," *Revue roumaine des sciences techniques Série Électrotechnique et Énergétique*, vol. 61, no. 4, pp. 338–342, 2016.

[44] T. Sultana, A. A. Mohammad, and N. Gupta, "Importance of the considering bottleneck intermediate node during the intrusion detection in MANET," in *Research in Intelligent and Computing in Engineering 2021*pp. 205–213, Springer, Singapore.

[45] A. Shastri and J. Joshi, "A wormhole attack in mobile ad-hoc network: detection and prevention," in *In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, pp. 1–4, 2016.

[46] R. Mudgal and R. Gupta, "An efficient approach for wormhole detection in manet," in *In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, pp. 1–6, 2016.

[47] N. Z. Jhanjhi, S. N. Brohi, and N. A. Malik, "Proposing a rank and wormhole attack detection framework using machine learning," in *In 2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, pp. 1–9, IEEE, 2019.

[48] H. Cheng, M. Shojafar, M. Alazab, R. Tafazolli, and Y. Liu, "PPVF: privacy-preserving protocol for vehicle feedback in cloud-assisted VANET," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021.

[49] M. Prasad, S. Tripathi, and K. Dahal, "Wormhole attack detection in ad hoc network using machine learning technique," in *In 2019 10th international conference on computing, communication and networking technologies (ICCCNT)*, pp. 1–7, IEEE, 2019.

[50] N. Z. Jhanjhi, S. N. Brohi, N. A. Malik, and M. Humayun, "Proposing a hybrid RPL protocol for rank and wormhole attack mitigation using machine learning," in *In 2020 2nd international conference on computer and information sciences (ICCIS)*, pp. 1–6, IEEE, 2020.

[51] T. Wang, X. Wei, J. Wang et al., "A weighted corrective fuzzy reasoning spiking neural P system for fault diagnosis in power systems with variable topologies," *Engineering Applications of Artificial Intelligence*, vol. 92, p. 103680, 2020.

[52] M. M. Singh, N. Dutta, T. R. Singh, and U. Nandi, "A technique to detect wormhole attack in wireless sensor network using artificial neural network," in *Evolutionary Computing and Mobile Sustainable Networks*, pp. 297–307, Springer, Singapore, 2020.

[53] R. Srilakshmi and J. Muthukuru, "Intrusion detection in mobile ad-hoc network using hybrid reactive search and bat algorithm," *International Journal of Intelligent Unmanned Systems.*, vol. ahead-of-print, no. ahead-of-print, 2021.

[54] N. Goyal, J. K. Sandhu, and L. Verma, "CDMA-based security against wormhole attack in underwater wireless sensor networks," in *Advances in Communication and Computational Technology*, pp. 829–835, Springer, Singapore, 2021.

[55] S. Wang, A. Zhou, M. Yang, L. Sun, C.-H. Hsu, and F. Yang, "Service composition in cyber-physical-social systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, pp. 82–91, 2020.

[56] T. Ni, D. Liu, Q. Xu, Z. Huang, H. Liang, and A. Yan, "Architecture of cobweb-based redundant TSV for clustered faults," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 28, no. 7, pp. 1736–1739, 2020.

[57] J. Amutha, S. Sharma, and S. K. Sharma, "Strategies based on various aspects of clustering in wireless sensor networks using classical, optimization and machine learning techniques: review, taxonomy, research findings, challenges and future directions," *Computer Science Review.*, vol. 40, no. 40, article 100376, 2021.

[58] Y. Jiang and X. Li, "Broadband cancellation method in an adaptive co-site interference cancellation system," *international journal of electronics just-accepted*, pp. 1–21, 2021.

[59] M. Ahmadi, A. Taghavirashidizadeh, D. Javaheri, A. Masoumian, S. J. Ghoushchi, and Y. Pourasad, "DQRE-SCnet: A novel hybrid approach for selecting users in federated learning with deep-q-reinforcement learning based on spectral clustering," *Journal of King Saud University-Computer and Information Sciences*, 2021.

[60] B. A. Tama and S. Lim, "Ensemble learning for intrusion detection systems: a systematic mapping study and cross-

benchmark evaluation," *Computer Science Review*, vol. 39, no. 39, article 100357, 2021.

[61] J. Chen, Y. Liu, Y. Xiang, and K. Sood, "RPPTD: robust privacy-preserving truth discovery scheme," *IEEE Systems Journal*, pp. 1–8, 2021.

[62] M. Boulaiche, "Survey of secure routing protocols for wireless ad hoc networks," *Wireless Personal Communications.*, vol. 114, no. 1, pp. 483–517, 2020.

[63] A. Kadam, N. Patel, and V. Gaikwad, "Detection and prevention of wormhole attack in MANET," *International Research Journal of Engineering and Technology (IRJET) e-ISSN*, 2016.

[64] F. Khan, S. Memon, and S. H. Jokhio, "Support vector machine based energy aware routing in wireless sensor networks," in *In 2016 2nd international conference on robotics and artificial intelligence (ICRAI)*, pp. 1–4, IEEE, 2016.

[65] J. Kang, Y. J. Park, J. Lee, S. H. Wang, and D. S. Eom, "Novel leakage detection by ensemble CNN-SVM and graph-based localization in water distribution systems," *IEEE Transactions on Industrial Electronics.*, vol. 65, no. 5, pp. 4279–4289, 2018.

[66] T. Wang, W. Liu, J. Zhao, X. Guo, and V. Terzija, "A rough set-based bio-inspired fault diagnosis method for electrical substations," *International Journal of Electrical Power & Energy Systems*, vol. 119, article 105961, 2020.

[67] M. Gholipour, A. T. Haghighat, and M. R. Meybodi, "Hop-by-hop congestion avoidance in wireless sensor networks based on genetic support vector machine," *Neurocomputing*, vol. 223, no. 223, pp. 63–76, 2017.

[68] W. Kim, M. S. Stanković, K. H. Johansson, and H. J. Kim, "A distributed support vector machine learning over wireless sensor networks," *IEEE transactions on cybernetics.*, vol. 45, no. 11, pp. 2599–2611, 2015.

[69] Y. Zhao, J. Jiao, N. Li, and Z. Deng, "MANet: multimodal attention network based point-view fusion for 3D shape recognition," in *In 2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 134–141, IEEE, 2021.

[70] H. Bangui and B. Buhnova, "Recent advances in machine-learning driven intrusion detection in transportation: survey," *Procedia Computer Science*, vol. 184, pp. 877–886, 2021.

[71] A. Sharifi, M. Ahmadi, M. A. Mehni, S. Jafarzadeh Ghoushchi, and Y. Pourasad, "Experimental and numerical diagnosis of fatigue foot using convolutional neural network," *Computer Methods in Biomechanics and Biomedical Engineering.*, vol. 24, no. 16, pp. 1828–1840, 2021.

[72] S. Laqtib, K. El Yassini, and M. L. Hasnaoui, "A technical review and comparative analysis of machine learning techniques for intrusion detection systems in MANET," *International Journal of Electrical and Computer Engineering.*, vol. 10, no. 3, p. 2701, 2020.

[73] V. Jafarizadeh, A. Keshavarzi, and T. Derikvand, "Efficient cluster head selection using naïve Bayes classifier for wireless sensor networks," *Wireless Networks.*, vol. 23, no. 3, pp. 779–785, 2017.

[74] Z. Wang, H. Liu, S. Xu, X. Bu, and J. An, "Bayesian device-free localization and tracking in a binary RF sensor network," *Sensors*, vol. 17, no. 5, p. 969, 2017.

[75] A. De Paola, P. Ferraro, S. Gaglio, G. L. Re, and S. K. Das, "An adaptive Bayesian system for context-aware data fusion in smart environments," *IEEE Transactions on Mobile Computing.*, vol. 16, no. 6, pp. 1502–1515, 2017.

[76] B. Yang, Y. Lei, and B. Yan, "Distributed multi-human location algorithm using naive Bayes classifier for a binary pyroelectric infrared sensor tracking system," *IEEE Sensors journal.*, vol. 16, no. 1, pp. 216–223, 2016.

[77] J. Shu, S. Liu, L. Liu, L. Zhan, and G. Hu, "Research on link quality estimation mechanism for wireless sensor networks based on support vector machine," *Chinese Journal of Electronics.*, vol. 26, no. 2, pp. 377–384, 2017.

[78] M. Ahmadi and M. Q. H. Abadi, "A review of using object-orientation properties of C++ for designing expert system in strategic planning," *Computer Science Review*, vol. 37, article 100282, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8745897, 13 pages
https://doi.org/10.1155/2021/8745897

Review Article

Key Research Issues and Related Technologies in Crowdsourcing Data Collection

Yunhui Li,[1] **Liang Chang**,[2] **Long Li**,[2] **Xuguang Bao**,[2] **and Tianlong Gu**[3]

[1]*School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China*
[2]*Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China*
[3]*College of Information Science and Technology, Jinan University, Guangzhou 510000, China*

Correspondence should be addressed to Tianlong Gu; gutianlong@jnu.edu.cn

Received 8 June 2021; Accepted 28 September 2021; Published 16 October 2021

Academic Editor: Lihua Yin

Crowdsourcing provides a distributed method to solve the tasks that are difficult to complete using computers and require the wisdom of human beings. Due to its fast and inexpensive nature, crowdsourcing is widely used to collect metadata and data annotation in many fields, such as information retrieval, machine learning, recommendation system, and natural language processing. Crowdsourcing helps enable the collection of rich and large-scale data, which promotes the development of researches driven by data. In recent years, a large amount of effort has been spent on crowdsourcing in data collection, to address the challenges, including quality control, cost control, efficiency, and privacy protection. In this paper, we introduce the concept and workflow of crowdsourcing data collection. Furthermore, we review the key research topics and related technologies in its workflow, including task design, task-worker matching, response aggregation, incentive mechanism, and privacy protection. Then, the limitations of the existing work are discussed, and the future development directions are identified.

1. Introduction

Machine learning and deep learning technologies have increasingly become a research topic in many fields, including computer vision, natural language processing, and other fields related to artificial intelligence. The study of these techniques requires large-scale, high-quality data (raw and/or labeled data) to train algorithms, and the quantity and quality of the data directly affect the performance of the trained algorithm. How to collect large-scale, high-quality data is an urgent problem to be solved.

Crowdsourcing [1] provides a distributed data collection solution. We call this solution "crowdsourcing data collection" and define it as

"*Crowdsourcing data collection is the scheme of undertaking collecting data tasks by an undefined, potentially large group of online workers in an open recruit format.*"

There are many examples of crowdsourcing data collection. ImageNet (http://www.image-net.org/about-stats), a dataset of more than 14 million images, was labeled by 50,000 online users on Amazon Mechanical Turk (AMT) (https://deepmind.com/research/open-source/kinetics). Kinetics (https://deepmind.com/research/open-source/kinetics), a dataset of human behavior that includes 700 motion categories and nearly 650,000 video clips, was collected via YouTube. LibriSpeech (http://www.openslr.org/12/), a speech corpus containing about 1000 hours of English, is from the LibriVox project. The famous Yelp dataset, from the largest public comments on Yelp (https://www.yelp.com/dataset), contains more than 8 million user comments and more than 20 images of over 200,000 businesses in 10 cities.

Crowdsourcing helps enable the collection of rich and large-scale data, which promotes the development of researches driven by data. However, crowdsourcing data collection relies on the uncertain crowd; the differences in people's ability and understanding of questions, as well as the motivation to participate in the task, will affect the effectiveness and efficiency of crowdsourcing, as well as harm the privacy of requesters and workers. Some technologies are applied to the crowdsourcing process to control the quality, cost, efficiency, and preserving privacy. These techniques focus on solving the following key issues: how to design a

task, how to select a worker (i.e., people who perform tasks), how to aggregate workers' responses, how to design an incentive mechanism, and how to protect privacy from disclosure. This survey describes the process of crowdsourcing data collection, reviews the key research topics and related technologies in its workflow, and discusses the limitations of the existing work and open problems.

This paper is organized as follows. Section 2 introduce the crowdsourcing data collection process, Sections 3–7 review the technologies adopted from five key aspects, respectively, including (1) task design, (2) task-worker matching, (3) response aggregation data, (4) incentive mechanism design, and (5) privacy-preserving. Section 8 discusses the limitations of the existing work and the future research direction. Section 9 concludes this paper.

2. Crowdsourcing Data Collection Process

Crowdsourcing data collection infrastructure comprises three major components: requester, worker, and crowdsourcing platform. A requester is a task owner, such as a person or an organization that requests a particular data collection task to be completed by workers (see Figure 1). A worker is an online user who potentially performs an assigned/selected task, motivated by interest or reward. A crowdsourcing platform is a server that manages requesters, workers, and tasks.

Figure 1 shows the process of crowdsourcing data collection. First, the requester submits the designed task and the corresponding reward to the platform (Step 1 in Figure 1). Then, the crowdsourcing platform publishes tasks (Step 2 in Figure 1). Then, the worker performs the assigned/selected tasks (Step 3 in Figure 1) and responds to the platform with collected data (Step 4 in Figure 1). Then, the platform aggregates the responses from workers and delegates them to the requester (Step 5 in Figure 1). Finally, the requester validates the task responses and determines whether to accept them, and once accepted, the reward is paid to the worker who has responded to the task (Step 6 in Figure 1).

Crowdsourcing data collection has the advantages of cheap price, fast collection speed, and large scale of data obtained. However, it still faces many challenges:

(1) *Control of the Crowdsourcing Result's Quality*. The crowdsourcing result's quality refers to the extent to which the data obtained meets and/or exceeds the requestor's expectations. The quality is affected by two aspects: task and worker. First, the task design (including whether a task description is clear and whether a task design is reasonable) has a direct influence on the worker's understanding of the task. If workers cannot accurately understand the task, it is difficult to provide high-quality data. Secondly, crowdsourcing mainly uses the online worker to collect data. Because the objective ability and subjective motivation of the worker may affect the reliability and/or the correctness of the collected data, it is certainly difficult to ensure the quality of their submitted data.

(2) *Control of the Cost*. From the task owner's perspective, the costs of crowdsourcing refer to the payment required to accomplish the task. Most crowdsourced tasks require the task owner to pay rewards to workers who have completed the task. On the AMT platform, the reward is usually a few cents per task. However, the total payment is a considerable expense, if the scale of the task is large. For example, if the price to tag a single image is 5 cents, the price to tag 50,000 images is $2,500, so it is important to consider not only the quality of the crowdsourced data but also the cost of doing so. In contrast, from the worker's point of view, since they need to expend the cost (including time, energy, and resources) to participate in tasks, they usually consider whether the reward is worthwhile compared with the cost.

(3) *Efficiency, which Refers to the Time between Publishing the Task and Completing the Task*. Efficiency is affected by the enthusiasm to participate in the task and the quality of the task completed by the workers. For example, 100 workers tag a set of photos faster than 10 workers. However, among the data submitted by workers, if the amount of qualified data is lower than the task requestor expects, a secondary publication task is required, increasing the overall time for the task to complete.

(4) *Privacy Threat, which Is an Important Issue in Crowdsourcing*. The data collected through crowdsourcing may contain a large amount of sensitive information, which is directly related to user privacy, such as the user's geographical location, travel trajectory, and personal preferences. This would cause serious security threats. For example, based on the personal information collected and tracked, Egyptian government officials' harassment and retaliation on Ushahidi reporters in 2011 can be seen as both physical intrusions to those protestors' solitude and an interference against their ideas and public demonstrations.

To address the above challenges, recently, research on crowdsourcing data collection has sprung up, for example, controlling the quality of crowdsourcing results [2–4], balancing between budget and quality [5–7], and aggregating the responses generated by workers to produce accurate results [8–10].

These challenges exist in the entire crowdsourcing process. Next, according to the crowdsourcing process, a variety of studies are reviewed from 5 aspects, including (1) task design, (2) task-worker matching, (3) response aggregation, (4) incentive mechanism design, and (5) privacy-preserving.

3. Task Design

Crowdsourcing task design is to design a task with a clear description and appropriate size to improve the readability

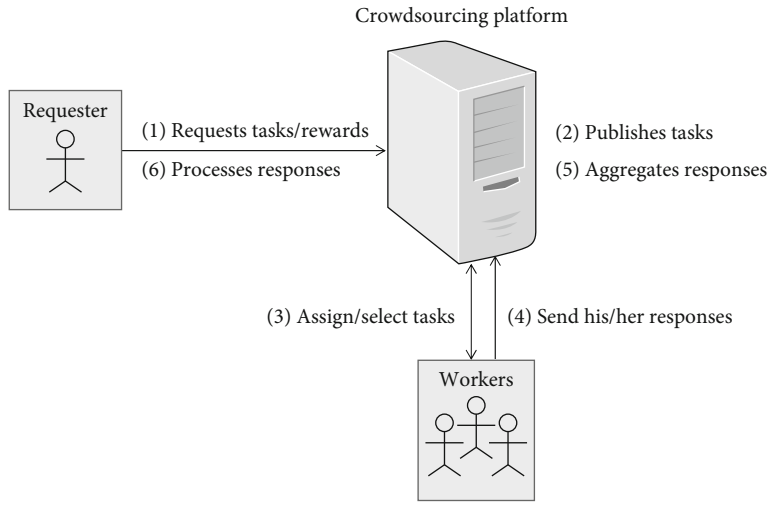

Figure 1: Crowdsourcing data collecting scenario.

of the task, to help workers complete the task quickly and correctly.

3.1. Task Description. Task description describes the task basic information (e.g., title, keywords, task content, task requirement, and task goal) and the task instructions of how to perform the task. The clarity of the task description affects the way workers perform the task and hence the quality of the crowdsourcing results [11–14].

Few task description studies have been conducted at this time. Gadiraju et al. [15] studied the quantification of task clarity. They published 71,000 microtasks on the CrowdFlower platform, including six task types: CC (Content Creation), IF (Information Finding), IA (Interpretation and Analysis), VV (Verification and Validation), CA (Content Access), and SU (Surveys). They collected workers' ratings on task goal clarity, task role clarity, and task clarity, then used the features (e.g., task type and task content) and the acquired labels to train and validate a supervised machine learning model for task clarity prediction. Gillier et al. [16] studied the influence of task instruction orientation on the quality of task (i.e., crowdsourcing innovative ideas) completion. They compared the quality of task completion under three types of task instructions: unbound, suggestive, and prohibitive. Suggestive task instruction leads to lower quality of idea originality, probably because it limits people's thinking. Wang et al. [17] believed that, if the samples in the suggestive instruction were highly original, it would motivate workers to produce high-quality original works. The research of Ipeirotis [18], on the AMT(Amazon Mechanical Turk) platform, suggests that task completion times were constrained by the way tasks were selected and followed a power-law distribution. Most tasks take 12 hours or 7 days to complete. Besides, graphic design [15] and gamification design [19, 20] not only make tasks more attractive to workers but also enhance workers' understanding of the task.

3.2. Task Decomposition. The size of a task affects the speed and quality with which it is completed. Microtasks with low granularity, such as image tagging and text recognition, generally do not require much professional skill and can be completed quickly. Macrotasks with great granularity, such as editing an article and writing a travel guide, are complex, require specific professional skills, and are difficult to accomplish by one person alone. Macrotasks usually need to be decomposed into multiple subtasks to reduce the difficulty and granularity of tasks, so as to improve the quality of task results and shorten the completion time [21–23].

According to the participants involved in task decomposition, the task decomposition method is divided into independent task decomposition and cooperative task decomposition. Independent task decomposition means that the task decomposition is completed independently by the task requestor. Collaborative task decomposition means that task decomposition is accomplished collaboratively by task requesters and workers. For example, Kulkarni et al. [23] designed an editable visual tool Turkomatic to allow workers to participate in the decomposition of crowdsourced tasks.

From the content of task decomposition, the task decomposition method can be divided into vertical task decomposition and horizontal task decomposition. The vertical decomposition method decomposes the task into multiple subtasks in a sequential sequence. The output of the former subtask is taken as the input of the latter subtask, and the output of the last subtask is the final output of the original task. For example, Bernstein et al. [21] split the text editing task into three simple subtasks: (1) find, finding what needs to be fixed; (2) fix, fixing what needs to be fixed; and (3) verify, verifying the correctness of the fix. The horizontal method divides tasks into multiple subtasks that can be done in parallel. The final output of the original task is obtained by aggregating the output of all subtasks. For example, Kittur et al. [22], based on the MapReduce framework [24], studied the decomposition of complex tasks and the integration of responses to subtasks and split the task writing an article into three simple subtasks: Partition-Map-Reduction. The "Partition" subtask is to create an outline. The "Map" subtask is to collect materials required for a chapter. Multiple instances of a "Map" subtask can be done

in parallel. The "Reduction" subtask writes paragraphs based on the collected materials. Finally, the "Reduction" subtask merges all the outlines and chapters into a single article.

4. Task-Worker Matching

There are two ways of matching between tasks and workers: (1) worker selection task and (2) platform assignment task.

(i) *Worker-selected tasks (WST)*: workers select data collection tasks of their interest from a given list published by the crowdsourcing platform.

(ii) *Platform-assigned tasks (PAT)*: the crowdsourcing platform selects available appropriate workers for a given data collection task based on various parameters, such as the quality of the worker and the budget of the task, to ensure that certain goals are achieved.

4.1. Worker-Selected Tasks (WST). With regard to WST, a worker searches the list of tasks in some sort or by entering keywords. For example, AMT, the most popular crowdsourcing platform for microtasks, sorts tasks by "recently released," "reward," and "most HITs" [25] and allows inputting keywords for searching tasks [18]. WAT helps workers find tasks quickly, but it has some limitations:

(1) Workers generally focus on the first 1~2 pages of search results, which means some tasks will not be completed for a long time, i.e., hungry task

(2) The tasks found are likely not suitable for the worker. However, to save task search time, some workers choose tasks randomly from the search results, resulting in (i) the quality of the workers' contribution being low and (ii) requestors losing contributions from other workers who are better suited to the task, and spend extra time dealing with suboptimal contributions

To complement the above search methods, some researchers propose task recommendation algorithms [26–30] based on worker characteristics and task characteristics, in order to provide workers with more appropriate tasks to choose from. Ambati et al. [28], based on the historical interactions between workers and tasks, built a preference model for workers and learned workers' preferences through "Bag-of-Words Approach" and "Classification Based Approach," so as to recommend tasks that might be of interest to workers. However, Ambati et al. [28] cannot solve the cold-start problem of a lack of historical information on new workers and tasks. To address the cold-start problem, Yuen et al. [26] proposed a task recommendation framework TaskRec based on Unified Probability Matrix Factor Decomposition, which is aimed at recommending tasks for workers in dynamic scenarios. In the real world, the time spent on completing a crowdsourcing data collection task is usually short (several minutes or even seconds), so it is possible that a task has been completed by other workers before it has been recommended to the right one. Safran et al. [30] proposed a real-time recommendation algorithm that recommends the task within milliseconds, including the following: (1) Top-K-T algorithm, recommending the most suitable K tasks for specific workers; (2) Top-K-W algorithm, recommending the most suitable K workers for a specific task.

4.2. Platform-Assigned Tasks (PAT). PAT involves assigning a given task to suitable workers based on various conditions, aimed at achieving optimization goals benefitting the requester, such as maximizing the number of tasks assigned, minimizing the cost currently, and improving the quality of task responses, or goals benefitting the worker, such as maximizing the reward received by the worker. These goals are related to the quality of workers; therefore, how to assess the quality of workers is vital. Section 4.2.1 reviews the factors influencing the quality of workers, Section 4.2.2 introduces the assessment of the quality of workers, and Section 4.2.3 introduces the assignment algorithms.

4.2.1. Factors Influencing the Quality of Workers. The quality of workers is influenced by both the workers and the tasks, specifically, including the worker's ability, the human factor of the worker, and the difficulty of the task.

The quality of workers is affected by their professional ability [25]. Workers perform better on crowdsourced tasks in areas of expertise they excel at [31]. A worker's professional ability refers to the knowledge and skills acquired by the worker through previous studies and work, which reflects the ability level of the worker in a certain field. In general, the worker's professional ability is evaluated based on his/her credentials (such as academic certificates, language level, and professional qualifications) and experience.

The quality of workers is affected by human factors [32, 33]. Kazai et al. [33] investigated the influence of human factors on the accuracy of labeling from six aspects: workers' participation motivation, familiarity with the subject involved in the task, awareness of the difficulty of the task, satisfaction with reward, and enjoyment of the task, and found that the accuracy of labeling was related to human factors.

The quality of workers is affected by task difficulty [33, 34]. Wei et al. [34] divided tasks into easy and difficult categories and learned the difficulty of tasks according to the workers' scores on the difficulty of tasks. The results of the image tagging experiment show that compared with easy tasks, difficult tasks have higher tagging accuracy, but this is related to the workers' perception of the difficulty of the task [33].

4.2.2. Assessment of Worker's Quality. An accurate assessment of the quality of workers is required before task assignment. Specifically, it can be divided into the following three evaluation methods shown in Table 1:

(1) Evaluate the quality of workers according to their reputation (EQWR)

(2) Evaluate the quality of workers by using the gold standard (EQWG)

Table 1: Assessment of worker's quality.

Method/literature	Traits	Limitation
EQWR [35, 37, 38, 36, 2, 33, 39]	Assume that the platform has obtained the worker's history of completion of the task.	The reputation value of the new user cannot be obtained.
EQWG [8, 40, 41, 33, 42]	Suppose the answer to the gold standard is known.	Increases the cost of the task requester and the time the worker takes to answer the question. Whether the gold standard is set suitable.
EQWA [8, 10]	Without knowing the correct answer to the task. The quality of the worker is inferred based on the response of the worker via data aggregation.	The limitation of the first two methods is solved, but the problem of long computation time may exist.

(3) Evaluate the quality of workers by the aggregation result of workers' responses (EQWA)

The quality of the worker is usually modeled by the reputation of the worker [35]. Reputation values are based primarily on explicit feedback (i.e., ratings of workers' contributions) from members of the crowdsourced community about workers' activities. For example, Xie et al. [36] evaluated workers' reputations based on the correctness of workers' responses. Allahbakhsh et al. [37] evaluated the reputation of the workers by using their timeliness and reliability in answering questions and their relationship with other workers or requestors. However, explicit feedback evaluation methods cannot avoid the influence of human factors such as the personal preferences or biases of the evaluator, which may result in an inaccurate assessment of the true quality of the worker [38]. Therefore, the evaluation method of implicit feedback appears, which is based on the worker's historical task completion and the worker/task profile. For example, the AMT platform typically considers highly reputable workers with more than 100 completed HIT (Human Intelligent Tasks) and more than 95% of those tasks accepted by the requester [2, 39]; these threshold values may be adjusted to accommodate your request. Reference [33] studied the accuracy of labeling under conditions of restricted and unrestricted worker qualifications; the results revealed that the accuracy is higher in the former condition.

Qualification tests for workers, using the gold standard contained in a task, is another way to assess the quality of workers [8, 40]. The gold standard refers to the known answers to the questions, usually used in qualification tests. The quality of workers is evaluated by the correct completion rate of the gold standard, so as to effectively evaluate the ability of workers to answer questions or the degree of attention to questions, thus filtering out low-quality workers. References [33, 41, 42] filtered out inattentive workers using Attention Check Questions (ACQs). The study [8] shows that the gold standard can effectively improve the accuracy of data annotation, by filtering out spammers. However, the addition of a gold standard would lead to additional tasks, resulting in increased costs for the requestor or an increase in unpaid work for the worker, which might make both parties reluctant to add a gold standard to the task.

Although the above two schemes realize the evaluation of worker quality, they both have some limitations: (1) EQWR assumes that the platform has obtained worker information; however, no information was available on the new worker or new task; (2) EQWG adds an additional cost to the requester and time spent for completion. Besides, the rationale for the gold standard is worth considering. EQWA addresses these limitations. Reference [10] uses a truth inference algorithm [8] to aggregate the workers' responses, infer the ground truth of the task, and evaluate the quality of the workers based on the ground truth. See Section 5 for a detailed description of the various aggregation methods.

4.2.3. Task Assignment. Crowdsourcing workers vary in their professional ability, work motivation, etc., resulting in different quality of workers when completing specific tasks, which makes it difficult to ensure the quality of crowdsourcing results. Although task redundancy and other methods improve the quality of crowdsourcing results [3], it will also lead to an increase in the cost paid by the task requestor and the time spent in response aggregation. Therefore, how to reasonably assign tasks to suitable workers has become one of the hottest research issues in crowdsourcing research.

At present, a large number of researches have been conducted on specific task assignment methods from the perspective of task requesters. The main idea is to balance the quality of crowdsourcing results, the number of tasks completed, and the cost (such as time and budget), to achieve the reasonable assignment of tasks.

Karger et al. [5–7] took classifying tasks as an example, aimed at obtaining reliable data annotation with minimum redundancy (the number of repeated assignments per task). In [5–7], the quality of a worker is modeled as a probability; the random regular bipartite graph is used to assign tasks to workers in the offline scenario. Ho et al. [43] proposed the exploration-exploitation algorithm in online scenarios, aimed at minimizing the total number of tasks assigned while the quality of crowdsourcing results is higher than the preset thresholds. Fan et al. [44] assumed that the quality of workers might differ in the different tasks they are engaged in and proposed an adaptive allocation framework, iCROWD. According to the similarity between tasks, a task is assigned to the workers who have performed better on similar tasks, to improve the quality and number of tasks completed as much as possible. The iRowd framework

Table 2: Factors considered and metrics of task assignments.

Reference	Factors			Metrics	
	Worker's professional ability	Worker's hobbies	Task's budget constraint	Quality of crowdsourcing result	Task assignment rate
[5]	×	×	√	√	×
[6]	×	×	√	√	×
[7]	×	×	√	√	×
[43]	√	×	√	√	×
[44]	×	×	×	√	×
[45]	√	×	×	√	√
[46]	√	×	×	√	×
[47]	√	√	×	√	×
[48]	√	√	×	√	×

includes a WarmUp component that is used to conduct qualification tests on new workers to assess their initial quality. Document [45] proposes an adaptive task allocation framework, Argo+, based on LDA (Latent Dirichlet Allocation) and Rocchio technology, to increase the success rate of task assignments. Reference [45] measures a worker's quality based on the similarity, calculated based on the worker's expertise and that required by a task. The new worker's expertise is provided by himself or set to any initial value. Literature [46] uses a decision tree to classify workers according to their expertise, then picks tasks for the worker he/she is good at, aimed at improving the quality of crowdsourcing results.

The above task assignment algorithms assume that the task to be assigned is a single task, which does not apply to the assignment of a combination task. A combination task is a task composed of several different microtasks. For example, a combination task might include making a city tour plan, selecting a book for a reading club, or rating a movie. A major feature of combined tasks is the diversity of task features [47]. Literature [48] studied the influence of task diversity on task assignment goals and proposed a task assignment method matching workers' professional abilities and hobbies.

Table 2 describes the factors considered and the metrics of various task assignments.

5. Response Aggregation

To improve the quality of a crowdsourcing result, the most commonly used crowdsourcing method based on redundancy is to assign one task to multiple workers and then aggregate the responses of multiple workers to produce a crowdsourcing result of the task [49, 50]. Much ground truth inference algorithms have been used for aggregating multiple workers' responses to infer the ground truth of the task [8, 9]. The ground truth, as a crowdsourcing result, is fed back to the requester. According to the calculation models [35], inference algorithms can be divided into the noniterative algorithm and iterative algorithm [10].

5.1. Noniterative Algorithm. The noniterative algorithm infers the ground truth of the task directly from the workers' responses [51]. Majority Voting (MV) [3], a simple method, takes a response that is consistent with the majority of the workers' responses as the ground truth. If multiple responses have the same maximum number of votes, one of them is randomly selected as the ground truth. For example, given a binary task t_i, the label option $x_i \in \{0, 1\}$, N workers label the task t_i, and the response of the worker w_j is represented by $y_i^j = w_j(x_i) \in \{0, 1\}$. The ground truth of the task t_i is $\widehat{y}_i$.

$$\widehat{y}_i = \begin{cases} 1, & \frac{1}{N}\sum_{j=1}^{N} y_i^j > \frac{1}{2}, \\ \text{random guess}, & \frac{1}{N}\sum_{j=1}^{N} y_i^j = \frac{1}{2}, \\ 0, & \frac{1}{N}\sum_{j=1}^{N} y_i^j < \frac{1}{2}. \end{cases} \quad (1)$$

MV, if (1) more than half of the workers voted unanimously and (2) the error rate of workers is uniformly distributed, can effectively improve the accuracy of the ground truth [49].

The typical MV method is only suitable for the discrete decision task. Mean and median are generally regarded as the truth of a numerical task [8]. These methods are simple to calculate and easy to implement in applications. However, if spammers are in the majority, the ground truth may seriously deviate from the real answer of the task [52]. In addition, these methods assumed there was no difference in response quality among all workers.

HP (Honeypot) [51], an advanced version of MV, is proposed. It first filters out low-quality workers based on the gold standard, then adopts MV to infer the truth based on the remainder of workers. Unlike the HP algorithm, ELICE (Expert Label Injected Crowd Estimation) [53] assumes that a worker's response is related to the difficulty of the task. Using labels provided by experts as the gold standard, the ratio of the number of workers who responded correctly to the total number of workers who participated in the gold standard is used to measure the difficulty of the task. ELICE and HP solved the problem of excessive spammers, but the

accuracy of the aggregation results depends heavily on the rationality of the gold standard and threshold setting.

5.2. Iterative Algorithm. The iterative algorithm iterates in two steps until the algorithm converges. Each iteration is divided into two steps: (1) update the aggregation truth of the task; (2) update the quality of workers. There are several iteration algorithms such as EM (Expectation Maximization), SLME (Supervised Learning from Multiple Experts), GLAD (Generative Model of Labels, Abilities, and Difficulties), FaitCrowd [54], TEST (Topic-missile-similar Tasks) [31], and ZenCrowd [55].

EM [56] carries out Maximum Likelihood Estimation (MLE) through iteration of the E (Expectation) step and the M (Maximization) step.

(i) *E step*: infer the truth of a task based on the worker quality and labels provided by workers.

(ii) *M step*: the worker quality based on the truth inferred in the Exceptation step and labels provided by workers.

When the algorithm is stopped, the inferred ground truths of tasks and the confusion matrix representing the error rate of the worker's response are returned.

EM algorithm improves the accuracy of task aggregation results since it considers the variation of worker quality. However, EM has a high computational cost and long-running time because of its iterative running characteristics, and the clustering results are closely related to the initial values of the parameters. Moreover, for a large number of label categories and a small number of labels, the confusion matrix obtained is a sparse matrix, which means that the inaccuracy of the estimated results is very high.

Similar to EM algorithm, SLME algorithm [51] also obtains aggregation results through the alternating calculation of the E step and the M step. It is assumed that the quality of workers is proportional to their professional ability, and the sensitivity and specificity of statistics are used to measure the professional ability of workers. Therefore, SLME algorithm is only applicable to binary-class tasks.

GLAD [57] takes extra consideration of the difficulty of the task based on EM algorithm and assumes the adversarial labeler can be reversed. Each iteration updates the aggregation label of the task, the professional ability of the worker, and the difficulty of the task. GLAD outperforms the commonly used "Majority Vote" heuristic for inferring image labels and is robust to both noisy and adversarial labelers.

FaitCrowd [54] uses the topic model LDA to model the professional ability of workers in different topics and assumes that a task belongs to only one topic. Different from FaitCrowd, TEST [31] assumes that a task may belong to multiple topics.

The above inference algorithms attempt to model the worker quality from multiple sides, including the expertise of a worker and the difficulty of a task. However, due to the sparsity of samples, the accuracy of the inferences is subject to certain risks. Demartini et al. [55] argued that a simple model would perform better than a complex model on a sparse dataset and therefore proposed the ZenCrowd algorithm [55]. Because the ZenCrowd algorithm uses fewer parameters, it avoids the problem of large deviation of variable estimation in the case of sparse data. Since the ZenCrowd algorithm uses maximum entropy to estimate the quality of a worker, its advantage is that it is suitable for a multiclass task that has more than two options.

Table 3 shows the main inference algorithms used in response aggregation. The factors considered by the algorithm, the suitable task type, and the efficiency are compared. Overall, the running time of the noniterative algorithm is lower than that of the iterative algorithm, but the accuracy of the aggregation result is related to the specific dataset.

6. Incentive Mechanism Design

Despite some workers being willing to work for free, most crowdsourcing workers want to be paid for their services. Hiring one user is cheap, but incentivizing extensive, reliable users to perform tasks is still crucial under a limited budget. Several studies have identified direct relations between incentives and workers' response quality and/or task execution speed [35, 58, 59]. Incentives may come in two different forms: extrinsic incentives (e.g., monetary [60] and virtual currency [61]) and intrinsic incentives (e.g., gamification point/leaderboards [19, 20]). Extrinsic incentives accelerate task execution speed [58]. Intrinsic incentives influence quality more significantly than extrinsic ones [37]. Task design typically combines extrinsic incentives and intrinsic incentives, to attract enough workers and ensure the quality of the response.

The goal of rational workers motivated by extrinsic incentives is to maximize the payoffs. A worker's payoff is the difference between the reward received by the worker and the cost incurred to complete the task. Maximizing payoffs implies minimizing the cost (e.g., effort to respond to tasks), which generally leads to the poor quality of responses. Much research on incentive mechanisms have been conducted, to design a payment rule trade-off between the number of tasks completed, the response's quality, and the payment paid by requesters (or the reward received by workers). Existing crowdsourcing incentive mechanisms can be divided into two categories: non-game theory-based incentive mechanisms and game theory-based incentive mechanisms.

6.1. Non-Game Theory-Based Incentive Mechanisms. Few studies on non-game theory-based incentive mechanisms are proposed. The existing incentive mechanisms are mainly designed from the perspective of the task requester. Reference [62] proposes a payment mechanism that takes a multiplicative form, where the worker's response to a golden question is evaluated using a score; the reward received by the worker is the sum of the minimum payment and bonus, which is the product of the score and unit bonus per task. This score is directly related to the quality of the workers' response; workers with lower response quality are given a lower score and hence paid less and vice versa. Thus, this

Table 3: Ground truth inference algorithms.

Algorithms		Worker quality model	Task model	Task type	Efficiency
Noniterative algorithm	MV [3]	×	×	Binary-class	High
	HP [51]	The probability of correctly responds to the gold standard	×	Binary-class	
	ELICE [53]	The probability of correctly responding to the gold standard	The task difficulty	Binary-class	
Iterative algorithm	EM [56]	Confusion matrix	×	Binary-class	Low
	SLME [134]	Sensitivity, specificity	×	Binary-class	
	GLAD [57]	The probability of responding correctly	The task difficulty	Binary-class	
	FaitCrowd [54]	LDA	LDA	Binary-class	
	TEST [31]	LDA	LDA	Binary-class	
	ZenCrowd [55]	The probability of responding correctly	×	Multiclass	

payment mechanism, on the one hand, prevents spammers from participating in the task and, on the other hand, encourages high-quality workers to actively participate in the task and ultimately achieves the goal of improving the quality of the response received by the requester.

Reference [63] examines the problem: if the budget is not enough to support one response per task, is it to motivate more tasks to be completed or to motivate better quality acquisition of a single response? Requallo, a flexible budget allocation framework, is proposed based on the Markov decision process, to determine the number of annotation instances and payments, ultimately maximizing the number of annotation instances under a limited budget, while ensuring quality does not fall below a certain threshold.

The aforementioned non-game theory-based approach designs the incentive mechanism directly based on the task requestor's estimate of the worker's quality. The worker either accepts or rejects the task, and there is no negotiation with the task requestor over the quality or price of the task. To solve this problem, game theory was introduced into incentive mechanisms, and hence, a large number of incentive mechanisms based on game theory have been proposed.

6.2. Game Theory-Based Incentive Mechanisms. During the crowdsourcing process, the behavior and interests of task requesters and workers interact and constrain. Based on this, researchers have proposed a large number of incentive mechanisms based on game theory to solve the utility maximization problem of the parties involved. In economics, utility refers to the degree of satisfaction people get from a good or a service.

We will focus on two types of game theory-based incentives: auction-based incentives and Stackelberg-based incentives.

6.2.1. Auction-Based Incentive Mechanism. The auction-based incentive mechanism models the interaction between stakeholders as an auction process and examines the properties of auctions under the behavior of the stakeholders. An auction-based incentive mechanism is generally evaluated according to the following desirable auction properties.

(1) *Individually Rational (IR)*. The utility of all participants is nonnegative.

(2) *Budget Feasible (BF)*. The payment paid by the requestor/platform must be less than or equal to his budget.

(3) *Compute Efficiently (CE)*. The computational complexity of the incentive mechanism algorithm is polynomial time.

(4) *Truthful (T)*. During the game, players will not provide false personal information (such as the cost of participating in the task or the value they can bring to the other party) or manipulate strategically to gain more utility. In other words, the players maximize their utility only if they have truly reported their personal information.

The first three properties ensure the feasibility of the incentive mechanism. The fourth property eliminates the fear of market manipulation among participating users.

Reference [64] designs the incentive mechanism, MSensing auction, based on the antiauction model to determine the optimal time for users to participate in the task, so as to maximize users' utility. MSensing auction is proved to be profitable (i.e., the platform should not incur a deficit.) instead of the property BF. Literature [65] models the interaction between workers and task requestors as an antiauction model and proposes an incentive mechanism under budget constraints. Initially, the workers submit bids, each of which is a task-price pair, to the platform. The platform then greedily selects bidders to maximize its utility and determines how much to pay. Bayesian inference is used to estimate worker quality and selection, and Myerson's lemma is used to determine the reward to be paid to the winner. Literature [66] considers the real-time arrival and departure of workers and proposes dynamic incentive mechanisms OMZ and OMG based on the online auction model. In the current period, workers first bid with the reserve price of accepting the task and the time of arrival and departure. The platform then decides whether to accept the service of the worker and the reward to be paid, under the remaining budget, so as to

Table 4: An auction-based incentive mechanism.

Reference	Quality	IR	BF	CE	T	Other properties	Goal	Scenario
MSensing auction [64]	×	√	×	√	√	Profitability	Maximize the user's utility	Offline
[65]	√	√	√	√	√	No	Maximize platform's utility	Offline
[66]	×	√	√	√	√	Consumer sovereignty, constant competitiveness	Maximize platform's utility	Online
TM [67]	×	√	×	√	√	Frugality	Minimize payment	Offline
[68]	√	√	√	√	√	Competitiveness	Maximize the total number of satisfied tasks	Online

maximize its utility. In addition to the four properties mentioned above, the incentive mechanisms also have the properties of *consumer sovereignty*, which guarantees that each participating user has a chance to win the auction, and *constant competitiveness*, which ensures that the mechanism has an approximate optimal solution in an offline scenario. Unlike the above work, literature [67] proposes a frugal auction-based mechanism (i.e., a nonuniform truthful mechanism, TM), which is committed to saving payment paid by the requester. Reference [68] considers that the worker quality may change over time and proposes a long-term, dynamic, quality-sensitive incentive mechanism, Melody, which models the interaction between requestors and workers as a reverse-auction model running continuously. Melody is proven to satisfy the competition property, which means that the ratio of this mechanism's solution to the optimal solution (OPT) that is computed in the offline case is O(1).

Table 4 compares auction-based incentive mechanisms.

6.2.2. Stackelberg-Based Incentive Mechanism. A Stackelberg game is used to model the competition between one player, called the *leader*, and a set of players, called the followers. A Stackelberg game consists of two stages: the leader first takes actions and knows the actions will be observed by the followers. The followers then take actions according to the actions observed. Both parties choose their own strategies according to the strategies of the other party, so as to maximize their utility under the strategies of the other party and, hence, achieve Stackelberg Equilibrium.

Literature [64] designed a truthful incentive mechanism based on the Stackelberg game, where the platform is a leader and each worker is a follower. First, the platform announces the total reward of the task. Next, each worker decides sensing time to maximize its utility. By solving the Stackelberg equilibrium of the utility function of the workers and the platform, the optimal strategy (i.e., the reward or the time) of both parties is obtained when the utility is maximum.

Reference [67] resolved the problem of minimizing payment in a scenario where both the cost of workers and the value they contribute are heterogeneous. CS-MECH, an incentive mechanism based on the Stackelberg game, was proposed. In the first stage, the requester, as a leader, announces the total payment that will be allocated to all participants. In the second stage, the workers, as followers, learn the task and other workers' information then decide their participation level which maximizes their utility. Reference [67] compares the two incentive mechanisms proposed: CS-MECH and TM, and finds that CS-MECH performs better than TM in terms of reducing the payment.

Reference [69], highlighting the collaboration between requestors and workers, proposes a novel framework, in which workers and requestors observe each other's strategies and share their information to maximize their benefit. First, the worker, as the leader, reports the optimal strategy maximizing its utility. The worker's strategy is the number of tasks it plans to complete. Next, each requester, as a follower, identifies the optimal strategy (i.e., the unit price of the task) based on the observed strategy and the private information owned (i.e., the worker's reputation) and then shares the information with the workers. Finally, based on the information observed, the worker adjusts the number of tasks to be accomplished, to maximize its utility.

Besides, privacy and security are the challenges of crowdsourcing. Given concerns about privacy disclosure and security threats, workers may be reluctant to participate in tasks [70]. Li and Cao [71] proposed two privacy-oriented incentive mechanisms, in which users are encouraged by credits to upload data without being disclosed. One scheme is implemented by a trusted third-party platform using a hashing verification equation; the other scheme is implemented by blind signature and delegate technology. Xiong et al. [72] have proposed a secure framework for reward-based spatial crowdsourcing (SECRSC), which uses homomorphism encryption technology to prevent the disclosure of information uploaded by workers.

7. Privacy-Preserving

Crowdsourced data collection faces three types of privacy threats:

(1) *Threat to Data Privacy of Workers.* The data collected may relate to the privacy of workers, such as the social activities, travel trajectory, political views, and health. The disclosure of such information probably harms workers' privacy.

(2) *Threats to Personal Information Privacy of Workers.* The personal information of the worker is uploaded by the worker when he/she registers in the crowdsourcing platform, including the worker's ID and

gender. Personal information can be stolen and used to commit crimes.

(3) *Threats to Task Privacy*. The task uploaded by the requester includes the information about the requester. An attacker may infer valuable requester information from the task description, thus endangering the requester's privacy.

To protect crowdsourcing from privacy security threats, several methods have been applied to task allocation, data aggregation, and incentive mechanisms.

(1) *Task Assignment with Privacy*. To et al. [73] introduced a trusted third party to protect worker location privacy based on differential privacy. Shen [74] designed a secure task assignment protocol using additively homomorphic encryption with the introduction of a semihonest third party. In contrast, [75–78] intend to protect both task privacy and worker privacy. References [76, 78] proposed the task assignment based on the encrypted locations of workers and requesters by homomorphic encryption. In [75, 77], a dual privacy-preserving algorithm based on anonymity is proposed in task matching in spatial crowdsourcing.

(2) *Data Aggregation with Privacy*. In the data aggregation stage, there are two common methods to protect privacy: homomorphic encryption [79, 80] and the addition of random noise perturbation data [81]. In [79], a data aggregation scheme based on additional homomorphic identity encryption (IBE) was proposed, in which data reported to SP should be encrypted using the worker's private key. This ensures that workers' data is not decrypted (except by the trusted third party). Zhuo et al. [80] proposed a data aggregation scheme that supports privacy protection and data integrity. Zhuo et al. [80], based on Brakerski–Gentry–Vaikuntanathan, proposed a verifiable homomorphic encryption scheme.

(3) *Incentive Mechanism with Privacy*. To protect personal information privacy, some incentive schemes try to support differential privacy by adding random disturbance to bidding information [82–84], which can protect workers' personal information well. Meanwhile, It also ensures that no worker can gain more benefit by claiming false bids. In addition, [85] applied homomorphic cryptography to protect the privacy of personal information (bidding) and considered the verification of incentive results. Sun and Ma [86] proposed a verifiable incentive mechanism for privacy protection based on signature and homomorphic encryption.

8. Discussion

In this section, we discuss some important open problems in crowdsourcing data collection.

8.1. More Effective Incentive Mechanism. Crowdsourcing data collection tasks involve three entities, including task requesters, workers, and crowdsourcing platforms. One of the most fundamental questions is how to recruit extensive appropriate workers. Existing research focuses on how to design tasks to attract enough reliable workers to participate in tasks. One of the most important components of the task design is the incentive mechanism. However, existing work mainly designs the incentive mechanism from the requester's perspective, in which the reward paid to workers is mainly based on the task requestor's evaluation of worker responses; the drawback is that the evaluation may not be accurate, because the task requestor may be malicious and deceptive. How to design an incentive mechanism from the perspective of the system, with constraints on both workers and requestors, is worth further study.

8.2. More Accurate Selection of Workers. Worker quality is the key basis of task assignment. Some workers exclude low-quality workers through the gold standard. However, the gold standard hidden in a crowdsourced task can lead to an increase in the cost (e.g., the payment by the requester and the time to complete tasks) of the task. Besides, if the gold standard is too difficult or not relevant to the real task, honest, professional workers may be eliminated, thus wasting the requester's resources. Therefore, further research is needed to ensure the elimination of spammers during task assignments and to ensure the accuracy of response aggregation.

8.3. Privacy-Preserving. Privacy is an important issue in crowdsourcing. The data collected through crowdsourcing may contain a large amount of sensitive information, which is directly related to user privacy, such as the user's geographical location, travel trajectory, and personal preferences. This would cause serious security threats, although some studies have incorporated privacy-preserving techniques into task assignment [76, 77], response aggregation [87], and incentive mechanisms [83, 85]. However, in crowdsourcing, malicious participants or the platform may deceive other stakeholders. Hence, the privacy threat has the unique characteristic of deceptive practices [88]. Further, developing effective strategies for protecting user privacy remains an open research problem in crowdsourcing.

8.4. Practical Application. Finally, researches on crowdsourcing data collection have been mainly carried out theoretically or verified on the prototype system of researchers. Applying theoretical research to the real world is an aspect of the development of crowdsourcing platforms that remains to be explored.

9. Conclusion

This paper summarizes the key issues faced in the process of crowdsourcing data collection, reviews relevant technologies proposed over the past decade, and discusses the similarities and differences between these technologies. Then, the present situation of crowdsourcing research and the problems that can be further studied are discussed. Finally,

we hope that our work can provide a reference for relevant researchers.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was funded by the Guangxi Key Laboratory of Trusted Software (No. kx201727), Project to Improve the Scientific Research Basic Ability of Middle-Aged and Young Teachers (No. 2019KY0226), Natural Science Foundation of China (Nos. 62066010, U1811264, and U1711263), and Natural Science Foundation of Guangxi Province (Nos. 2019GXNSFBA245049, 2019GXNSFBA245059, 2018GXNSFDA281045, and 2020GXNSFAA159055).

References

[1] M. Sahlins, "The conflicts of the faculty," *Wired Magazine*, vol. 35, no. 4, pp. 997–1017, 2009.

[2] E. Peer, J. Vosgerau, and A. Acquisti, "Reputation as a sufficient condition for data quality on Amazon Mechanical Turk," *Behavior Research Methods*, vol. 46, no. 4, pp. 1023–1031, 2014.

[3] W. Wang and Z.-H. Zhou, "Crowdsourcing label quality: a theoretical analysis," *Science China Information Sciences*, vol. 58, no. 11, pp. 1–12, 2015.

[4] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on Amazon Mechanical Turk," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*, pp. 64–67, Washington, D.C., USA, 2010.

[5] D. R. Karger, S. Oh, and D. Shah, "Efficient crowdsourcing for multi-class labeling," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1, pp. 81–92, 2013.

[6] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011*, pp. 1953–1961, Granada, Spain, 2011.

[7] D. R. Karger, S. Oh, and D. Shah, "Budget-optimal crowdsourcing using low-rank matrix approximations," in *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 284–291, Monticello, IL, USA, 2011.

[8] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing," *Proceedings of the VLDB Endowment*, vol. 10, no. 5, pp. 541–552, 2017.

[9] J. Zhang, X. Wu, and V. S. Sheng, "Learning from crowdsourced labeled data: a survey," *Artificial Intelligence Review*, vol. 46, no. 4, pp. 543–576, 2016.

[10] N. Q. V. Hung, N. T. Tam, L. N. Tran, and K. Aberer, "An evaluation of aggregation techniques in crowdsourcing," in *Web Information Systems Engineering – WISE 2013*, vol. 8181 of Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), , pp. 1–15, Springer, 2013.

[11] J. Chandler, G. Paolacci, and P. Mueller, "Risks and rewards of crowdsourcing marketplaces," in *Handbook of Human Computation*, pp. 377–392, Springer, 2013.

[12] M.-h. Wu and A. J. Quinn, "Confusing the crowd : task instruction quality on Amazon Mechanical Turk," in *InProceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Quebec, Canada, 2017.

[13] L. C. Irani and M. Six Silberman, "Turkopticon: interrupting worker invisibility in Amazon Mechanical Turk," in *CHI '13: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Paris, France, 2013.

[14] A. Kulkarni, P. Gutheim, P. Narula, D. Rolnitzky, T. Parikh, and B. Hartmann, "Mobileworks: designing for quality in a managed crowdsourcing architecture," *IEEE Internet Computing*, vol. 16, no. 5, pp. 28–35, 2012.

[15] U. Gadiraju, J. Yang, and A. Bozzon, "Clarity is a worthwhile quality-on the role of task clarity in microtask crowdsourcing," in *HT 2017- Proceedings of the 28th ACM Conference on Hypertext and Social Media*, pp. 5–14, New York, NY, USA, 2017.

[16] T. Gillier, C. Chaffois, M. Belkhouja, Y. Roth, and B. L. Bayus, "The effects of task instructions in crowdsourcing innovative ideas," *Technological Forecasting and Social Change*, vol. 134, pp. 35–44, 2018.

[17] K. Wang, J. Nickerson, and Y. Sakamoto, "Crowdsourced idea generation: the effect of exposure to an original idea," *Creativity and Innovation Management*, vol. 27, no. 2, pp. 196–208, 2018.

[18] P. G. Ipeirotis, "Analyzing the Amazon Mechanical Turk marketplace," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 17, no. 2, pp. 16–21, 2010.

[19] B. Morschheuser, "Gamification in crowdsourcing : a review," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pp. 4375–4384, Koloa, HI, USA, 2016.

[20] L. von Ahn, "Games with a purpose," *Computer*, vol. 39, no. 6, pp. 92–94, 2006.

[21] M. S. Bernstein, G. Little, R. C. Miller et al., "Soylent: a word processor with a crowd inside," in *UIST '10: Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pp. 313–322, New York, 2010.

[22] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut, "CrowdForge: crowdsourcing complex work," in *UIST'11- Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pp. 43–52, Santa Barbara, CA, 2011.

[23] A. Kulkarni, M. Can, and B. Hartmann, "Collaboratively crowdsourcing workflows with Turkomatic," in *CSCW '12: Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pp. 1003–1012, Seattle, Washington, USA, 2012.

[24] J. Dean and S. Ghemawat, "MapReduce," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[25] F. Daniel, P. Kucherbaev, C. Cappiello, B. Benatallah, and M. Allahbakhsh, "Quality control in crowdsourcing," *ACM Computing Surveys*, vol. 51, no. 1, pp. 1–40, 2018.

[26] M.-C. Yuen, I. King, and K.-S. Leung, "TaskRec: probabilistic matrix factorization in task recommendation in crowdsourcing systems," in *Neural Information Processing*, vol. 7664 of Lecture Notes in Computer Science, pp. 516–525, 2012.

[27] D. Geiger and M. Schader, "Personalized task recommendation in crowdsourcing information systems – current state of the art," *Decision Support Systems*, vol. 65, pp. 3–16, 2014.

[28] V. Ambati, S. Vogel, and J. Carbonell, "Towards task recommendation in micro-task markets," in *Proceedings of the 25th*

AAAI Workshop in Human Computation, pp. 80–83, San Francisco, USA, 2011.

[29] Y. Gong, L. Wei, Y. Guo, C. Zhang, and Y. Fang, "Optimal task recommendation for mobile crowdsourcing with privacy control," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 745–756, 2016.

[30] M. Safran and D. Che, "Real-time recommendation algorithms for crowdsourcing systems," *Applied Computing and Informatics*, vol. 13, no. 1, pp. 47–56, 2017.

[31] S. S. Gong, W. Hu, W. Y. Ge, and Y. Z. Qu, "Modeling topic-based human expertise for crowd entity resolution," *Journal of Computer Science and Technology*, vol. 33, no. 6, pp. 1204–1218, 2018.

[32] S. Basu Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das, "Task assignment optimization in knowledge-intensive crowdsourcing," *VLDB Journal*, vol. 24, no. 4, pp. 467–491, 2015.

[33] G. Kazai, J. Kamps, and N. Milic-Frayling, "An analysis of human factors and label accuracy in crowdsourcing relevance judgments," *Information Retrieval*, vol. 16, no. 2, pp. 138–178, 2013.

[34] W. Wang, X.-Y. Guo, S.-Y. Li, Y. Jiang, and Z.-H. Zhou, "Obtaining high quality label by distinguishing between easy and hard items in crowdsourcing," in *IJCAI'17 Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2964–2970, Melbourne, Australia, 2017.

[35] A. I. Chittilappilly, L. Chen, and S. Amer-Yahia, "A survey of general-purpose crowdsourcing techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2246–2266, 2016.

[36] H. Xie, J. C. S. Lui, and D. Towsley, "Incentive and reputation mechanisms for online crowdsourcing systems," in *In Proceeding 2015 IEEE 23rd International Symposium on Quality of Service, IWQoS 2015*, pp. 207–212, Portland, OR, USA, 2016.

[37] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H. R. Motahari-Nezhad, E. Bertino, and S. Dustdar, "Quality control in crowdsourcing systems: issues and directions," *IEEE Internet Computing*, vol. 17, no. 2, pp. 76–81, 2013.

[38] H. Xie and J. C. S. Lui, "Incentive mechanism and rating system design for crowdsourcing systems: analysis, tradeoffs and inference," *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 90–102, 2018.

[39] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka, and T. M. Mitchell, "Coupled semi-supervised learning for information extraction," in *In WSDM 2010- Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pp. 101–110, New York, USA, 2010.

[40] C. Akkaya, A. Conrad, J. Wiebe, and R. Mihalcea, "Amazon Mechanical Turk for subjectivity word sense disambiguation," in *In Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, CSLDAMT'10*, pp. 195–203, Morristown, NJ, USA, 2010.

[41] D. J. Hauser and N. Schwarz, "Attentive Turkers: MTurk participants perform better on online attention checks than do subject pool participants," *Behavior Research Methods*, vol. 48, no. 1, pp. 400–407, 2016.

[42] Y. S. Lee, Y. W. Seo, and E. Siemsen, "Running behavioral operations experiments using Amazons Mechanical Turk," *Production and Operations Management*, vol. 27, no. 5, pp. 973–989, 2018.

[43] C. J. Ho, S. Jabbari, and J. W. Vaughan, "Adaptive task assignment for crowdsourced classification," in *In: Proceedings of the 30th International Conference on Machine Learning*, ICML, 2013.

[44] J. Fan, G. Li, B. C. Ooi, K.-l. Tan, and J. Feng, "ICrowd: an adaptive crowdsourcing framework," in *In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 1015–1030, SIGMOD'15. New York, NY, USA: ACM, 2015.

[45] S. Castano, A. Ferrara, and S. Montanelli, "Crowdsourcing task assignment with online profile learning," in *In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 226–242, Valletta, Malta, 2018.

[46] P. Mavridis, D. Gross-Amblard, and Z. Miklós, "Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing," in *In 25th International World Wide Web Conference, WWW 2016*, Montréal Québec Canada, 2016.

[47] N. Kaufmann, T. Schulze, and D. Veit, "More than fun and money. Worker motivation in crowdsourcing – a study on Mechanical Turk," in *Proceedings of the Seventeenth Americas Conference on Information Systems*, pp. 1–11, Detroit, Michigan, 2011.

[48] M. Alsayasneh, S. Amer-Yahia, E. Gaussier et al., "Personalized and diverse task composition in crowdsourcing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 1, pp. 128–141, 2018.

[49] S. Alexander and F. David, "Utility data annotation with amazon mechanical turk," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Anchorage, AK, USA, 2008.

[50] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast - but is it good? Evaluating non-expert annotations for natural language tasks," in *In Proceedings of the 2008 Conference on Empirical Methods on Natural Language Processing*, Honolulu, Hawaii, 2008.

[51] L. Nassar and F. Karray, "Overview of the crowdsourcing process," *Knowledge and Information Systems*, vol. 60, no. 1, pp. 1–24, 2019.

[52] P. G. Ipeirotis, F. Provost, V. S. Sheng, and J. Wang, "Repeated labeling using multiple noisy labelers," *Data Mining and Knowledge Discovery*, vol. 28, no. 2, pp. 402–441, 2014.

[53] F. Khan-Khattak and A. Salleb-Aouissi, *Quality Control of Crowd Labeling through Expert Evaluation*, Workshop Comput. Social Sci, Wisdom Crowds, 2011.

[54] F. Ma, Y. Li, Q. Li et al., "Faitcrowd: fine grained truth discovery for crowdsourced data aggregation," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015*, pp. 745–754, Sydney, Australia, 2015.

[55] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *In: proceedings of the 21st international conference on world wide web, ACM*, pp. 469–478, Lyon, France, 2012.

[56] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 20–28, 1979.

[57] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan, *Whose vote should count more: optimal integration of labels*

from labelers of unknown expertise, Advances in Neural Information Processing Systems, 2009.

[58] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's Mechanical Turk," *Perspectives on Psychological Science*, vol. 6, no. 1, pp. 3–5, 2011.

[59] Y. Singer and M. Mittal, "Pricing mechanisms for crowdsourcing markets," in *In Proceedings of the 22Nd International Conference on World Wide Web, 1157–66. WWW'13*, New York, NY, USA: ACM, 2013.

[60] I. Krontiris and A. Albers, "Monetary incentives in participatory sensing using multi-attributive auctions," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 27, no. 4, pp. 317–336, 2012.

[61] J. S. Lee and B. Hoh, "Dynamic pricing incentive for participatory sensing," *Pervasive and Mobile Computing*, vol. 6, no. 6, pp. 693–708, 2010.

[62] N. B. Shah and D. Zhou, "Double or nothing: multiplicative incentive mechanisms for crowdsourcing," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 5725–5776, 2016, http://dl.acm.org/citation.cfm?id=2946645.3053447.

[63] Q. Li, F. Ma, L. S. Jing Gao, and C. J. Quinn, "Crowdsourcing high quality labels with a tight budget," in *In WSDM 2016-proceedings of the 9th ACM international conference on web search and data mining*, San Francisco, California, USA, 2016.

[64] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *In proceedings of the annual international conference on mobile computing and networking*, MOBICOM, 2012.

[65] Q. Zhang, Y. Wen, X. Tian, X. Gan, and X. Wang, "Incentivize crowd labeling under budget constraint," in *In proceedings-IEEE INFOCOM*, Hong Kong, China, 2015.

[66] Z. Dong, X.-y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility : online incentive mechanisms with budget constraint," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 1213–1221, Toronto, Canada, 2014.

[67] W. Wu, W. Wang, M. Li, S. Member, and J. Wang, "Incentive mechanism design to meet task criteria in crowdsourcing: how to determine your budget," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 502–516, 2017.

[68] H. Wang, S. Guo, J. Cao, and M. Guo, "MELODY: a long-term dynamic quality-aware incentive mechanism for crowdsourcing," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 933–943, Atlanta, GA, USA, 2017.

[69] X. Wu, S. Wang, C. Liu, W. Sun, and C. Wang, "Stackelberg game based tasks assignment mechanism using reputation in crowdsourcing," in *In proceedings -2016 international conference on identification, information and knowledge in the internet of things, IIKI 2016*, pp. 332–339, Beijing, China, 2018.

[70] W. Feng, Y. Zheng, H. Zhang, K. Zeng, X. Yu, and Y. T. Hou, "A survey on security, privacy, and trust in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2971–2992, 2018.

[71] Q. Li and G. Cao, "Providing efficient privacy-aware incentives for mobile sensing," in *In proceedings-international conference on distributed computing systems*, Madrid, Spain, 2014.

[72] P. Xiong, D. Zhu, L. Zhang, W. Ren, and T. Zhu, "Optimizing rewards allocation for privacy-preserving spatial crowdsourcing," *Computer Communications*, vol. 146, pp. 85–94, 2019.

[73] To, Hien, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," *Proceedings of the VLDB Endowment*, vol. 7, no. 10, pp. 919–930, 2014.

[74] Y. Shen, "Towards preserving worker location privacy in spatial crowdsourcing," in *in Proc. IEEE GLOBECOM*, pp. 1–6, San Diego, CA, USA, 2015.

[75] J. Shu, X. Liu, X. Jia, K. Yang, and R. H. Deng, "Anonymous privacy-preserving task matching in crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3068–3078, 2018.

[76] S. Han, J. Lin, S. Zhao et al., "Location privacy-preserving distance computation for spatial crowdsourcing," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7550–7563, 2020.

[77] S. Wang, X. J. Xiang, and Q. Sang, "A dual privacy preserving algorithm in spatial crowdsourcing," *Mobile Information Systems*, vol. 2020, 6 pages, 2020.

[78] A. Liu, W. Wang, S. Shang, Q. Li, and X. Zhang, "Efficient task assignment in spatial crowdsourcing with worker and task privacy protection," *GeoInformatica*, vol. 22, no. 2, pp. 335–362, 2018.

[79] F. Günther, M. Manulis, and A. Peter, "Privacy-enhanced participatory sensing with collusion resistance and data aggregation," in *in Proc. Conf. Cryptol. Netw. Security (CANS)*, pp. 321–336, Heraklion, Greece, 2014.

[80] G. Zhuo, Q. Jia, L. Guo, M. Li, and P. Li, "Privacy preserving verifiable data aggregation and analysis for cloud assisted mobile crowdsourcing," in *in Proc. Annu. IEEE Conf. Comput. Commun. (INFOCOM)*, pp. 1–9, San Francisco, CA, USA, 2016.

[81] J. Chen, H. Ma, and D. Zhao, "Private data aggregation with integrity assurance and fault tolerance for mobile crowd-sensing," *Wireless Networks*, vol. 23, no. 1, pp. 131–144, 2017.

[82] Y. Wen, J. Shi, Q. Zhang et al., "Quality-driven auction-based incentive mechanism for mobile crowd sensing," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4203–4214, 2015.

[83] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling privacy-preserving incentives for mobile crowd sensing systems," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pp. 344–353, Nara, Japan, 2016.

[84] T. Luo, S. K. Das, H. P. Tan, and L. Xia, "Incentive mechanism design for crowdsourcing," *ACM Transactions on Intelligent Systems and Technology*, vol. 7, no. 3, pp. 1–26, 2016.

[85] Y. Zhang, H. Zhang, S. Tang, and S. Zhong, "Designing secure and dependable mobile sensing mechanisms with revenue guarantees," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 100–113, 2016.

[86] J. Sun and H. Ma, "Privacy-preserving verifiable incentive mechanism for online crowdsourcing markets," in *in Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, pp. 1–8, Shanghai, China, 2014.

[87] X. Zhang, X. Chen, H. Yan, and X. Yang, "Privacy-preserving and verifiable online crowdsourcing with worker updates," *Information Sciences*, vol. 548, pp. 212–232, 2021.

[88] H. Xia and B. McKernan, "Privacy in crowdsourcing: a review of the threats and challenges," *Computer Supported Cooperative Work: CSCW: An International Journal*, vol. 29, no. 3, pp. 263–301, 2020.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 1690669, 11 pages
https://doi.org/10.1155/2021/1690669

Research Article

GCNRDM: A Social Network Rumor Detection Method Based on Graph Convolutional Network in Mobile Computing

Dawei Xu,[1,2] **Qing Liu**,[2] **Liehuang Zhu**,[1] **Zhonghua Tan**,[3] **Feng Gao**,[1] **and Jian Zhao**[2]

[1]*School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China*
[2]*College of Cybersecurity, Changchun University, Changchun 130022, China*
[3]*College of International Education, Hainan Normal University, Haikou 571000, China*

Correspondence should be addressed to Dawei Xu; xudw@ccu.edu.cn

Received 19 June 2021; Accepted 13 September 2021; Published 8 October 2021

Academic Editor: Lihua Yin

Mobile computing is a new technology emerging with the development of mobile communication, Internet, database, distributed computing, and other technologies. Mobile computing technology will enable computers or other information intelligent terminal devices to realize data transmission and resource sharing in the wireless environment. Its role is to bring useful, accurate, and timely information to any customer at anytime, anywhere, and to change the way people live and work. In mobile computing environment, a lot of Internet rumors hidden among the huge amounts of information communication network can cause harm to society and people's life; this paper proposes a model of social network rumor detection based on convolution networks, the use of adjacency matrix between the nodes represent user and the relationship between the constructions of social network topology. We use a high-order graph neural network (K-GNN) to extract the rumor posting features. At the same time, the graph attention network (GAT) is used to extract the association features of other nodes of the network topology. The experimental results show that the method of the detection model in this paper improves the accuracy of prediction classification compared with deep learning methods such as RNN, GRU, and attention mechanism. The innovation of the paper proposes a rumor detection model based on the graph convolutional network, which lies in considering the propagation structure among users. It has a strong practical value.

1. Introduction

In the 5G communication network environment, the number of data transmission is increasing, and there are many types of data. Different types of data storage methods are different. Therefore, it is difficult to collect, store, analyze, and query big data. The commonly used big data analysis and collection technology cannot meet the development needs of all walks of life when applied. When the technology is improved and optimized, the appropriate data mining algorithm should be selected to extract the effective information. After the analysis of the mined big data, it should be presented to users in the form of visualization of data charts, and it should be evaluated quantitatively. In order to further improve the existing data mining technology, we can automatically extract relevant information from valid data through the artificial intelligence algorithm and semantic search engine design, so as to improve the ability of data collection and screening.

As mobile computing expands into every aspect of our lives, arm-based smartphones, laptops, wearables, and other smart devices are everywhere. The number of compute-intensive use cases for these devices is rising every year, performing tasks we could only dream of in the past. Mobile computing is a new technology that covers many disciplines and a wide range. It appears with the development of mobile communication, Internet, database, distributed computing, and other technologies. Mobile computing technology will enable computers or other information intelligent terminal devices to realize data transmission and resource sharing in

the wireless environment. Its role is to bring useful, accurate, and timely information to any customer at anytime, anywhere, and to change the way people live and work.

The internet technique has been rapidly developed for the recent half-century, which causes the social media to become a convenient online platform for users to obtain information, express their opinions, and communicate. Increasingly, people are eager to participate in the discussion on some hot topics and exchange their points via social media. Therefore, some false information has been disseminated [1]. Due to the scale of social media users being large and the information on social media being easy to access for everyone, rumors can be spread rapidly in a nuclear fission manner through social media, which often triggers many instability factors and makes a great impact on economy and society. Therefore, it is particularly urgent to identify rumors on social media effectively and early to deal with the panic and threat.

Traditional rumor detection methods mainly rely on semisupervised learning of automatically labeled features, such as user features, message content features, microblog topic features, location information features, and network-type features [2]. But the abovementioned feature extraction methods are not only time-consuming and labor-intensive but also their extracted feature information is insufficient, while such methods fail to reflect the deep social network topology. Hence, they are not sufficient to judge rumors.

Since traditional machine learning rumor detection methods result in the aforementioned drawbacks, researchers have been conducting some researches about deep learning methods into introducing a rumor detection model in recent years. Typical deep learning models include recurrent neural networks (RNN), gated recurrent units (GRU), and recurrent neural networks [3]. Although these methods are able to learn time series features from rumor propagation, they ignore the effects of rumor propagation for the reason that their temporal structure features only focus on the serial propagation of rumors.

The graph attention network (GAT) is a new type of convolutional neural network which operate on graph-structured data using hidden self-attentive layers. The graph attention layer used in GAT is computationally efficient (does not require complex matrix operations with parallel computation over all nodes in the graph). In GAT, each node in the graph can be assigned a different weight based on the characteristics of its neighbors and does not rely on prior knowledge of the entire graph structure. It allows the model to reduce a large amount of physical memory during intermediate computations and enhance the efficiency of the operation of model.

Another graph neural network, K-GNN, is a generalization of GNN based on k-WL. This new model is stronger than GNN in distinguishing nonisomorphic (sub)graphs and is able to distinguish more graph attributes [4]. Triangle counting is an algorithm for counting graph structures, where the number of triangles indicates the degree of association and the tightness of organization of nodes in graph. This counting method is often used as an identification method for social network topological graphs, which can distinguish the properties of graph structures. K-GNN has better results for triangle counting problems. Therefore, in this work, we adopt K-GNN as a convolutional layer.

In order to mine the difference between rumor and non-rumor implied layer structure features with better performance, we propose a two-layer graph convolutional attention network, which obtains the propagation and dispersion properties through two parts of top-down and bottom-up GAT, respectively [5]. K-GNN obtains the information of the parent node of a node in the rumor tree. GAT aggregates the information of the children of a node in the rumor tree. Then, the propagation and dispersion representations converged at the embedding of K-GNN and GAT are combined by full concatenation to obtain the final result. Meanwhile, we connect the root features of the rumor tree with the hidden features of each graph convolution layer to enhance the influence of the root of the rumor. In addition, we use drop edge [6] in the training phase to avoid the overfitting problem of the model. The main contributions of this paper are as follows:

(1) In this paper, we solve the obstacle of traditional convolution methods when it cannot extract structural features of social networks by applying the graph convolution method for rumor detection

(2) For the problem of extracting the effective feature of different data, we obtain the more effective features by using two-layer graph convolution GAT and K-GNN as the implicit layer

(3) The detection is performed on two public datasets. The experimental results show that the results are better than the accuracy of existing schemes

2. Related Work

Automatic detection of rumors on social media has attracted a large amount of attention in recent years. Previous works on rumor detection focus on extracting rumor features from text content, user configurations and propagation structures, learning classifiers from labeled data [7–11], etc. Jing et al. [12] used time series to classify rumors, simulating changes in handcrafted social context features. Yin et al. [13] combined RBF kernels with random traversal-based graph kernels to propose a graph kernel-based hybrid SVM classifier. Ma et al. [14] constructed a rumor propagation tree kernel to detect rumors by evaluating the similarity between rumor propagation tree structures. The aforementioned works are less efficient and heavily rely on manual feature engineering to extract information feature sets.

To implement the automatic learning of high-level features, there are some rumor detection methods based on deep learning models that have been proposed recently. Yu et al. used recurrent neural networks (RNNs) to capture hidden representations from temporal content features [15]. Tong et al. [16] improved this approach by combining attention mechanisms with RNNs in order to text features with different attention. Su et al. [17] proposed a convolutional neural network- (CNN-) based approach to learn key

features scattered in the input sequence form high-level interactions between important features. Ke et al. [18] combined RNN and CNN to obtain user features based on time series. Recently, Ma et al. [19] used an adversarial learning approach to improve the performance of a rumor classifier, where the discriminator acts as a classifier and the corresponding generator based on the design of the generative model improves the discriminator by generating conflicting noise. In addition, Ma et al. constructed a tree-structured recurrent neural network (RNN) to capture the hidden representation of the propagation structure and text content [20]. However, these methods are less efficient in learning the structural features of the spread of rumors and ignore the global structural features of rumor propagation.

Compared with the deep learning models, GCN captures global structural features from graphs or trees. Su et al. [21] theoretically analyzed a graph convolution method for undirected graphs based on spectral graph theory. Subsequently, Defferrard et al. [22] developed a method called Chebyshev spectral CNN and used Chebyshev polynomials as filters. Kipf and Welling [23] proposed a new semisupervised classification method based on graph structure data. Based on the GCN model, researchers used an efficient hierarchical propagation rule, which is based on a first-order approximation of the spectral convolution on the graph. Experiments on a large number of network datasets show that the proposed GCN model is able to encode graph structure and node features in a way that facilitates semisupervised classification. After that, Veličković et al. [24] proposed the graph attention network (GAT), which operates on graph-structured data and utilizes a hidden self-attention layer allowing different importance (implicitly) assigned to different nodes. In the process of rumor propagation, it is often the important nodes of social networks that play a key role. GAT can increase the weight of important nodes, so the rumor detection process can be convolved to dig out the implied malicious nodes. We can use the Weisfeiler-Leman algorithm [25] to determine if two graphs have the same structure. In the rumor spreading process, emotional rumors are able to make rumor audiences produce similar positive and negative emotions through emotional infection, and under the influence of emotions, audiences lack rational analysis of information, thus increasing the forwarding of rumors. The emotions of rumor audiences play a mediating effect in rumor forwarding. We think that the structure of the feature map extracted between rumors should be isomorphic. Morris et al. [4] in 2019 proposed a k-order GNN based on the ensemble k-WL algorithm. Thus, we use GAT, K-GNN double-layer convolution for the rumor detection process.

3. Preliminary

3.1. Graph Attention Network. Recently, there has been an increasing interest in extending convolution to the graph domain. Graph convolution (GCN) is the first proposed model, of which the convolution operation is considered as a general "message passing" architecture, as follows:

$$H_k = M(A, H_{k-1}; W_{K-1}), \tag{1}$$

where $H_k \in \mathbb{R}^{n \times v_k}$ is the hidden feature matrix computed by the graph convolution layer and M is the message propagation function depending on the adjacency matrix A. H_{k-1} and W_{K-1} are the hidden layer feature matrix and the parameters for training, respectively.

Veličković et al. [24] proposed the graph attention network (GAT), which for each node implements a self-attentive mechanism. The attention correlation coefficient is

$$e_{ij} = a\left(W\vec{h}_i, W\vec{h}_j\right), \tag{2}$$

where e_{ij} is the attention correlation coefficient between node i and node j; W is the matrix parameter for training; $\vec{h}_i$ and $\vec{h}_j$ are the feature vectors of nodes i and j, respectively; and the corresponding weights are assigned to different neighboring nodes without either matrix operations or prior knowledge of the graph structure. For simplifying the calculation and comparison among correlation coefficients, softmax is introduced to regularize all neighboring nodes as follows:

$$\alpha_{ij} = \mathrm{softmax}_j\left(e_{ij}\right) = \frac{\exp\left(e_{ij}\right)}{\sum_{k \epsilon \mathbb{N}_i} \exp\left(e_{ik}\right)}. \tag{3}$$

The attention mechanism α is a single-layer feedforward neural network added LeakyRelu nonlinear activation with $\vec{a} \epsilon \mathbb{R}^{2F'}$ determined by the weight vector. Thus, we finally obtained the attention activation function as follows:

$$\alpha_{ij} = \frac{\exp\left(\mathrm{LeakyRelu}\left(\vec{a}^T\left[W\vec{h}_i \middle\| W\vec{h}_j\right]\right)\right)}{\sum_{k \epsilon \mathbb{N}_i} \exp\left(\mathrm{LeakyRelu}\left(\vec{a}^T\left[W\vec{h}_i \middle\| W\vec{h}_k\right]\right)\right)}. \tag{4}$$

3.2. K-GNN Convolutional Layer. The Weisfeiler-Leman algorithm [25] is used to determine whether two graphs are isomorphic, and the basic idea is to determine the independence of the current central node by iteratively aggregating the information of neighboring nodes to update the coded representation of the whole graph with the following updated formula:

$$c_l^{(t)}(v) = \mathrm{HASH}\left(\left(c_l^{(t-1)}(v), \left\{c_l^{(t-1)}(u) \mid u \epsilon \mathbb{N}(v)\right\}\right)\right), \tag{5}$$

where HASH is a mapping of graph structure nodes. By executing the above function on two graphs, one can determine whether the two graphs are isomorphic.

The GNN-based base model [4] can be implemented by the following equation:

$$f^{(t)}(v) = \sigma\left(f^{(t-1)}(v) \bullet {W_1}^{(t)} + \sum_{\omega \epsilon N(v)} f^{(t-1)}(\omega) \bullet {W_2}^{(t)}\right). \tag{6}$$

In each layer, we compute a new eigenvector $\mathbb{R}^{1\times e}$ for node v. ${W_1}^{(t)}, {W_2}^{(t)}$ is the matrix of weight parameters updated by $\mathbb{R}^{d\times e}$, and σ is a nonlinear activation function, such as rectified linear unit (ReLU) or Sigmoid.

According to the work of Gilmer et al. [26], it is also possible to replace the summation defined on the neighborhood in the above equations by a substitution invariant differentiable function, or to replace the external summation by a column vector tandem or LSTM-style update step. Thus, in the fully general case, the computation of a new identity $f^{(t)}(v)$ can be expressed as

$$f_{\text{merge}}^{W_1}\left(f^{(t-1)}(v), f_{\text{aggr}}^{W_2}(v), \left\{f^{(t-1)}(\omega) \mid \omega\epsilon\mathbb{N}(v)\right\}\right), \quad (7)$$

where $f_{\text{aggr}}^{W_2}$ aggregates the features of neighborhood nodes and $f_{\text{merge}}^{W_1}$ aggregates the neighborhood features representation calculated in the previous step. We can analogize $f_{\text{merge}}^{W_1}$ and $f_{\text{aggr}}^{W_2}$ to ${W_1}^{(t)}$ and ${W_2}^{(t)}$ of the GNN base formula. Then, we can conclude that there exists a specific set of GNN models whose effects are fully equal to the Weisfeiler-Leman algorithm (WL algorithm).

Drawing on the expansion of first-order WL to higher-order WL, the GNN is expanded to K-GNN by the following equation:

$$f_{k,L}^{(t)}(s) = \sigma\left(f_{k,L}^{(t-1)}(s)\bullet {W_1}^{(t)} + \sum_{\omega\epsilon N_L(s)} f_{k,L}^{(t-1)}(u)\bullet {W_2}^{(t)}\right), \quad (8)$$

where s denotes the subgraph consisting of k nodes and u is the neighboring subgraph of this subgraph, for a given k, we consider all k-element subsets $[V(G)]^k$ over $V(G)$. Let $s = \{s_1, \cdots, s_k\}$ be a k-set in $[V(G)]^k$, then, we define the neighborhood of s as follows:

$$N(s) = \left\{t \in [V(G)]^k, |s \cap t| = k-1\right\}. \quad (9)$$

That is, a subgraph consisting of k nodes must have and only $k-1$ common nodes in its neighboring subgraphs. With such an idea in mind, we can consider more higher-order information sets when modeling tasks with multilayer graph structures like social networks:

3.3. DropEdge. DropEdge is a new method to reduce overfitting of training models based on graph convolutional networks [27]; in each training cycle, some edges are randomly removed from the input graph and different deformation structures are generated at a certain rate, as shown in Figure 1. Thus, this method increases the randomness and diversity of the input data. Assuming that the total number of edges in graph A is N_e and the drop rate is set to p, the adjacency matrix after DropEdge calculation is shown as follows:

$$A' = A - A_{\text{drop}}, \quad (10)$$

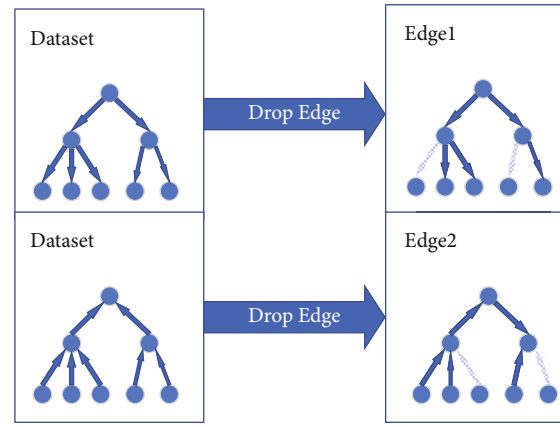

Figure 1: Schematic diagram of DropEdge process.

where A_{drop} is a matrix constructed using $N_e \times p$ edges randomly sampled from the original edges.

4. Model

4.1. Rumor. Rumor has three different meanings: words fabricated without the existence of facts, unacknowledged legends, and words circulated by folk to comment on current affairs [28]. The research of this paper is social network rumors, which refer to rumors spread through online media (e.g., microblogs, foreign websites, online forums, social networking sites, and chat software) without factual basis and offensive, purposeful discourse. They are mainly related to emergencies, public health, food and drug safety, political figures, subversion of tradition, and deviance.

Rumors spread suddenly and quickly and therefore have a negative impact on the normal social order. The rumor mill is not able to prevent the spread of rumors because of the misappropriation of concepts and generalization; the herd mentality accelerates the spread of rumors because it is better to believe in them than not to believe in them. Internet rumors, especially political rumors, can easily cause serious social problems and even social unrest and political instability due to their indistinguishability and confusing nature [29]. Many countries have made combating online political rumors an important part of rumor management and have taken comprehensive measures to crack down on them.

4.2. Social Network Rumor Detection. The current mainstream approaches treat social network rumor detection as a dichotomous problem, which is formally defined as follows.

The tweets in the social network are treated as a set $P = \{p_1, p_2, p_3, \cdots, p_i\}$, where p_i represents a tweet. Each tweet is given a label $L = \{l_1, l_2\}$, where l_1 and l_2 represent rumor and nonrumor, respectively. The task of social network rumor detection is to learn a classifier model M that maps tweet p_i into a category label l_j. The input of the model is an event containing several tweets, and the output is the rumor or nonrumor label corresponding to the event.

The social network rumor detection usually includes four stages: data processing, feature selection and extraction, model training, and rumor detection.

Data processing includes the collection of raw data and data annotation. The purpose of data collection is twofold: one is to build a dataset for training models and another is to monitor and obtain information to be detected, such as user interaction information. Data annotation is to label the data according to different needs. The data mostly is labeled as rumors or nonrumors. The experimental data in this paper are derived from two publicly available datasets. The datasets have been annotated with the correctness of each data item, and the user interaction information can be extracted from the datasets.

Feature selection and feature extraction is to select and construct the set of feature vectors that represent the data from the collected raw data optimally. For machine learning methods, feature selection and extraction are even more important than model selection. Therefore, the important work based on the machine learning method is to find more effective features to improve the accuracy of rumor detection. Rumor detection based on deep learning has a strong feature learning capability, which can obtain more high-dimensional, complex, and abstract feature data than traditional machine learning without manual feature extraction. In this paper, the signs are extracted by top-k topic word selection, and then, feature vectors are constructed based on whether the word occurs in a sentence. Although this method seems relatively simple, the complex selection of feature vectors is easy to over fit the later model training.

Model training refers to the process of selecting a model from existing classification models according to a specific problem scenario and adjusting the parameters to find an optimal model based on the classification performance of the model on the training dataset. For the social network rumor problem, it is the toughest challenge to train a classifier with accuracy in the massive data which is full of noise and still unbalanced. The main part of the model training in this paper is to adjust the parameters. The adjustment of different parameters will make different effects on the model and deploy the next parameter adjustment based on the model's performance solution.

Rumor detection is to identify the information authenticity of the information spread in social networks based on the rumor classifier obtained from model training. Our goal is to build a binary classifier to determine if a sentence is a rumor or not a rumor.

4.3. Symbols. In the following, the notation used in the model of this paper will be defined uniformly.

Let $C = \{c_1, c_2, \cdots, c_m\}$ be the rumor dataset, c_i be the ith tweet, and M be the total number of tweets. $C_i = \{r_i, w_1^i, w_2^i, \cdots w_{n_i-1}^i, G_i\}$, where n_i is the reply or retweet of c_i tweets, and r_i is the source post tweets. Each w_j^i denotes the jth relevant reply or retweet tweet, and G_i is the propagation structure of the tweet. For G_i is defined as a graph structure$\langle V_i, E_i\rangle$ [13, 14], r_i as the root node, $V_i = \{r_i, w_1^i, w_2^i, \cdots w_{n_i-1}^i\}$ and $E_i = \{e_{st}^i \mid s, t = 0, \cdots, n_i - 1\}$ denotes the set of edges from the replied post to the forwarded post or the replied post, for example, suppose w_2^i has a response to w_1^i, then there exists a directed edge $w_1^i \longrightarrow w_2^i$, which is e_{12}^i, and if w_1^i has a response to r_i, then there exists a directed edge $r_i \longrightarrow w_1^i$, which is e_{01}^i. Define $A_i \in \{0, 1\}^{n^i \times n^i}$ as the adjacency matrix, where

$$a_{st}^i = \begin{cases} 1, & \text{if } e_{st}^i \in E_i, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Define $X_i = [x_0^{i\,T}, x_1^{i\,T}, \cdots, x_{n_i-1}^{i\,T}]^T$ as a feature matrix from c_i, where x_0^i denotes the feature vector of r_i and each x_j^i denotes the feature vector of the corresponding row w_j^i.

Moreover, each source-posted tweet is associated with a real label $y_i \in \{F, T\}$, and the goal of rumor detection is to learn a classifier that

$$f = C \longrightarrow Y, \tag{12}$$

where C and Y are the set of events and labels, respectively, and the labels of the event are predicted based on the textual content, user information, and propagation structure constructed from the related posts of the event.

4.4. My Model. In this subsection, the rumor detection model proposed in this paper will be described. The core idea is to extract features from root rumors by the K-GNN layer and obtain more features of neighboring subgraphs with GAT layer. We call this model GAT_GNN. My model in this paper is shown in Figure 2.

We first discuss how to apply the GAT_GNN model to one event; X denotes the original feature matrix input to the GAT_GNN model and Edge1 and Edge2 are the matrices obtained after DropEdge processing. $X1_{\text{root}}$ and $X2_{\text{root}}$ is the first row of the $X1$ matrix and $X2$ matrix, After convolving $X1$ and $X2$ in two layers, we get $Y1$ and $Y2$. The final detection result is obtained by putting the matrix of $Y1$ and $Y2$ stitching into the binary classifier FC.

We can obtain in the rumor dataset the information of the original text and its retweets and comments. We integrate all the texts into a text database; firstly, we use jieba to split the words of this database and then use top-k to extract 5000 high-frequency words. Each rumor (root) and its forwarding and commenting message is represented by a vector of 5000 rows, each column represents a word, and the word is recorded as n if it appears n times in the text, and as 0 if it does not appear in the text message.

Each rumor and its forwarded comments form subgraph G_i, which is defined as a graph structure$\langle V_i, E_i\rangle$, and V_i is an $n \times d$ matrix, where n is the total number of users who post rumor information and its forwarded comments, and d is the number of feature vectors introduced above, and the comparison experiments show that the experimental results obtained by taking 5000 for d are better. The first row of V_i is the feature vector of rumor posting users, and we will mark this row of each subgraph to facilitate the subsequent part of the model to read this vector.

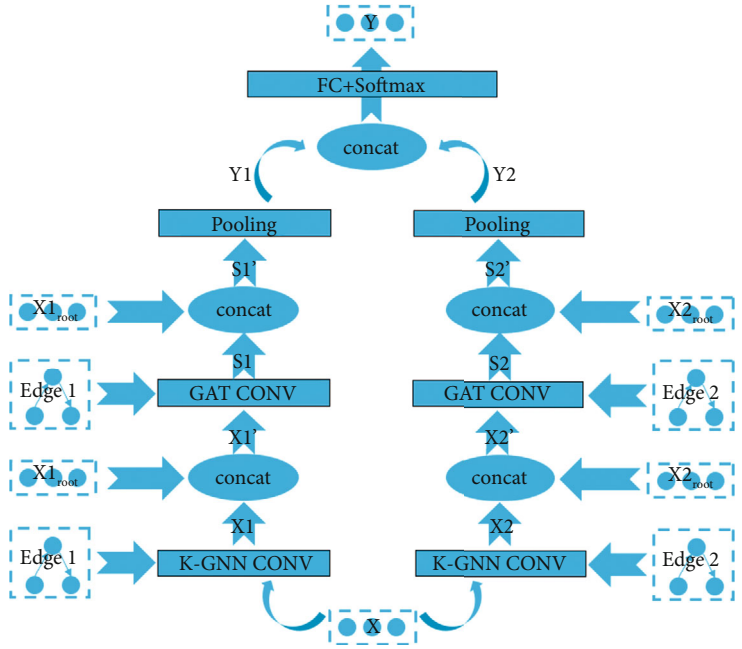

Figure 2: GAT_GNN rumor detection model.

The user information in the subgraph is extracted to get E_i. Each node represents a user, where the source node of the published rumor is used as the root node. The root node has no parent node, and the parent and child nodes of a node can be obtained from the dataset (for example, in set of microblog review, the blogger is parent node and the reviewer are child nodes. Generally, there is only one parent node and more than one child nodes.) We take the directed graph of the parent node pointing to the child nodes as Edge$D$, and the child node pointing to the parent node is Edge$U$. We do not consider the relationship between user nodes under different rumors, but only consider the relationship of each node in a subgraph. The rumor dataset includes a large number of roots to child relationship and a small number of child-to-child relationship. A large number of users choose to review and transponder, which causes unbalanced distribution of samples in the dataset. Therefore, the subgraph does not have a deep hierarchy. For this feature, we reduce the percentage of P edges by Equation (10) to generate two new adjacency matrices Edge1 and Edge2, which can avoid the overfitting problem of the model.

The detection model we built is a binary prediction of the root rumor information. The information of rumors and retweeted comments are used to construct the subgraph G_i, which also translates into a binary classification problem for the subgraph G_i. In the following, we present the whole model.

First, Edge1 and X are put into the K-GNN CONV layer for convolution, and the formula used is Equation (6) above. The original 5000 features of each node are extracted to 32; a large number of features are not conducive to node classification; we use this convolution for feature compression to obtain $X1$. Then, we extract the first row of $X1$ to note as $X1_{\text{root}}$, which is the result of compressed features of root rumor. In order to enhance the impact of the root rumor, we splice the feature vectors of the compressed root rumor into each original feature by using the horizontal splicing, which only increases the number of features and does not change the number of nodes. Each subgraph becomes a matrix H_k with $N \times 5032$. This new feature matrix is called $X1$. So, we can still use Edge1 as the adjacency matrix for calculation. The formula for the splicing process is shown below:

$$H_k = \text{concat}(X1, X1_{\text{root}}). \tag{13}$$

After that we perform the second layer of convolution by putting Edge1 and $X1^{'}$ into the GAT layer for convolution, and the formulas used are those in Equations (1)–(4) above. Here, the 5032 features of each node are compressed to 32, and the $N \times 32$ feature matrix $S1$ is obtained, in order to further enhance the effect of root rumors. We splice $X1_{\text{root}}$ with $S1$ horizontally again, which is used to increase the number of features, to obtain $N \times 64$ feature matrix $S1^{'}$. The formula for the splicing process is shown below:

$$S1^{'} = \text{concat}(S1, X1_{\text{root}}). \tag{14}$$

Finally, using mean pooling, N rows in $S1^{'}$ are turned into 1 row to obtain a 1×64 eigenvector to represent a rumor subgraph $Y1$. In exactly the same way, Edge2 is processed with the data according to the above steps to obtain $Y2$, which should also be a 1×64 eigenvector, calculated as

$$\begin{aligned} Y_1 &= \text{MEAN}\left(S_1{}'\right), \\ Y_2 &= \text{MEAN}\left(S_2{}'\right). \end{aligned} \tag{15}$$

The two information representations are then combined:

$$W = \text{concat}(Y_1, Y_2). \tag{16}$$

Finally, the label $\hat{y}$ of event y is calculated by the fully connected layer and softmax:

$$\hat{y} = \text{softmax}(FC(W)), \tag{17}$$

where $\hat{y} \in \mathbb{R}^{1\times C}$ is the probability vector used to predict all classes of event labels. In this experiment, the model parameters of this paper are trained by minimizing the cross-entropy through the real distribution of labels. The L_2 regularizer is used in the loss function of all model parameters.

5. Experiment

The performance of the model proposed in this paper is first empirically evaluated, then compared with several other baseline models. Finally, the ability of the method in this paper verified for other rumor-type detections.

The datasets chosen for the experiments in this paper are the publicly available datasets Chinese_Rumor_Dataset [30], Twitter15, and Twitter16 [31]. In the experimental dataset, nodes represent users, edges represent retweet and response relationships, and features are extracted from text messages after data processing in TF-IDF values of top 5000 words. The Twitter dataset contains two tags, namely, false rumors (F) and true rumors (T). the Twitter15 and Twitter16 datasets contain four tags: nonrumor (N), false rumor (F), true rumor (T), and unconfirmed rumor (U). Each event in Weibo is labeled according to the Sina Community Management Center, which reports all kinds of false information. Each event in Twitter15 and Twitter16 are labeled according to the authenticity labels of articles in disinformation sites (e.g., http://snopes.com, http://Emergent .info). The statistical results of the three datasets are shown in Table 1.

Table 1: Dataset statistics.

Statistics	Weibo	Twitter15	Twitter16
Posting	2062501	331612	204820
User	1265387	276663	173487
Events	3387	1490	818
True rumor (T)	1538	374	205
False rumors (F)	1849	370	205
Unconfirmed rumors (U)	0	374	203
Not a rumor (N)	0	372	205
Maximum number of post retweets	56155	1768	2765
Minimum number of post retweets	5	55	81
Length of time (hours)	68064	1337	848

5.1. Contrasting Models. We compare the proposed approach with some state-of-the-art baseline models, including the following models:

(i) DTC [32]: a rumor detection method that uses decision tree classifiers based on various handcrafted features to obtain information credibility

(ii) SVM-TS [33]: a linear SVM classifier that uses handcrafted features to construct a time series model

(iii) GRU [34]: a RNN-based model that learns temporal linguistic patterns from user comments

(iv) cPTK [35]: a SVM classifier based on propagation tree kernels is proposed based on the propagation structure of rumors

(v) RvNN [36]: a rumor detection method based on the tree recurrent neural network with GRU units, which learns rumor representation by the propagation structure

(vi) PPC_RNN+CNN [37]: a rumor detection model combining RNN and CNN, which learns rumor representation by the user's features in the rumor propagation path

(vii) GAT_GNN [38]: constructs a generalized network rumor detection model based on the GAT and GNN layers using a bidirectional propagation structure

For a fair comparison, we randomly divide the dataset into 5 parts and perform a 5-fold cross-test to obtain more stable results. On this dataset, this paper evaluates the accuracy (Acc.) of two classifications, as well as the precision (Prec.), recall (Rec.), and F1 value (F1) of each classification. The stochastic gradient descent algorithm was used to update the model parameters, and the model was optimized using Adam's algorithm [39]. The dimensionality of the feature vector hidden by each node was 64. The parameter of DropEdge was 0.1, and the parameter of dropout was 0.5. The training process was iterated for 30 cycles and applied to the early stop when the test loss stop was reduced by 5 cycles.

5.2. Experimental Results. Figures 3 and 4 give the performance of the methods in this paper and all comparative methods on the Twitter and Weibo datasets, respectively. First, in the benchmark algorithm, we observe that the deep learning method performs significantly better than those using handcrafted features. This is because deep learning methods are able to learn the high-level representations of rumors to capture effective features. This illustrates the importance and necessity of studying deep learning for rumor detection. Second, the method in this paper outperforms the PPC RNN+CNN method in all performance metrics, demonstrating the effectiveness of introducing discrete structures for rumor detection. Since RNNs and CNNs cannot process data with graph structure, PPC RNN+CNN

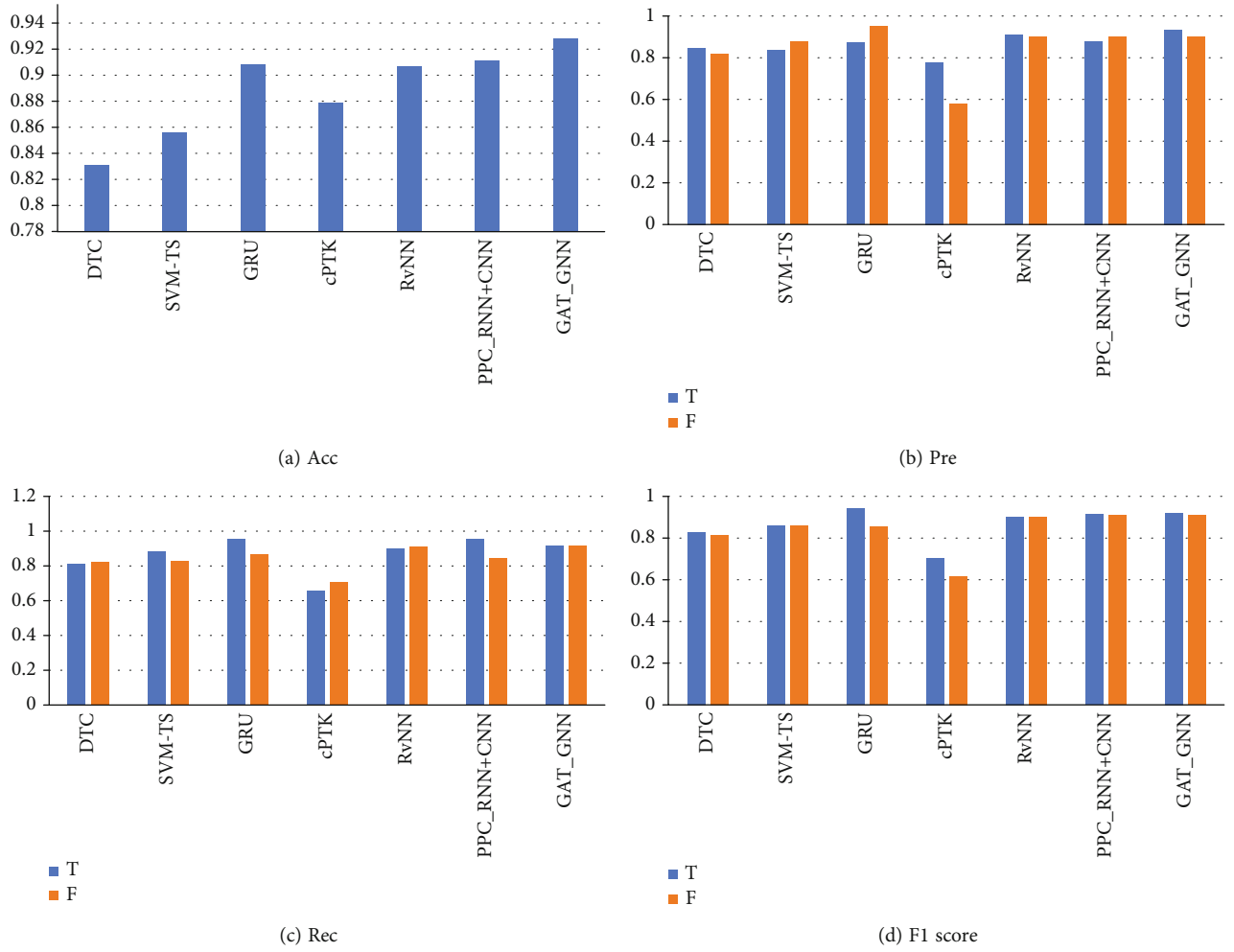

Figure 3: Rumor detection results of Weibo dataset (F: false rumor; T: true rumor).

ignores the important structural features of rumor scattering. This makes it impossible to obtain an efficient high-level representation of rumors, which leads to poor performance of rumor detection. Finally, the GAT_GNN method clearly outperforms the RvNN method. Since RvNN only uses the hidden feature vectors of all leaf nodes, it is heavily influenced by the information of the latest posts. However, the latest posts are always missing information such as comments and just follow the previous posts. Unlike RvNN, root feature augmentation makes the proposed method more focused on the information of source posts, which helps to further improve our model.

6. Discussion

Our solution is compared with other solutions in 4 aspects: accuracy, recall, precision, and F1 value. In the Weibo dataset, the GAT-GNN model is 1.2 percentage points higher in accuracy than the highest solution among other models. Some models, such as GRU, have higher recall and precision than us in the T classification but do not perform well in the F classification. In rumor detection, the classification of F, which we define as untrue rumors, is more important. This is because the goal of automated rumor detection is to save labor costs, nonrumors still account for a large proportion of text messages in the entire social network. Our model of GAT-GNN has a higher classification accuracy and performs more consistently in other judging metrics. This facilitates the application of the model to real-world detection scenarios.

While in the Twitter dataset, our model performs better relative to other models. Twitter is a four-category dataset, so we only compared the accuracy relationships under each category. Although there are some models such as cPTK, GRU, and RvNN that can be close to our judgment in a certain class of rumors, they do not perform as well in other classes of rumors. Since rumors are time-sensitive, multicategorizing rumor detection helps us to analyze where the spread will go next. Therefore, our model obtains a relatively good performance in the problem of four classifications; it depends on the ability of bilayer graph convolution to analyze complex problems. The stability of our model in experiments also paves the way for systematizing automatic rumor detection.

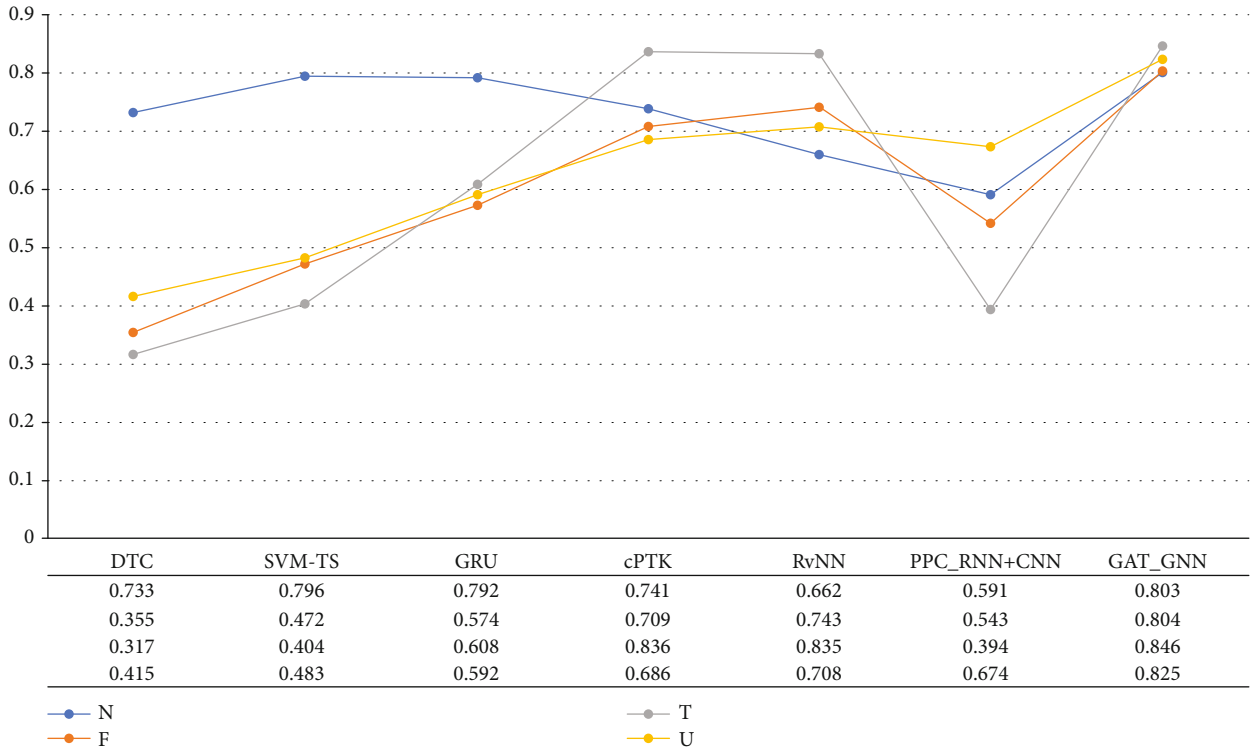

Figure 4: Rumor detection results on Twitter15 and Twitter16 datasets (N: non-rumor; F: false rumor; T: true rumor; U: unconfirmed rumor).

7. Conclusion and Future Work

In this paper, we propose a social media rumor detection model based on GAT and K-GNN, called GAT_GNN, where the graph convolution model has the ability to handle graph or tree structures, making the model more conducive to represent deeper topological networks. Also, the parent-to-child node connectivity relationship is used to model the propagation pattern. Experimental results on two real datasets show that the GAT- and K-GNN-based methods outperform the existing baseline in terms of accuracy and efficiency. On the one hand, the model in this paper considers the causal features of top-down propagation pattern of rumors along the relationship chain. On the other hand, it considers the structural features of bottom-up aggregation and diffusion of rumors within the community. Comparing with existing social network rumor detection methods, the method in this paper has better performance.

In future, we will add a module of sentiment analysis to the detection model, which will be used to improve the interpretability of rumor detection and to give a corresponding confidence level to the detection results. Finally, we aim to design a rumor detection system to detect real-time rumor comments.

Data Availability

The detailed parameter data of this article has been listed in the paper; according to this data, everyone can get the results of this paper.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the Key Program of the Joint Fund of National Natural Science Foundation of China (No. U1836212) and State Administration of Science, Technology and Industry for National Defence, PRC (No. JCKY2020602B008).

References

[1] Q. Wu, F. C. Chen, R. Y. Huang, and C. Z. Chao, "A semantic path-based approach to heterogeneous network community discovery," *Journal of Electronics*, vol. 44, no. 6, pp. 1465–1471, 2016.

[2] L. Wu, J. Li, X. Hu, and H. Liu, "Gleaning wisdom from the past: early detection of emerging rumors in social media," in *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 99–107, Houston, Texas, USA, 2017.

[3] J. Ma, W. Gao, P. Mitra, S. Kwon, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *International Joint Conference on Artificial Intelligence*, New York, USA, 2016.

[4] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, and M. Grohe, "Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4602–4609, 2019.

[5] T. Bian, X. Xiao, T. Xu et al., "Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 549–556, 2020.

[6] Y. Rong, W. Huang, T. Xu, and J. Huang, *DropEdge: Towards Deep Graph Convolutional Networks on Node Classification, 8th International Conference on Learning Representations*, ICLR, Addis Ababa, Ethiopia, 2020.

[7] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis can we trust what we RT?," in *Proceedings of the First Workshop on Social Media Analytics*, pp. 71–79, Washington, DC, USA, 2010.

[8] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on Sina Weibo," in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics - MDS '12*, New York, USA, 2012.

[9] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in *2013 IEEE 13th International Conference on Data Mining*, Dallas, TX, USA, December 2013.

[10] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, and S. Shah, "Real-time rumor debunking on twitter," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, New York, USA, October 2015.

[11] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: early detection of rumors in social media from enquiry posts," in *Proceedings of the 24th International Conference on World Wide Web*, Florida, Italy, May 2015.

[12] M. Jing, G. Wei, Z. Wei, Y. Lu, and K.-F. Wong, "Detect rumors using time series of social context information on microblogging websites," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, New York, USA, October 2015.

[13] L. Yin, X. Luo, C. Zhu, L. Wang, Z. Xu, and H. Lu, "ConnSpoiler: disrupting C&C communication of IoT-based botnet through fast detection of anomalous domain queries," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1373–1384, 2020.

[14] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 708–717, Vancouver, Canada, 2017.

[15] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A convolutional approach for misinformation identification," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3901–3907, AAAI Press, 2017.

[16] C. Tong, L. Xue, H. Yin, and J. Zhang, "Call attention to rumors: deep attention based recurrent neural networks for early rumor detection," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, Cham, 2018.

[17] S. Su, Z. Tian, S. Li et al., "IoT root union: a decentralized name resolving system for IoT based on blockchain," *Information Processing & Management*, vol. 58, no. 3, article 102553, 2021.

[18] W. Ke, Y. Song, and K. Q. Zhu, "False rumors detection on Sina Weibo by propagation structures," in *2015 IEEE 31st International Conference on Data Engineering*, Seoul, Korea (South), April 2015.

[19] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors on twitter by promoting information campaigns with generative adversarial learning," in *The World Wide Web Conference on - WWW '19*, pp. 3049–3055, San Francisco, USA, 2019.

[20] J. Ma, W. Gao, and K.-F. Wong, "Rumor detection on twitter with tree-structured recursive neural networks," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1980–1989, Melbourne, Australia, July 2018.

[21] S. Su, Z. Tian, S. Liang, S. Li, S. du, and N. Guizani, "A reputation management scheme for efficient malicious vehicle identification over 5G networks," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 46–52, 2020.

[22] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 29, pp. 3844–3852, 2016.

[23] N. T. Kipf and M. Welling, "Semi supervised classification with graph convolutional networks," in *Proceedings of the International Conference on Learning Representations*, Toulon, France, 2017.

[24] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, https://arxiv.org/abs/1710.10903.

[25] J. Qiu, Y. Chai, Z. Tian, X. du, and M. Guizani, "Automatic concept extraction based on semantic graphs from big data in smart city," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 1, pp. 225–233, 2020.

[26] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*, Sydney, Australia, 2017.

[27] Y. Rong, W. Huang, T. Xu, and H. Junzhou, "The truly deep graph convolutional networks for node classification," 2019, https://arxiv.org/abs/1907.10903.

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

[29] "thunlp/Chinese_Rumor_Dataset [EB/OL]," 2019, https://github.com/thunlp/Chinese_Rumor_Dataset.

[30] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[31] D. Xu, Z. Tian, R. Lai, X. Kong, Z. Tan, and W. Shi, "Deep learning based emotion analysis of microblog texts," *Information Fusion*, vol. 64, pp. 1–11, 2020.

[32] D. Mudali, L. K. Teune, R. J. Renken, K. L. Leenders, and J. B. T. M. Roerdink, "Classification of Parkinsonian syndromes from FDG-PET brain data using decision trees with SSM/PCA features," *Computational and Mathematical Methods in Medicine*, vol. 2015, Article ID 136921, 2015.

[33] H. Bisgin, O. U. Kilinc, A. Ugur, X. Xu, and V. Tuzcu, "Diagnosis of long QT syndrome via support vector machines classification," *Journal of Biomedical Science and Engineering*, vol. 4, no. 4, pp. 264–271, 2011.

[34] H. Ming, Y. Lu, Z. Zhang, and M. Dong, "A light-weight method of building an LSTM-RNN-based bilingual tts system," in *2017 International Conference on Asian Language Processing (IALP)*, Singapore, December 2017.

[35] A. Wu, D. Pi, J. Chen, M. Xie, and J. Cao, "Rumor detection based on propagation graph neural network with attention mechanism," *Expert Systems with Applications*, vol. 158, article 113595, 2020.

[36] L. Yang and Y. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, 2018.

[37] F. Xing and C. Guo, "Mining semantic information in rumor detection via a deep visual perception based recurrent neural networks," in *2019 IEEE International Congress on Big Data (BigDataCongress)*, Milan, Italy, July 2019.

[38] S. Y. Louis, Y. Zhao, A. Nasiri et al., "Graph convolutional neural networks with global attention for improved materials property prediction," *Physical Chemistry Chemical Physics*, vol. 22, no. 32, pp. 18141–18148, 2020.

[39] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral Networks and Locally Connected Networks on Graphs, Computer Science," 2013, http://arxiv.org/abs/1312.6203.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9969867, 9 pages
https://doi.org/10.1155/2021/9969867

WILEY | Hindawi

Research Article

Exploring Security Vulnerabilities of Deep Learning Models by Adversarial Attacks

Xiaopeng Fu,[1] **Zhaoquan Gu**,[1] **Weihong Han**,[1] **Yaguan Qian**,[2] **and Bin Wang**[3]

[1]*Cyberspace Institute of Advanced Technology (CIAT), Guangzhou University, Guangzhou 510006, China*
[2]*School of Big Data Science, Zhejiang University of Science and Technology, Hangzhou 310023, China*
[3]*Network and Information Security Laboratory, Hangzhou Hikvision Digital Technology Co, Ltd., Hangzhou 310051, China*

Correspondence should be addressed to Zhaoquan Gu; zqgu@gzhu.edu.cn

Received 4 March 2021; Accepted 16 August 2021; Published 27 September 2021

Academic Editor: Federico Tramarin

Nowadays, deep learning models play an important role in a variety of scenarios, such as image classification, natural language processing, and speech recognition. However, deep learning models are shown to be vulnerable; a small change to the original data may affect the output of the model, which may incur severe consequences such as misrecognition and privacy leakage. The intentionally modified data is referred to as adversarial examples. In this paper, we explore the security vulnerabilities of deep learning models designed for textual analysis. Specifically, we propose a visual similar word replacement (VSWR) algorithm to generate adversarial examples against textual analysis models. By using adversarial examples as the input of deep learning models, we verified that deep learning models are vulnerable to such adversarial attacks. We have conducted experiments on several sentiment analysis deep learning models to evaluate the performance. The results also confirmed that the generated adversarial examples could successfully attack deep learning models. As the number of modified words increases, the model prediction accuracy becomes lower. This kind of adversarial attack implies security vulnerabilities of deep learning models.

1. Introduction

With the fast development of artificial intelligent technologies, deep learning models have been widely adopted in more and more areas [1–3]. In particular, they have been adopted not only in target detection, image classification, and other applications in the field of CV (Computer Vision) [4, 5] but also in more and more NLP (Nature Language Processing) applications, such as sentiment classification, spam classification, and machine translation [6–8].

Compared with traditional machine learning models, deep learning models have the following advantages. First, deep learning models have a strong fitting ability, which can approximate any complex function. The dimensionality of deep learning models can reach an infinite number; hence, the data fitting ability is much more powerful than traditional models. Second, deep neural networks contain many hidden layers that contain many hidden nodes; more hidden nodes are shown to provide stronger performance capabilities than traditional machine learning models. Third, the introduction of a convolutional neural network and recurrent neural network further improves the performance of neural networks, so that they can better deal with specific problems by feature extraction and contexture analysis. Finally, deep learning models can also be combined with probabilistic methods, which enable these models with high inference ability as the random factors could improve the reasoning ability of deep neural networks. Meanwhile, compared with traditional machine learning, deep learning models have better mobility, which makes the models easily adapted in various application scenarios.

Even though deep learning models play an important role in both CV and NLP fields, it does not imply that these models are completely secure and trustful. Since deep learning models lack theoretical analysis, recent studies have shown that deep learning models are very vulnerable to

adversarial attacks, which generate adversarial examples to mislead the model by adding small perturbations to the original input. These security risks may incur severe consequences such as misrecognition in security-sensitive applications and privacy leakage during the deployment and execution of deep learning models.

In this paper, we are to explore the security vulnerabilities of deep learning models by adversarial attacks. This vulnerability property of deep learning models was first discovered in the image processing field. Only a small change of one or several pixels in the original image can cause the deep learning models to output an incorrect label to the modified data. Since this change compared to the original image is very small, human eyes can hardly detect any difference, while deep learning models for image classification would make incorrect prediction, which may lead to serious consequences. For example, a driverless system may cause a serious traffic accident if the system misidentifies a STOP sign on the road.

Not only image recognition tasks but also many NLP tasks face the challenge of adversarial examples. In this paper, we focus on the adversarial attacks in the NLP field. In [9], it proved that adversarial examples could successfully attack Google perspective API, making the models output an incorrect toxicity degree. Chinese text classification models are also threatened by such adversarial examples. Compared with adversarial attacks in image processing, generating adversarial examples in the text field is quite different and much more difficult. The challenges of adversarial attacks in the NLP field include the following aspects:

(1) The text data is discrete [10]. In the image processing field, the image can be regarded as continuous data and the adversarial attacks can be conducted by traditional gradient-based methods. However, text data is discrete, and it is more difficult to adopt traditional gradient-based methods directly to generate adversarial examples for textual analysis

(2) When generating adversarial samples for image data, only one or a few pixels in the original data are modified. This modification is basically indistinguishable to human eyes. However, in the textual analysis field, even if only a character in a word is modified, it will be much easier to be caught by humans and such modification might cause people to misunderstand the meaning of the original text

Therefore, we need to address the above two when exploring security vulnerabilities in textual analysis deep learning. In this paper, we propose the visual similar word replacement (VSWR) algorithm to solve these challenges. To begin with, to solve the problem of data discreteness, the proposed VSWR algorithm directly adds perturbations to the original text, instead of mapping the original text to a vector space. Afterwards, our proposed method could use the gradient-based method to find out the appropriate word to be modified. Second, to solve the second problem, we use words that are visual similar to replace the words in the original text, which would not cause obvious differences to humans and could not be noticed easily by humans.

We summarized the contributions of this paper as follows:

(1) We proposed an algorithm called visual similar word replacement (VSWR) to generate adversarial examples for textual data, and we show the security vulnerability of the deep learning models when faced with such adversarial examples

(2) We use the VSWR algorithm to generate adversarial examples on sentiment analysis datasets, and the adversarial examples are utilized to attack the pretrained deep learning classification models

(3) The experimental results show that the generated adversarial examples can successfully interfere with the classification of the deep learning model. Specifically, only changing 25% of the original text can reduce the classification accuracy of the model from 95% to 60%

The rest of the paper is organized as follows. The next section briefly introduces related research results on textual adversarial examples. Section 3 presents the preliminaries, including the system model and the problem definition, and then proposes the VSWR algorithm. And the experimental results are provided in Section 4; the discussion is also shown here. Finally, we make a brief summary of this paper and shed light on some future directions in Section 5.

2. Related Work

2.1. White-Box Attacks. The attacker fully understands all the information of the model and conducts an adversarial attack on the model on this basis. Therefore, the attacker can find out the relatively weak module of the model to perform targeted adversarial attacks. This attack method can test the robustness of the model against adversarial attacks in the worst case.

Although there are differences between textual data and image data, the idea of generating adversarial examples in the image field can also be used in the textual field. In [11], it puts forward a method to generate text adversarial examples named HotFlip, which represents text data as one-hot vectors, then modified one character of a certain word in the text, so as to achieve the effect of attacking neural networks. In [12], it applies FGSM [13] and JSMA [14] algorithms that use gradient descent to determine the perturbation in the image domain to generate text adversarial examples.

In fact, we have little knowledge about the neural network models we are using, including the parameter value of each layer even its structure. So, gradient methods have many restrictions.

2.2. Black-Box Attacks. Because of the limitation of white-box attacks in practical scenarios, many researchers turn their attention to black-box attacks.

In black-box attacks, an attacker knows nothing about the internal structure of the attacked model, training

parameters, defense methods (if any defense methods are applied), or other information about the attacked model. The attackers can only interact with the model through input and output. Since manufacturers will not disclose information about the models they apply, most of the current contacts are black-box attacks.

In this case, the attacker generates adversarial examples by directly modifying the words in the text data or the letters/characters in the words. In [15], it proposed the AddSent method to attack the reading comprehension system by adding a carefully constructed sentence after the original text. The generated sentence could make the system make incorrect results. However, the way of adding sentences is very imperceptible, and these sentences could be easily discovered by human readers. In [16], it proposed the attack method based on the Metropolis-Hastings algorithm to replace, insert, or delete a word in the text to generate adversarial examples, while it modifies a character in the word in text in [17].

2.3. Limitation. Although much progresses have been made in attacking deep learning models, there is still much space for improvement. For example, the adversarial examples, generated by the sentence-level attack and the word-level attack, can be easily recognized by humans, while the adversarial example generated by the char-level attack can be defended by the spell check module [18]. In this paper, we propose a novel method based on the word replacement strategy of visual similar words to generate textual adversarial examples.

Figure 1 is a simple example of generating an adversarial example by the visual similar word replacement method. As shown in the figure, the original text is recognized as a positive review by the designed deep neural network model. However, we only change the word "sweet" to the word "sweat" which looks similar; the modified text is recognized as a negative review by the deep neural network model. According to the example, only changing one character of a single word in the original data could lead to a contrary label by the pretrained model.

3. Materials and Methods

3.1. Materials. Before giving our method for adversarial example generation, we show briefly the introduction of some definitions that are used in our method. In addition, we also formulate the proposed problems to explore security vulnerabilities of deep learning models.

3.1.1. System Model. We use T to represent an English text and get a word list W by segmenting the original text. An English text T which is made up of n words can be represented as $T = [w_1, w_2, \cdots, w_i, \cdots, w_n]$, where the ith value of T stands for the ith word $w_i \in W$ of this text. We use $Y = [y_1, y_2, \cdots, y_m]$ to represent the label of text T; m means that this dataset has m categories. It is expressed as a one-hot vector. For example, all the portion's values in Y_j of a text T_j with a label k are 0 except y_k. Since there are only two categories in the dataset we use, there are only two portions in Y.

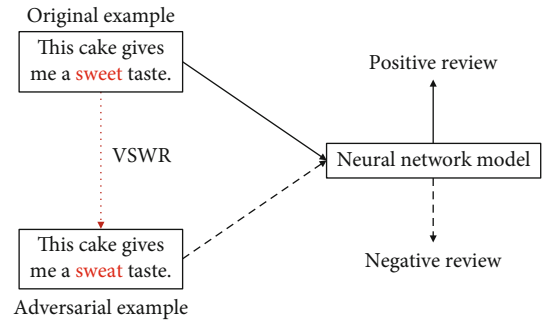

Figure 1: An example of generated texts by the visual similar word replacement algorithm.

A mapping f_θ from a text T to its label Y needs to be learned by a deep learning model which we call M, where θ are the parameters of M; they are optimized by calculating the gap of $f_\theta(T)$ and its label; the smaller of the difference between $f_\theta(T)$ and Y, the more suitable θ is.

3.1.2. Adversarial Examples. Given a well-trained model M, whenever we enter a text T_a into this model, it can give us the label Y_{T_a} of the text. An adversarial example T_a' of T_a is almost the same to T_a except a little bit of artificial perturbations δ; in this paper, δ is a visual similar word of the keyword w_i of T_a; we use $T_a' = (w_1, w_2, \cdots, \delta, \cdots, w_n)$ to represent the adversarial of T_a. When using adversarial examples T_a' as the input of model M, the model will give a different prediction $Y_{T_a'}$ from Y_{T_a}. We summarize this process into the following formulas:

$$T_a' = T_a + \delta, M(T_a) = Y_{T_a}, \quad M\left(T_a'\right) \neq Y_{T_a}. \tag{1}$$

3.1.3. Problem Definition. Since the dataset we used to verify the effect of the algorithm proposed in this paper is a binary dataset, there are only two possibilities for the label of a text T_a, $M(T_a) = 1$ or $M(T_a) = 0$. Assume that a piece of text data T_a whose label $M(T_a) = 0$. The problem we solved in this paper is to generate T_a' by the method proposed in this paper; when we use T_a' as the input of model M, $M(T_a') = 1$. And T_a' must follow the following principles:

(1) The difference between T_a and T_a' must be as small as possible, which means we can only replace a small number of words in the original data to ensure a human's reading

(2) All the visual similar words we choose to replace keywords in original data must be in word list W, and it must be spelled similarly to keywords to ensure the imperceptibility of adversarial examples

3.2. Method. In black-box attacks, the attacker knows nothing about the internal structure and parameters of the model, so it is impossible to calculate the influence of the

gradient change on the model prediction result. The method proposed in this paper is to solve the problem of the inability to pass the gradient, in the case of calculating the words that need to be modified, how to modify the original text to generate adversarial examples, which mainly includes the following steps: word scoring, visual similar word searching, and visual similar word replacement.

3.2.1. Word Scoring. Since each word in a text data has a different contribution to the final label given by pretrained models when models classify text data, for example, in sentiment analysis tasks, words with a particularly strong emotional color such as wonderful will have a greater impact on the results of the classification than other words. Therefore, when generating text adversarial examples in a black-box context, in order to ensure the success rate of the attack, the importance of the words in the original text needs to be ranked first.

According to the scores of these words, we extracted those words with higher scores which means they have the greatest impact on the text label in the original text, as "keywords," and then, adversarial examples of the original text are generated through operations on the words such as destruction or replacement. We use a method that combines context and the position of the word in the entire text to score the word. Through this method, the words in the original text are scored to obtain the words that have the greatest impact on the label.

First of all, we use the training dataset to train a neural network model M. Whenever you input a text data to M, it gives the label of this text and the confidence of each label. Since text data has strong contextual relevance, when scoring a word in the text, it is necessary to consider the context of the word. Assuming a piece of text data T consists of n words, then the text can be expressed as $T = [x_1, x_2, x_3, \cdots, x_n]$. Given a piece of text data which can be presented to $T_{\text{text}} = [x_1, x_2, x_3, \cdots, x_n]$, we give the ith word of this text by the following ways.

(1) Head Score. As we have already trained a model M to classify text data, when we give M a piece of text T_{text}, it will return the confidence of each label, and we present it by $M(T_{\text{text}})$. We define the head score of the ith word to be the score of the text composed of the first $i-1$ words minus the score of the text composed of the first i words. We first choose the first $i-1$ words to form a text T_{head}, using it as the input of model M to get $M(T_{\text{head}})$ of T_{head}. Next, ith is added to T_{head} to form T'_{head}, so that we can get M'_{head} by query model M. So, the head score of the ith word can be presented as follows:

$$S_i^{\text{head}} = M(T_{\text{head}}) - M\left(T'_{\text{head}}\right) = s(x_1, x_2, \cdots, x_{i-1}) - s(x_1, x_2, \cdots, x_i). \tag{2}$$

(2) Tail Score. The same as head score, we define the tail score of the ith word to be the score of the text composed of the words which are after the ith word minus the score of the text which added the ith word to the former text.

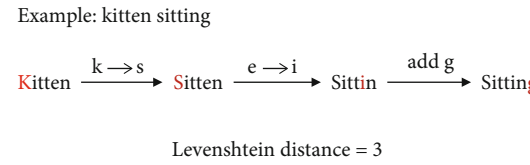

Figure 2: Levenshtein distance.

These two texts are presented as $T_{\text{tail}} = (x_{i+1}, x_{i+2}, \cdots, x_n)$ and $T'_{\text{tail}} = (x_i, x_{i+1}, \cdots, x_n)$; using these two texts, we query model M to get $M(T_{\text{tail}})$ and $M(T'_{\text{tail}})$, so the tail score of the ith word is as follows:

$$S_i^{\text{tail}} = M(T_{\text{tail}}) - M\left(T'_{\text{tail}}\right) = s(x_{i+1}, x_{i+2}, \cdots, x_n) - s(x_i, x_{i+1}, \cdots, x_n). \tag{3}$$

(4) Without Score. Without score is calculated by the text without the ith word $T_{\text{without}} = (x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$ and all of this text $T_{\text{text}} = (x_1, x_2, \cdots, x_n)$. We query these two texts above and then get $M(T_{\text{without}})$ and $M(T_{\text{text}})$. Without score is presented as follows:

$$\begin{aligned} S_i^{\text{without}} &= M(T_{\text{without}}) - M(T_{\text{text}}) \\ &= s(x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n) \\ &\quad - s(x_1, x_2, \cdots, x_n). \end{aligned} \tag{4}$$

(4) Combined. Since the position of the ith word in the whole text is different, a certain weight needs to be added when combining the above three scores. For the words at the top of the text, we reduce the weight of the head score, and the others are on the contrary. We determine the weight by calculating the proportion of the text before and after the ith word in the entire text. The bigger the i is, the higher the weight of the head score is. Finally, we can get the final score of the ith word through the following formula.

$$S_i^{\text{combined}} = \frac{2i/nS_i^{\text{head}} + 2(n-i)/nS_i^{\text{tail}} + S_i^{\text{without}}}{3}. \tag{5}$$

After the words are scored, the scores of the words need to be sorted. The higher the score of the word, the greater the influence of the word on the final prediction label given by the model when the model classifies the text, and this word is a "keyword" in the original text; modifying the "keywords" can improve the offensiveness of the adversarial samples and increase the attack success rate.

3.2.2. Finding Visual Similar Words. When generating text adversarial examples, the imperceptibility of adversarial examples needs to be considered (that is, the modified text cannot be changed too much from the original text). Therefore, the adversarial example generation algorithm proposed in this paper selects words that are similar in spelling to the keywords in the original text when replacing keywords. But there are many ways to calculate the similarity of two strings, such as Euclidean distance, Levenshtein distance, and cosine

Let English text data X be presented as $X = [\omega_1, \omega_2, \cdots, \omega_i, \cdots \omega_n]$
A neural network model M
A dataset list consist of all the word in this dataset O
for each $\omega_i \in X$ do:

$$s_i^{head} = s(\omega_1, \omega_2, \cdots, \omega_{i-1}) - s(\omega_1, \omega_2, \cdots, \omega_i)$$

$$s_i^{tail} = s(\omega_{i+1}, \omega_{i+2}, \cdots, \omega_n) - s(\omega_i, \omega_{i+1}, \cdots, \omega_n)$$

$$s_i^{without} = s(\omega_1, \omega_2, \cdots, \omega_{i-1}, \omega_{i+1}, \cdots, \omega_n) - s(\omega_1, \omega_2, \cdots, \omega_n)$$

$$s_i^{combine} = (2i/n)s_i^{head} + (2(n-i)/n)s_i^{tail} + s_i^{without}/3$$

sort w_i by $s_i^{combine}$ to get keywords list
for each w in keywords list and o in O do:
  calculating $lev_{(w.o)}(|\omega|, |o|)$
  for each w_i find out o_{w_i} by min $(lev_{w.o}(|\omega|, |o|))$
use o_{w_i} replace ω_i
$X' = [w_1, w_2, \cdots, o_{w_i}, \cdots, w_n]$ is the adversarial example of X

Algorithm 1: The visual similar word replacement algorithm.

similarity. In this paper, we choose Levenshtein distance to measure the similarity between two strings. We also tested other similarity calculation methods and finally chose Levenshtein distance because it is the most direct and fastest method. Figure 2 depicts an example which shows the calculation method of Levenshtein distance, from which we can easily see that Levenshtein distance can be used to easily calculate the similarity of two words, and the smaller the distance is, the closer the two words are.

Levenshtein distance is also known as edit distance, which refers to the minimum number of edit operations required to convert one string to another between two strings. Editing operations include replacing one character with another, inserting a character, and deleting a character.

For two strings a, b with lengths of $|a|$ and $|b|$, it is necessary to calculate the edit distance between $a(1, 2, \cdots, |a| - 1)$ and $b(1, 2, \cdots, |b|)$ and then add 1 (for the case of an increased operation, add the last one character). Or calculate the edit distance between $a(1, 2, \cdots, |a|)$ and $b(1, 2, \cdots, |b| - 1)$, and then add 1 (for the deletion operation, delete the last character). Or calculate the edit distance between $a(1, 2, \cdots, |a| - 1)$ and $b(1, 2, \cdots, |b| - 1)$, and then add 1 (for the modification operation in case, modify one character) and then take the minimum of these three as the minimum edit distance of the previous step, and so on to the first character.

Use $|a|$ and $|b|$ to represent the length of the two strings a and b, respectively; then, the Levenshtein distance between the two strings is $\text{lev}_{a,b}(|a|, |b|)$, where

$$\text{lev}_{a,b}(i,j) = \begin{cases} \max(i,j), & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i-1,j) + 1, \\ \text{lev}_{a,b}(i,j-1) + 1, \\ \text{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)}, \end{cases} & \text{otherwise}, \end{cases} \quad (6)$$

Table 1: Detailed information of Yelp review dataset.

Yelp review dataset	Train	Test
Classes	2	2
Num	400000	30000
Positive : negative	1 : 1	1 : 1
Average words	209	194

in which $1_{(a_i \neq b_j)}$ is an indicator function, when $a_i = b_j$, its value is 0; otherwise, its value is 1. $\text{lev}_{a,b}(i, j)$ represents the Levenshtein distance between the first i characters of a and the first j characters of b (i and j are subscripts starting from 1).

We use Levenshtein distance to measure the similarity between two words. The smaller the Levenshtein distance is, the more similar these two words are. So, we exchange the word chosen by the scoring module with another word whose Levenshtein distance to the keyword is smallest. In this way, the difference between the replaced text and the original text will not be very large, and it is not easy to be noticed by humans and affect human reading.

3.2.3. Generating Adversarial Examples. In the first step, we found those words with high scores, which means they are more important than the other words to the label given by deep learning models; these selected words are called "keywords." Then, we use these keywords to form a keyword list; by calculating the Levenshtein distance between the words in the keyword list and the words in the word list of the dataset, we can find out those words that are similar to the keywords in the keyword list. We only need to find out the word with the shortest Levenshtein distance to keywords. Finally, we only need to use the visual similar words found in the previous step to replace the corresponding keywords in the original text. Then, we can generate the adversarial examples that could fool deep learning models with high imperceptibility. The VSWR algorithm is described in Algorithm 1.

Table 2: Detailed information of Amazon review dataset.

Amazon review dataset	Train	Test
Classes	2	2
Num	400000	30000
Positive : negative	1 : 1	1 : 1
Average words	180	188

Table 3: Accuracy of deep learning models on the Yelp review dataset.

Model	LSTM	BiLSTM
Accuracy	95.6934%	95.647%

Table 4: Accuracy of deep learning models on the Amazon review dataset.

Model	LSTM	BiLSTM
Accuracy	88.483%	88.550%

Table 5: Accuracy changes when the num of replaced words increases (Yelp review dataset).

Num of replaced words	0	10	20	30	40	50
LSTM	0.9569	0.9296	0.8906	0.7812	0.6640	0.6171
BiLSTM	0.9564	0.8984	0.8906	0.7578	0.6562	0.5546

Table 6: Accuracy changes when the num of replaced words increases (Amazon review dataset).

Num of replaced words	0	10	20	30	40	50
LSTM	0.8848	0.7891	0.7500	0.6875	0.6406	0.6484
BiLSTM	0.8855	0.7656	0.7266	0.6953	0.6328	0.5781

4. Results and Discussion

4.1. Result

4.1.1. Dataset. We use the following two datasets: Yelp review dataset and Amazon review dataset to train two deep learning models that are designed for sentiment analysis. Then, we use our proposed algorithm to attack the pre-trained models.

The Yelp review dataset is the comment data of the Yelp website, which is extracted from the Yelp Dataset Challenge 2015. There are only two categories: positive and negative in this dataset. This dataset contains two parts. The first part is the label of the data in this dataset. The second part is the detailed comments of users on the products they bought. The entire dataset contains 560000 pieces of English texts, including 280000 positive samples and 280000 negative samples. However, many texts are only composed of few words. We delete these texts from the dataset and only remain the texts with enough words for attack. The detailed information of the dataset is shown in Table 1. As for the Amazon review dataset, it consists of reviews from Amazon, and these reviews are also divided into two categories: positive and negative. Each piece of data consists the comment text and the label. In the experiment, we also deleted redundant data from the Amazon review dataset; the detailed information is shown in Table 2.

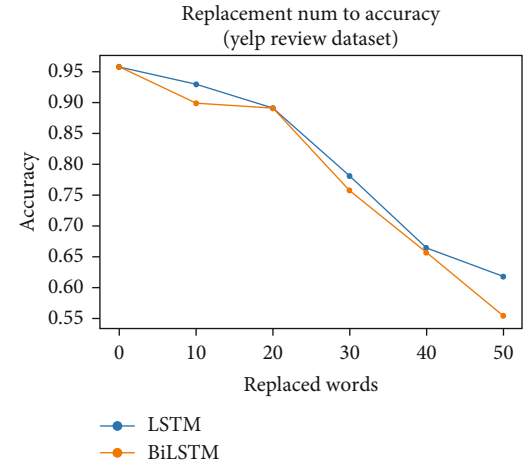

Figure 3: Accuracy changes with the number of replaced words (Yelp review dataset).

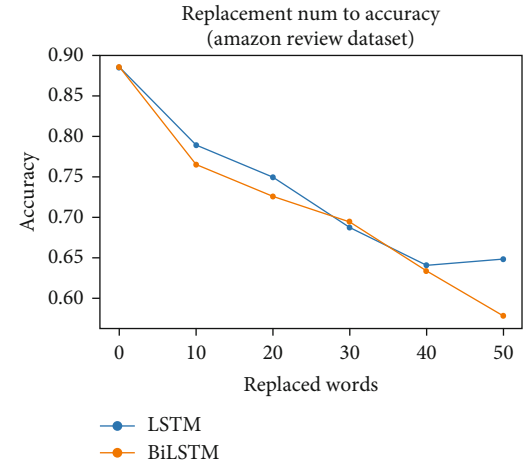

Figure 4: Accuracy changes with the number of replaced words (Amazon review dataset).

4.1.2. Models. Since we consider the context of text data when generating adversarial examples, the recurrent neural network- (RNN-) based model is the most suitable model for the experiment. To provide high accuracy, we finally choose LSTM (Long Short-Term Memory) and BiLSTM (Bidirectional Long Short-Term Memory) as the trained models to attack.

Tables 3 and 4 show the classification accuracy of these models on the Yelp review dataset and the Amazon review dataset. According to these tables, we can find that the two deep learning models could achieve good performance when solving the sentiment analysis task. Specifically, the accuracy rate of both two models can reach as high as 95% on the Yelp review dataset, while the accuracy rate of the models on the Amazon review dataset exceeds 88%. In the following

Table 7: A generated adversarial example by the VSWR algorithm (Yelp review dataset).

	Original text	Adversarial example
Text	i went here for the lunch buffet and was blown away! not the greatest location, i was a little weary at first. but, i was proven wrong because the food was out of this world. i have been to numerous thai places around the valley and this is by far the **best**!	i went here for the lunch buffet and was blown away! not the greatest location, i was a little weary at first. but, i was proven wrong because the food was out of this world. i have been to numerous thai places around the valley and this is by far the **Rest**!
Label	Positive	Negative

Table 8: A generated adversarial example by the VSWR algorithm (Amazon review dataset).

	Original text	Adversarial example
Text	while not expensive to initially purchase, this system is expensive in the long run. if you mop your floors regularly or have a lot of rooms to mop, you'll be constantly replacing pads and cleaner and batteries, none of which are cheap.additionally, this system does not clean my laminate floors well. it leaves them cloudy/ hazy and they never look or feel 100% clean afterwards. when i go over with a damp paper towel where i just mopped, it always shows a fair amount of dirt. and looking at the pad when you remove it from the mop you can see why. the way it is designed, the most pressure is on the front edge of the mop. the center and back of the pad do not receive nearly as much pressure and pick up almost nothing.in short, for small jobs and irregular cleaning, this mop will do. otherwise, it is inefficient, expensive **and** ultimately **frustrating**.	while not expensive to initially purchase, this system is expensive in the long run. if you mop your floors regularly or have a lot of rooms to mop, you'll be constantly replacing pads and cleaner and batteries, none of which are cheap.additionally, this system does not clean my laminate floors well. it leaves them cloudy/ hazy and they never look or feel 100% clean afterwards. when i go over with a damp paper towel where i just mopped, it always shows a fair amount of dirt. and looking at the pad when you remove it from the mop you can see why. the way it is designed, the most pressure is on the front edge of the mop. the center and back of the pad do not receive nearly as much pressure and pick up almost nothing.in short, for small jobs and irregular cleaning, this mop will do. otherwise, it is inefficient, expensive **andd** ultimately **frustarating**.
Label	Negative	Positive

parts, we show that the trained two deep learning models would achieve bad performance against generated adversarial examples.

4.1.3. Attack Performance. We use the method proposed in this paper to process the test dataset and then evaluate the effectiveness of each model, respectively. For the same model, as the number of words which are replaced in the original text increases, the prediction accuracy of the trained deep neural network model becomes lower and lower. As shown in Tables 5 and 6, with the number of replaced words (by its visual similar word), the models' prediction accuracy decreases.

In Figures 3 and 4, we show the change of the models' accuracy when the number of replaced words increases on two datasets. The x-axis represents the number of replaced words in the original text, and the y-axis denotes the accuracy of the deep learning models. From the figures, the original accuracy of both models is higher than 95% and 88%, respectively, when no word is replaced. When we replace more words as the x-axis, the accuracy of both models decreases as the two curves in the figures.

In Tables 7 and 8, we show some generated adversarial texts on the two datasets by the VSWR algorithm. In Table 7, an original text from the Yelp review dataset is recognized as "positive" by the BiLSTM model. However, as the algorithm only changes "best" to "Rest," the model classifies the generated text as "negative." Similarly, an original text from the Amazon view dataset is recognized as "negative"; by changing two words to their visual similar words, the generated text is classified as "positive." Clearly, the trained models would achieve bad performance against the generated adversarial examples, which implies the effectiveness of our proposed method.

5. Discussion

From Figures 3 and 4, we can find that the BiLSTM model is more susceptible to the influence of adversarial examples compared with the LSTM model. This is because the BiLSTM model fully considers the relationship between the scored words and the context. In addition, the extracted keywords by our method for the replacement in generating adversarial examples are more suitable for the BiLSTM model. During the preprocessing step of the dataset, we filter out the texts that are composed with only a small number of words; this is because humans could easily recognize short texts that are modified. Hence, we select relatively long texts in the dataset for the attack experiment. During our experiments, when the number of modified words is small, the attack effect on the two models is not good, but when we increase the number of modified words to 25% of the original text (on Yelp review dataset), the classification accuracy of the model can be reduced from 0.95 to 0.55. This result was also confirmed on the Amazon review dataset; the change of the words can reduce the model accuracy from 0.88 to 0.57.

Actually, some existing adversarial attacks could largely reduce the prediction accuracy of the neural network models. However, some of them change the characters in a

word or split a word by some special symbols; the generated adversarial texts can be easily noticed by humans since some generated words do not exist in the vocabulary. In our paper, we select words that look similar to the original one for replacement, which could successfully fool both humans and deep neural network models. Although we need to modify 50 words to achieve good attack performance for the dataset, the modified words look quite similar to the original one and only 25% of words are modified on average, which is also acceptable.

However, there are also some questions that can be concluded by the examples above. For example, those words with strong emotions such as "interesting" and "bad" have not been changed during the algorithm, which means the trained deep learning models do not mainly rely on these words for classifying.

In this paper, we only verify the security vulnerabilities of deep learning models by the adversarial attack methods. Indeed, there are also many other methods to show security vulnerabilities. For example, we can modify the training data such that the trained deep learning model cannot study the correct data distribution; this kind of attack is also called data poisoning. In addition, some methods are proposed to steal privacy of deep learning models, such as inferring data from the training set, stealing parameters of the models. These methods would cause privacy leakage of deep learning models.

6. Conclusions

In this paper, we explore the security vulnerabilities of deep learning models by adversarial attacks. Specifically, we propose the visual similar word replacement method to attack several deep learning. This method firstly sorts the importance of the words in the original dataset and selects the words that have the greatest influence on the classification result as the keywords. At the same time, the original data is processed to obtain a word list containing all the words in the original dataset, and then, we use the word found in the word list whose Levenshtein distance between keywords is 1 to replace the keyword. The replaced text is the generated adversarial example of the original text. We also conducted experiments on the sentiment analysis datasets, and the results proved that the adversarial examples generated by this method could successfully attack the deep learning models such that they would make misclassification. In addition, as the number of modified words increases, the impact on the neural network model becomes more and more significant.

In the future, we will try to extend this method to attack more classification models for other textual analysis tasks, such as text generation, spam filtering, and machine translation. At the same time, we also try to improve the proposed VSWR method such that less words could be selected for replacement.

Data Availability

The Yelp review dataset is the comment data of the Yelp website. In our work, we filter out the texts in the dataset that contain words less than 50, since the changes of short texts are easier to be noticed by humans. Readers who are interested can get the dataset processed in https://drive.google.com/drive/folders/1AWhczX50NVyr-gciAIz96Vtb-WVWS9L5D?usp=sharing.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported in part by the National Key R&D Program of China 2019YFB1706003, the Guangdong Key R&D Program of China 2019B010136003, and the National Natural Science Foundation of China under Grant Nos. 61972106 and 61902082.

References

[1] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is BERT really robust? A strong baseline for natural language attack on text classification and entailment," 2019, https://arxiv.org/abs/1907.11932.

[2] Z. Gu, L. Wang, X. Chen et al., "Epidemic risk assessment by a novel communication station based method," *IEEE Transactions on Network Science and Engineering*, 2021.

[3] L. Wang, J. Niu, and S. Yu, "SentiDiff: combining textual information and sentiment diffusion patterns for Twitter sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 10, pp. 2026–2039, 2020.

[4] Z. Gu, W. Hu, C. Zhang, H. Lu, L. Yin, and L. Wang, "Gradient shielding: towards understanding vulnerability of deep neural networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 921–932, 2021.

[5] Z. Gu, Y. Su, C. Liu et al., "Adversarial attacks on license plate recognition systems," *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1437–1452, 2020.

[6] W. Zou, S. Huang, J. Xie, X. Dai, and J. Chen, "A reinforced generation of adversarial examples for neural machine translation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2019, http://arxiv.org/abs/1911.03677.

[7] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012.

[8] Z. Gu, Y. Cai, S. Wang et al., "Adversarial attacks on content-based filtering journal recommender systems," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1755–1770, 2020.

[9] H. Hossein, S. Kannan, B. Zhang, and R. Poovendran, "Deceiving Google's perspective API built for detecting toxic comments," 2017, https://arxiv.org/abs/1702.08138.

[10] P. Yang, J. Chen, C.-J. Hsieh, J. L. Wang, and M. I. Jordan, "Greedy attack and Gumbel attack: generating adversarial examples for discrete data," *Journal of Machine Learning Research*, vol. 21, pp. 1–36, 2020.

[11] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "Hotflip: white-box adversarial examples for text classification," 2017, https://arxiv.org/abs/1712.06751.

[12] Z. Gong, W. Wang, B. Li, D. Song, and W. S. Ku, "Adversarial texts with gradient methods," 2018, https://arxiv.org/abs/1801.07175.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, https://arxiv.org/abs/1412.6572.

[14] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387, Saarbruecken, Germany, 2016.

[15] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," 2017, https://arxiv.org/abs/1707.07328.

[16] H. Zhang, H. Zhou, N. Miao, and L. Li, "Generating fluent adversarial examples for natural languages," 2020, https://arxiv.org/abs/2007.06174.

[17] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 50–56, San Francisco, CA, USA, 2018.

[18] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "TextFool: fool your model with natural adversarial text," 2019, http://groups.csail.mit.edu/medg/ftp/psz-papers/2019%20Di%20Jin.pdf.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 4214784, 9 pages
https://doi.org/10.1155/2021/4214784

Research Article

An Effective Algorithm for Intrusion Detection Using Random Shapelet Forest

Gongliang Li,[1,2] **Mingyong Yin**,[2] **Siyuan Jing**,[3] **and Bing Guo**[1]

[1]*School of Computer, Sichuan University, Chengdu, China 610000*
[2]*Institute of Computing Applications, China Academy of Engineering Physics, Mianyang, China 621000*
[3]*School of Artificial Intelligence, Leshan Normal University, Leshan, China 614000*

Correspondence should be addressed to Bing Guo; guobin@scu.edu.cn

Received 19 June 2021; Accepted 16 August 2021; Published 3 September 2021

Academic Editor: Lihua Yin

Detection of abnormal network traffic is an important issue when builds intrusion detection systems. An effective way to address this issue is time series mining, in which the network traffic is naturally represented as a set of time series. In this paper, we propose a novel efficient algorithm, called RSFID (Random Shapelet Forest for Intrusion Detection), to detect abnormal traffic flow patterns in periodic network packets. Firstly, the Fast Correlation-based Filter (FCBF) algorithm is employed to remove irrelevant features to decrease the overfitting as well as the time complexity. Then, a random forest which is built upon a set of shapelet candidates is used to classify the normal and abnormal traffic flow patterns. Specifically, the Symbolic Aggregate approXimation (SAX) and random sampling technique are adopted to mitigate the high time complexity caused by enumerating shapelet candidates. Experimental results show the effectiveness and efficiency of the proposed algorithm.

1. Introduction

Intrusion detection system (IDS) is an important part of modern network security protection infrastructure. It is aimed at analyzing the traffic packages online or offline to identify the intrusion behaviors from networks. However, some attacks are very difficult to be detected. For example, distributed denial of service (DDos) attack creates tens of thousands of zombie computers and orders them attack a target server at the same time. It not only fabricates source IP address to avoid detection, but also increases the traffic exponentially. Therefore, an efficient technique for detection of intrusion behaviors is required.

The basic principle of intrusion detection technology is to build a normal or abnormal behavior model through the analysis of relevant data which may be stored in security log or audit database and compare the model with the user behavior to identify the potentially harmful behavior [1]. It is obvious that the key to victory is the discovery of the effective behavior characteristics (or patterns) from relevant data. As an effective technology to search and mine hidden information from massive data, data mining is very suitable for intrusion detection. So far, a variety of data mining technologies, including classification, clustering, and anomaly detection, have been successfully applied in intrusion detection.

Classification is a popular technology in intrusion detection. Given a set of labeled instances, it learns a function which can assign a label to a new unlabeled instance. Lee and Stolfo [2] firstly extracted rules from audit data and used the rules for detection of abnormal behavior in network traffic. Gao et al. [3] employed the Apriori algorithm to extract traffic flow patterns from network data and subsequently used the K-means cluster algorithm to generate a detection model. Besides that, many popular techniques of classification were adopted in intrusion detection, such as K-nearest neighbor [4, 5], decision tree [6, 7], and support vector machine [8]. Recently, deep learning, which attracts lots of attentions from community, is employed in intrusion detection and achieves state-of-art performance [9, 10]. However,

the intrinsic defect of deep learning, a.k.a. lack of interpretability, prevents it to be a ready-made panacea.

In this paper, we employ time series classification (TSC) technique to detect abnormal behaviors based on the offline traffic flow data. Specifically, the adopted technique is called shapelet, which is a new primitive in the field of TSC. The contributions of this work include the following:

(1) We propose a novel TSC framework for intrusion detection which is composed of a feature selection algorithm (FCBF) and a shapelet-based random forest classifier

(2) The traffic flow data is represented by SAX and the shapelet candidates, which are used to train the classifier and are sampled randomly. By this way, the running time is greatly mitigated

(3) The proposed algorithm, called RSFID, is validated on several data sets of intrusion detection. The results prove that RSFID is effective to detect abnormal behavior in traffic flow. Since the intrinsic advantage of the shapelet-based method, i.e., good interpretability, our work provides a different solution to solve the problem of intrusion detection

The rest of the paper is organized as follows. Section 2 briefly introduces the development of IDS and recalls the basic knowledge of shapelet-based TSC. Section 3 explains the details of the RSFID algorithm, and the theoretical analysis of complexity is also given. Next, the experimental details are introduced, and the results are analyzed in deep in Section 4. Finally, Section 5 gives conclusions.

2. Background

2.1. Intrusion Detection and Time Series Classification. Intrusion detection is aimed at extracting patterns or characteristics of user's behaviors by analyzing the security log and then identifying the dangerous behavior in the system. The solutions can be divided into two types. The first is building a safe/normal behavior model as the evaluation criteria of user behavior. When the user behavior is obviously different from the safe/normal behavior model, it is considered to be an intrusion. The second is building an unsafe/abnormal model (a.k.a. intrusion behavior) based on a set of obtained data of intrusion. If the detected behavior is similar with the unsafe/abnormal model, we think it is an intrusion.

There are abundant ways to handle the intrusion detection problem, such as classification, clustering, and abnormal detection. Besides those, time series classification is considered to be a suitable solution because the traffic flow data is temporal ordered. Luo et al. [11] modeled the brain activity as time series and used the K-nearest neighbor algorithm to detect the abnormal. Chin et al. [12] evaluated abundant algorithms of anomaly detection which based on symbolic time series analysis. Recently, Wei et al. [13] proposed an assumption-free technique for anomaly detection using time series classification. Kim et al. [14] introduced a shapelet-based method to detect abnormal behavior in network traffic. However, the algorithm is based on exhaustive search; hence, it is too time consuming.

2.2. Shapelet-Based Time Series Classification. Shapelet refers to time series subsequences that are maximally representative of a class [15]. Due to the strong interpretability, it has attracted abundant attentions from the community. In the last decade, over a hundred papers have been published to develop this technique. Later, we will recall some basic knowledge in this field.

Definition 1 (Time series). The time series is denoted by a sequence of values $T = t_1, t_2, \cdots, t_{|T|}$, where $|T|$ is the length of time series. Data points $t_1, t_2, \cdots, t_{|T|}$ are typically arranged by temporal order and spaced at equal time interval.

Definition 2 (Time series data set). A time series data set D is a set of pairs of time series T_i and its corresponding label $c_i \in C$, i.e., $D = \{\langle T_1, c_1\rangle, \langle T_2, c_2\rangle, \cdots, \langle T_n, c_n\rangle\}$, where n is the number of time series in the data set and C is the set of labels.

Furthermore, since most of the time series data in real world are multidimensional, such as the monitoring data collected from Internet of Things system, the ECG monitoring system, and the IDS, we use $T_{i,j}$ to represent the j-th dimension of the i-th time series and the k-th position of $T_{i,j}$ can be written as $T_{i,j,k}$.

Definition 3 (Subsequence). A time series subsequence S is a contiguous sequence of a time series. Subsequence of length l of time series $T_{i,j}$ starting at position k can be denoted as $S_{i,j}^{k,l} = T_{i,j,k}, T_{i,j,k+1}, \cdots, T_{i,j,k+l-1}$. Furthermore, the overall subsequence of time series T with length l is denoted as $\Psi(T, l)$.

For simplicity, lots of concepts introduced below only explain the one-dimension time series and all of them can be naturally extended to multidimension.

Definition 4 (α distance and β distance). The α distance and β distance define the distance between two time series T_1, T_2 with the same length and the distance between a subsequence S and a time series T, respectively.

In this paper, we also use Euclidean distance to measure the two types of distance and the formulas are given below, where m is the length of two time series.

$$\begin{aligned} \mathrm{dist}_\alpha(T_1, T_2) &= \sqrt{\frac{1}{m}\sum_{i=1}^{m}(T_{1,i} - T_{2,i})}, \\ \mathrm{dist}_\beta(S, T) &= \min_{S' \in \Psi(T,|S|)} \mathrm{dist}_\alpha\left(S, S'\right). \end{aligned} \tag{1}$$

Shapelets which are maximally representative of a class are essentially a set of subsequence. Our purpose is to choose a subset of subsequences which have strong discriminatory

power to build a classifier. To measure the discriminatory power of a shapelet candidate, we give the definition of split and information gain (IG).

Definition 5 (Split). A split is a tuple $\eta = \langle S, \tau \rangle$, where S is a time series subsequence and τ is a distance threshold which can split the data set D into two subsets D_L and D_R.

Given a time series subsequence S, we can calculate the distance between S and all series in D, i.e., $\text{dist}_\beta(S, T_i)$. If $\text{dist}_\beta(S, T_i) \leq \tau$, the time series T_i will be added to D_L; otherwise it will be added to D_R.

Definition 6 (IG). The information gain of a split $\eta = \langle S, \tau \rangle$ can be calculated as follows:

$$\text{IG}(\eta) = E(D) - \frac{n_L}{n} E(D_L) - \frac{n_R}{n} E(D_R). \quad (2)$$

The symbols n_L and n_R denote the number of time series in D_L and D_R, respectively, and $E(D) = \sum_{i=1}^{|C|} (n_i/n) \log (n_i/n)$ is the entropy of data set.

Given a time series subsequence S and a data set of time series D, we can calculate the distance between S and all series in D and obtain a set of distance sorted in ascending order $\langle d_1, d_2 \cdots, d_n \rangle$. We say a split $\eta = \langle S, \tau \rangle$ is a shapelet candidate that there is no $\eta' = \langle S, \tau' \rangle$ that $\text{IG}(\eta') > \text{IG}(\eta)$. To distinguish the shapelet candidate with split, we use symbol $\theta = (S, \tau)$ to represent it. It is not difficult to find that there are infinite splits for a specific subsequence. To limit the search space, we only detect the mean value of any two adjacent distance value, i.e., $(d_i + d_{i+1})/2$.

Ye and Keogh [15] firstly introduced the concept of shapelet; meanwhile, they proposed a Brute-Force algorithm to search the best candidate to be the final shapelet embedded into a decision tree classifier. The algorithm suffers from two problems that the exhaustive search is too time-consuming, and the decision tree training is embedded in the search process. There are some solutions to address the first problem, including [15–17]. Due to the limit of page, we skip the introduction of these techniques. Next, we introduce an interesting technique, called shapelet transformation, which separates the shapelet searching and the classifier building by transforming the original time series data set to a new feature space [18].

Definition 7 (Shapelet transformation). Given a time series data set $D = \{T_1, T_2, \cdots, T_n\}$ and a feature space Σ consisted of a set of selected shapelet, i.e., $\Sigma = \{S_1, S_2, \cdots, S_k\}$, shapelet transformation is a matrix M with n rows and k columns, where $M_{i,j} = \text{dist}_\beta(S_j, T_i)$.

It is easy to find that, by shapelet transformation, the temporal characteristic in original time series has been removed. Hence, a large amount of classical data mining techniques can be applied to the time series mining. However, there are also some problems in this technique. For example, the process of shapelet selection is also time-consuming, and the selected shapelets are always be irrelevant and redundant [19–21].

3. The Proposed Method

3.1. The RSFID Algorithm. The idea of the RSFID algorithm (Random Shapelet Forest for Intrusion Detection) is descripted as Figure 1. There are five steps that learn a random shapelet forest (a.k.a. the classifier) from the original time series. Firstly, the raw data of network traffic requires to be represented by SAX [22]. Although there are some other techniques for presentation of time series data, such as PAA, APCA, and DFT, SAX has been proven to be the most efficient technique to compress time series data [23]. The details of SAX technique can be found in [22]. It must be noted that the traffic flow data not only contains real value, but also includes other data types. For example, the KDD CUP 99, which is a famous data set of intrusion detection, contains real value and nominal value. Therefore, the raw data must be preprocessed and converted to normalized real value. After that, the time series data is represented by a set of symbolic words.

The second step is in charge of randomly selecting a set of shapelet candidates. In [19], the authors have validated that random sampling is an effective technique which can greatly reduce the running time by 3~4 orders of magnitude than the Fast Shapelet (FS) algorithm, but without loss of accuracy. Different with [19], we combine the random sampling with SAX presentation which can further improve the scalability of the algorithm. The third step is merging shapelets extracted from the instances of different classes in the same dimension. During this step, part of self-similar shapelets will be removed to reduce the redundancy of the features. Then, the time series data are transformed to the new feature space. We should calculate the distance between shapelets and all series in data sets. In the fifth step, we adopt classical feature selection algorithm to reduce the dimension of new data sets, i.e., the matrix. Finally, we train a set of random forest classifiers for each dimension, which will be used to adjudge whether a network traffic is an intrusion attack or not.

The pseudo-code of the RSFID algorithm is given Algorithm 1. It is not difficult to obtain the idea of the proposed algorithm. From steps 3 to 9, it is composed of two loops. The first loop is aimed at generating m random forest classifiers, i.e., each forest corresponds to a dimension of the time series (a.k.a. network traffic data). For prediction of a new time series, the label is decided by the voting of all classifiers. The inner loop is for generation of p decision trees for the forest. There are two key steps in the inner loop. The function shapelet_sampling is to randomly sample r shapelets from the j-th dimension of the data set D', which is represented by the SAX method. Another function random_shapelet_tree is to generate a decision tree based on the obtained shapelets $S_{i,j}$ and D'. Next, we will explain the two functions in detail.

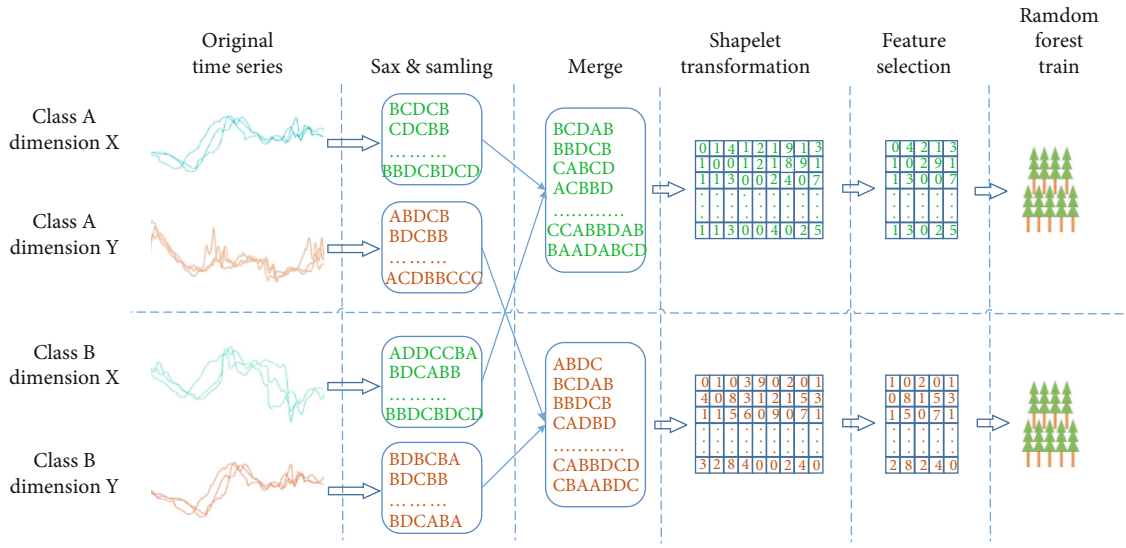

Figure 1: Description of the RSFID algorithm.

3.2. Shapelet Sampling. Since exhaustive search leads to exponential growth of training time, researchers tested the random sampling technique and the results show that it can reduce the running time by 3~4 orders of magnitude than the exhaustive search, without loss of accuracy [19]. However, the existing work does not consider the redundancy and diversity of the sampled shapelets. In this section, we firstly introduce definitions of self-similarity and utility, which are used to filter out nonsimilar shapelets with strong power of discrimination. Then, we explain the code of shapelet_sampling(D', j, r).

Definition 8 (Self-similarity) [23]. Given two subsequences of time series S_1 and S_2, let id_1 and id_2 be the index number of time series that we extract S_1, S_2 from, and $\mathrm{pos}_1, \mathrm{pos}_2$ and $\mathrm{len}_1, \mathrm{len}_2$ denote the start position and the length of S_1, S_2, respectively. We say S_1 and S_2 have self-similarity, when $\mathrm{id}_1 = \mathrm{id}_2 \wedge |\mathrm{pos}_1 - \mathrm{pos}_2| \le \sigma \wedge |\mathrm{len}_1 - \mathrm{len}_2| \le \lambda$.

Here, symbols σ and λ are two user-defined threshold. The former denotes the allowed distance between the starting positions of two shapelets, and the latter represents the allowed difference of two shapelet lengths. Next, we give the definition of utility.

Definition 9 (Utility). Given a shapelet candidate $\theta = \langle S, \tau \rangle$, c denotes the label of the instance that we extract θ from, $C(\cdot)$ is a function that returns the label of an instance. We denote the precision, recall, and utility as follows:

$$P(\theta) = \frac{\left\|\mathrm{dist}_\beta(S, T) \le \tau \wedge c = C(T)\right\|}{\left\|\mathrm{dist}_\beta(S, T) \le \tau\right\|}, T \in D,$$

$$R(\theta) = \frac{\left\|\mathrm{dist}_\beta(S, T) \le \tau \wedge c = C(T)\right\|}{\|c = C(T)\|}, T \in D, \quad (3)$$

$$\mathrm{Utility}(\theta) = \frac{2P(\theta)R(\theta)}{P(\theta) + R(\theta)}.$$

It is easy to find that utility is, essentially, the f-score integrated with precision value and recall value which is regarded as the quality score of a shapelet candidate. Next, we show the pseudo-code in Algorithm 2. In step 2, the algorithm refines the data set of time series that only keeps the j-th dimension of D. From steps 3 to 8, the algorithm randomly extracts a subsequence of a time series and generates a shapelet candidate θ. If the θ is self-similar with any candidates in Θ, it would discard it and resample a new one; otherwise, the θ would be added into the shapelet set Θ. The extraction will be repeated for $r \times \kappa$ times where κ is a coefficient for controlling the total number of shapelet candidates for evaluation. After that, we sort the shapelet candidates in Θ by their utility; then, we keep the top r best shapelets as the final choice.

3.3. Random Shapelet Tree Generation. The pseudo-code of the function random_shapelet_tree is shown in Algorithm 3. It is aimed at generating a decision tree based on a set of shapelets. The algorithm is a typical recursive algorithm which is usually adopted in tree generation. In the third step, the function bestShapelet is to find the best shapelet from Θ which has the highest information gain. If two or more shapelets have the same gain, we choose the one that maximizes the separation gap [16]. After that, we remove the selected shapelet from Θ in step 4. The function distribute is used to separate the instances in D into two groups, those with a distance $\mathrm{dist}_\beta(S, T_i) \le \tau$ and those with a distance $\mathrm{dist}_\beta(S, T_i) > \tau$. Then, we invoke random_shapelet_tree to generate the left subtree and right subtree based on D_L and D_R, respectively. Finally, the function makeLeaf returns a representation of a leaf in the generated tree by simply assigning the class label that occurs most frequently among the instances reaching the node, dealing with ties by selecting a label at random according to a uniform distribution.

3.4. Time Complexity Analysis. Since the time complexity of applying SAX to represent the original time series data is far

Input: D: a data set of time series; p: the number of trees in forest; r: the number of shapelet for each tree
Output: $\Omega = \{F_1, F_2, \cdots, F_m\}$: a set of random forests and each for one dimension.
1 $\Omega \longleftarrow \varnothing$;
2 $D' \longleftarrow \mathrm{SAX}(D)$;
3 for $j = 1$ to m do
4 $\quad F_j \longleftarrow \varnothing$;
5 $\quad$ for $i = 1$ to p do
6 $\quad\quad \Theta_{i,j} \longleftarrow$ shapelet_sampling(D', j, r);
7 $\quad\quad ST_{i,j} \longleftarrow$ random_shapelet_tree$(D', \Theta_{i,j})$;
8 $\quad\quad F_j \longleftarrow F_j \cup ST_{i,j}$;
9 $\quad \Omega \longleftarrow \Omega \cup F_j$;
10 return Ω;

Algorithm 1: RSFID (Random Shapelet Forest for Intrusion Detection).

Input: D: the data set of time series; j: The dimension of the time series; r: the number of shapelet for each tree
Output: Θ: a set of shapelets
1 $\Theta \longleftarrow \varnothing$;
2 $D' \longleftarrow$ refine(D, j);
3 for $i = 1$ to $r \times \kappa$ do
4 $\quad$ id $\longleftarrow$ rand $(1, |D'|)$, $l \longleftarrow$ rand $(3, \mathrm{len}(T_{\mathrm{id}}) - 3)$, pos $\longleftarrow$ rand $(1, \mathrm{len}(T_{\mathrm{id}}) - l + 1)$;
5 $\quad \theta \longleftarrow$ generateShapelet$(D'$, id, l, pos$)$;
6 $\quad$ if self_similar$(\theta, \Theta) =$ true do
7 $\quad\quad i \longleftarrow i - 1$; continue;
8 $\quad \Theta \longleftarrow \Theta \cup \theta$;
9 sort_by_utility(Θ);
10 $\Theta \longleftarrow$ select_best_top_$r(\Theta)$;
11 return Θ;

Algorithm 2: Shapelet_sampling ().

Input: D: the data set of time series; Θ: a set of shapelets; r: the number of shapelets
Output: ST: a shapelet tree
1 if isTerminal(D) do
2 $\quad$ return makeLeaf(D);
3 $\theta \longleftarrow$ bestShapelet(D, Θ);
4 $\Theta \longleftarrow \Theta/\theta$;
5 $(D_L, D_R) \longleftarrow$ distribute(D, θ) ;
6 $ST_L \longleftarrow$ random_shapelet_tree(D_L, Θ)
7 $ST_R \longleftarrow$ random_shapelet_tree(D_R, Θ)
8 Return (ST_L, ST_R);

Algorithm 3: Random_shapelet_tree ().

less than the generation of random shapelet forest, we only discuss the latter part. In Algorithm 2, the function generateShapelet requires to find the best split of a subsequence whose worst time complexity is $O(nl^2)$ where n is the number of instances in data set and l is the length of time series. Besides, the time complexity of the function self_similar is $O(r^2)$ which is far less than $O(nl^2)$. Therefore, the time complexity of shapelet_sampling is $O(r\kappa nl^2)$. In Algorithm 3, the function random_shapelet_tree requires to select the shapelet that has the highest information gain whose time complexity is $O(rn^2l^2)$. Then, it recursively builds the left subtree and the right subtree. The worst case is that the data set is separated into two subsets with equal size in each

Table 1: Description of UNIT.

	Normal	Attack	S-effect	Total
#training instance	962	198950	768	200680
#testing instance	942	198047	522	199511

Table 2: Description of KDD CUP 99.

	Normal	Probing	DoS	U2R	R2L	Total
#training instance	10000	4107	5467	52	1126	20752
#testing instance	60593	4166	229853	228	16189	3111029

Table 3: The precision and recall values of five algorithms on UNIT (%).

	1NN+DTW		NS		ST+CART		ST+SVM		IDRSF	
	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
Attack	99.9	87.7	99.9	81.7	100.0	89.3	100.0	89.4	100.0	92.4
S-effect	31.6	77.2	46.7	75.1	64.7	86.6	65.3	87.7	87.4	98.1
Normal	3.7	92.5	2.0	76	3.7	86.8	4.1	94.9	5.7	96.4

Table 4: The precision and recall values of five algorithms on KDD CUP 99 (%).

	1NN+DTW		NS		ST+CART		ST+SVM		IDRSF	
	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
Probing	82.7	75.2	69.5	71.4	81.7	83.2	91.1	80.4	90.9	87.5
DoS	91.9	86.3	88.2	84.0	98.5	90.2	98.9	94.6	99.9	97.6
U2R	7.5	7.1	11.1	4.2	18.3	15.1	29.6	14.4	48.5	14.0
R2L	26.2	15.9	27.7	3.8	48.8	12.8	65.3	12.2	77.5	13.2
Normal	59.2	87.4	46.2	75.8	63.3	92.5	76.0	92.3	75.7	99.4

iteration. Thus, the complexity is as follows:

$$O\left(rn^2l^2 + (r-1)\left(\frac{n}{2}\right)^2 l^2 + (r-2)\left(\frac{n}{2}\right)^2 l^2 + (r-3)\left(\frac{n}{4}\right)^2 l^2 + \cdots\right) \approx O(rn^2l^2). \tag{4}$$

Obviously, $O(r\kappa nl^2)$ is less than $O(rn^2l^2)$; hence, the overall time complexity of the IDRSF algorithm is $O(rpmn^2l^2)$. Recall that, the symbols r, p, and m represent the number of sampled shapelets, the number of trees in forest, and the number of dimensions in time series, respectively, and it is not difficult to finger out that its time complexity is far less than the time complexity of classical shapelet algorithm, i.e., $O(mn^2l^4)$.

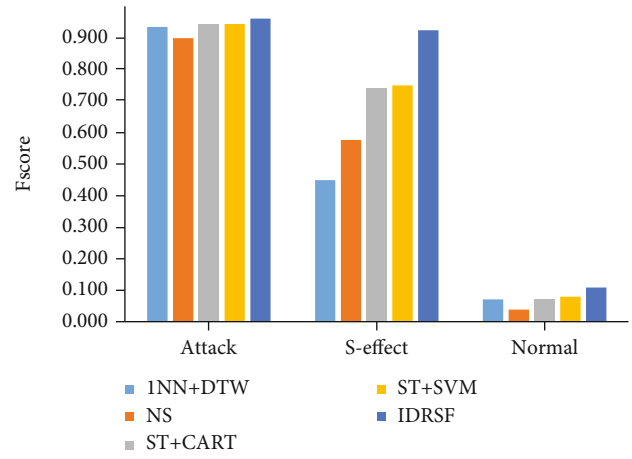

Figure 2: The f-score of five algorithms on UNIT.

4. Experiments

4.1. Data Sets and Parameter Setting. The data sets in the experiments include UNIT [24] and KDD CUP 99 [25], both of which are usually adopted in the field of network security. The UNIT data set includes 14 million records of network attack flows. The collected instances are divided into three groups, which are malicious traffic (attact), side-effect traffic (S-effect), and unknown traffic (normal). Because the size of the UNIT data set is too large to be handled, and meanwhile it lacks normal network traffic, Winter et al. [26] sampled part of instances according to the distribution of the original data set and supplement 1904 instances of normal network traffic. In this paper, we adopt Winter et al.'s data set. KDD CUP 99 is a famous data set for intrusion detection which has 5 million instances of network traffic. There are four different types of network attack in KDD CUP 99, which are labeled as Probing, DoS, U2R, and R2L, respectively. We also sampled 10 percent instances of the original data set and obtained a training data set with 494021 instances and a testing data set with 311029 instances. The details of UNIT and KDD CUP 99 data sets are given in Tables 1 and 2, respectively. Additionally, both data sets were preprocessed, including transform of nominal value to integer value and z-normalization.

The experiments were performed on a PC with Intel Core i7-8700 3.2 GHz CPU and 32 GB RAM. In the proposed algorithm, there are five parameters. We performed cross-validation to decide the parameter settings. The

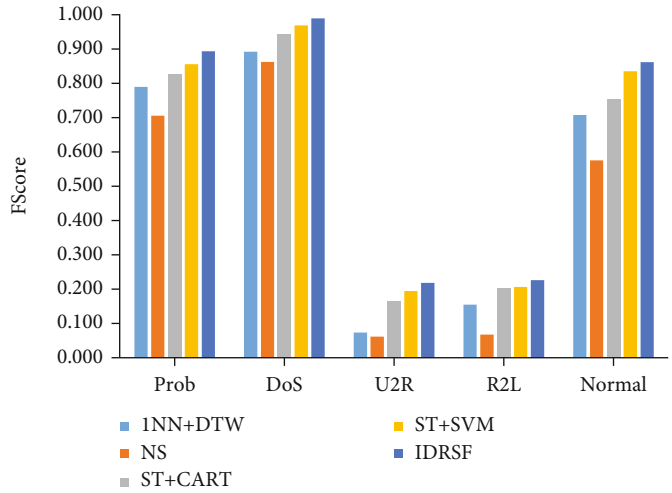

Figure 3: The f-score of five algorithms on KDD CUP 99.

number of shapelets sampled for each forest r is set to $|C| \times \sqrt{m}$, the number of trees p in forest is set to 500, the sampling coefficient κ is set to 1.2, and the two parameters, i.e., δ and λ, for self-similarity detection are set to 2 and 5, respectively.

4.2. Experimental Results and Analysis. To evaluate the effectiveness of the IDRSF algorithm, we choose four algorithms for comparison in the experiments. The first is a classical algorithm, named as 1NN+DTW, which employs one-nearest-neighbor classifier and dynamic time warping. Wang et al. [27] proved that the 1NN+DTW is a classic algorithm for time series classification which is hard to be defeated. Except 1NN+DTW, other three algorithms are all based on shapelet technique, including Naïve Shapelet (NS), Shapelet Transform-based CART (ST-CART) algorithm, and Shapelet Transform-based SVM (ST-SVM).

Additionally, we employed three metrics to evaluate the effectiveness, which are recall, precision, and f-score. It is well known that there are four possible results when predicting a new instance, i.e., true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP and TN refer to the correct prediction of normal behavior and attack behavior. FP and FN refer to the incorrect prediction. Then, the formulas of the three metrics are given below.

$$\begin{aligned} \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{f-score} &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \end{aligned} \tag{5}$$

The precision and recall value of five algorithms on two data sets are given in Tables 3 and 4, respectively. In each table, the experimental results are listed according to the class label. We can find in Table 3 that all the precision values of five algorithms on class "normal" are very low, just from 2% to 5.7%. The rationale behind the result is the unbalance of the UNIT data set. The number of instances in class "normal" is only several hundreds, but tens of thousands of "attack" instances are assigned the label "normal" in the prediction. This dramatically decreases the precision value. The same phenomenon appears in Table 4, e.g., the precision value and the recall value on class "U2L."

For more intuitive comparison, the f-scores obtained by the five algorithms on two data sets are shown in Figures 2 and 3. From the two tables and the two figures, it is not difficult to find that the precision value and the recall value obtained by the IDRSF algorithm on two data sets are obviously better than other four algorithms. Moreover, we can see that the IDRSF algorithm usually performs better on the recall metric, and this is very important for an IDS system. Furthermore, we compare the results of the IDRSF algorithm with that of the state-of-art algorithm (named DSSVM) reported in [28], and we can find that the IDRSF is superior to the DSSVM algorithm. This proves the effectiveness of the proposed algorithm.

However, we cannot ignore that the precision and recall values obtained by the IDRSF algorithm on classes "U2L" and "R2L" are not satisfactory. The reason behind the results is that the testing instances of the two classes include lots of "new patterns" which not appears in the training data set. The shapelet-based technique is essentially a pattern-based method; therefore, it is not easy for the IDRSF to deal with this problem.

5. Conclusions

In this paper, we propose a novel algorithm, named IDRSF, to handle the intrusion detection problem. The algorithm is based on a new primitive "shapelet" in the field of TSC. The advantages of this technique not only include the better ability of classification than traditional techniques in TSC, but also have good interpretability which is not provided by the deep learning methods. The IDRSF algorithm is evaluated on two famous data sets of intrusion detection, i.e., UNIT and KDD CUP 99, and it is compared with four classical algorithms in the field of TSC in which three are based

on shapelet. Experimental results prove the effectiveness of the proposed algorithm. Next, we will try to extend this technique to further handle the unbalance data set and new patterns which not appear in the training set.

Data Availability

The data sets in the experiments include UNIT and KDD CUP 99, both of which are usually adopted in the field of network security [24, 25].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank the anonymous reviewers for their constructive comments, which help the paper quality improved. This work is supported by the National Natural Science Foundation of China (Grant No. 61472050) and the State Key Program of National Natural Science Foundation of China (Grant No. 61332001).

References

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

[2] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 227–261, 2000.

[3] N. Gao, D. Feng, and J. Xiang, "A data mining based dos detection technique," *Chinese Journal of Computers*, vol. 29, no. 6, pp. 944–951, 2006.

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

[5] M. Shafiq, Z. Tian, A. Bashir, X. du, and M. Guizani, "IoT malicious traffic identification using wrapper-based feature selection mechanisms," *Computers & Security*, vol. 94, p. 101863, 2020.

[6] F. Jiang, Y. Sui, and C. Cao, "An incremental decision tree based on rough sets and its application in intrusion detection," *Artificial Intelligence Review*, vol. 40, no. 4, pp. 517–530, 2013.

[7] S. S. Sivatha Sindhu, S. Geetha, and A. Kannan, "A decision tree based light weight intrusion detection," *Expert Systems with Applications*, vol. 39, no. 1, pp. 129–141, 2012.

[8] L. Khan, M. Awad, and B. M. Thuraisingham, "A new intrusion detection system using support vector machine and hierarchical clustering," *The VLDB Journal*, vol. 16, no. 4, pp. 507–521, 2007.

[9] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2021.

[10] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2020.

[11] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, and Z. Tian, "A novel web attack detection system for internet of things via ensemble classification," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5810–5818, 2021.

[12] S. C. Chin, A. Ray, and V. Rajagopalan, "Symbolic time series analysis for anomaly detection: a comparative evaluation," *Signal Processing*, vol. 85, no. 9, pp. 1859–1868, 2005.

[13] L. Wei, N. Kumar, V. N. Lolla, and E. J. Keogh, "Assumption-free anomaly detection in time series," in *Proceedings of the 17th SSDBM*, pp. 237–240, Santa Barbara, CA, USA, 2005.

[14] Y. Kim, J. Sa, S. Kim, and S. Lee, "Shapelets-Based Intrusion Detection for Protection Traffic Flooding Attacks," in *Database Systems for Advanced Applications*, pp. 227–238, Springer, 2018.

[15] L. Ye and E. Keogh, "Time series shapelets: a new primitive for data mining," in *Proceedings of the 15th SIGKDD*, pp. 947–956, Paris, France, 2009.

[16] A. Mueen, E. J. Keogh, and N. E. Young, "Logical-shapelets: an expressive primitive for time series classification," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1154–1162, San Diego, CA, USA, 2011.

[17] E. J. Keogh and T. Rakthanmanon, "Fast shapelets: a scalable algorithm for discovering time series shapelets," in *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 668–676, Austin, Texas, USA, 2013.

[18] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, "Classification of time series by shapelet transformation," *Data Mining and Knowledge Discovery*, vol. 28, no. 4, pp. 851–881, 2014.

[19] I. Karlsson, P. Papapetrou, and H. Boström, "Generalized random shapelet forests," *Data Mining and Knowledge Discovery*, vol. 30, no. 5, pp. 1053–1085, 2016.

[20] J. Grabocka, M. Wistuba, and L. Schmidt-Thieme, "Fast classification of univariate and multivariate time series through shapelet discovery," *Knowledge and Information Systems*, vol. 49, no. 2, pp. 429–454, 2016.

[21] Z. C. Fang, P. Wang, and W. Wang, "Efficient learning interpretable shapelets for accurate time series classification," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 497–508, Paris, France, 2018.

[22] J. Lin, E. J. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[23] W. H. Yan, G. L. Li, Z. D. Wu, S. Wang, and P. S. Yu, "Extracting diverse-shapelets for early classification on time series," *World Wide Web*, vol. 23, no. 6, pp. 3055–3081, 2020.

[24] M. Sheikhan and Z. Jadidi, "Flow-based anomaly detection in high speed links using modified GSA-optimized neural network," *Neural Computing and Applications*, vol. 24, no. 3-4, pp. 599–611, 2014.

[25] K. Siddique, Z. Akhtar, F. Aslam Khan, and Y. Kim, "KDD Cup 99 data sets: a perspective on the role of data sets in network intrusion detection research," *Computer*, vol. 52, no. 2, pp. 41–51, 2019.

[26] P. Winter, E. Hermann, and M. Zeilinger, "Inductive intrusion detection in flow-based network data using one-class vector support machine," in *2011 4th IFIP International Conference*

on New Technologies, Mobility and Security, pp. 1–5, Paris, France, 2011.

[27] X. Y. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2013.

[28] C. Guo, Y. J. Zhou, Y. Ping, Z. Zhang, G. Liu, and Y. Yang, "A distance sum-based hybrid method for intrusion detection," *Applied Intelligence*, vol. 40, no. 1, pp. 178–188, 2014.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5545648, 10 pages
https://doi.org/10.1155/2021/5545648

Research Article

A Privacy Protection Scheme for IoT Big Data Based on Time and Frequency Limitation

Lei Zhang[1,2,3] **Yu Huo**[1,3] **Qiang Ge**[2,3] **Yuxiang Ma**[1,3] **Qiqi Liu**[3] **and Wenlei Ouyang**[3]

[1]*Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475004, China*
[2]*Institute of Data and Knowledge Engineering, Henan University, Kaifeng 475004, China*
[3]*School of Computer and Information Engineering, Henan University, Kaifeng 475004, China*

Correspondence should be addressed to Yuxiang Ma; y.x.ma@hotmail.com

Received 3 March 2021; Revised 2 April 2021; Accepted 16 June 2021; Published 14 July 2021

Academic Editor: Lihua Yin

Various applications of the Internet of Things assisted by deep learning such as autonomous driving and smart furniture have gradually penetrated people's social life. These applications not only provide people with great convenience but also promote the progress and development of society. However, how to ensure that the important personal privacy information in the big data of the Internet of Things will not be leaked when it is stored and shared on the cloud is a challenging issue. The main challenges include (1) the changes in access rights caused by the flow of manufacturers or company personnel while sharing and (2) the lack of limitation on time and frequency. We propose a data privacy protection scheme based on time and decryption frequency limitation that can be applied in the Internet of Things. Legitimate users can obtain the original data, while users without a homomorphic encryption key can perform operation training on the homomorphic ciphertext. On the one hand, this scheme does not affect the training of the neural network model, on the other hand, it improves the confidentiality of data. Besides that, this scheme introduces a secure two-party agreement to improve security while generating keys. While revoking, each attribute is specified for the validity period in advance. Once the validity period expires, the attribute will be revoked. By using storage lists and setting tokens to limit the number of user accesses, it effectively solves the problem of data leakage that may be caused by multiple accesses in a long time. The theoretical analysis demonstrates that the proposed scheme can not only ensure safety but also improve efficiency.

1. Introduction

The development of emerging computing technologies (e.g., cloud computing) have brought opportunity for various industries, such as hyperspectral remote sensing image algorithms [1, 2], classification algorithms [3], matrix operations under linear systems [4, 5], and data generated by Internet of Things (IoT) devices. If the data in a solution is stored in the cloud or the calculation is outsourced to the cloud, the local storage and calculation pressure will be greatly reduced. Among them, for IoT big data, because IoT devices generate huge amounts of data, the structure of the traditional machine learning model is relatively simple, which can no longer meet the new needs of IoT applications. Thus, deep learning technology has been widely used in IoT applications [6], e.g., smart home [7], smart city [8, 9], and autonomous driving [10].

In the scenario of applying deep learning technology to big data in the IoT, in order to train a neural network, large amounts of data need to be obtained from the IoT devices. For example, crowdsensing systems collect data that comes from sensors embedded on personally owned mobile devices [11]. These data may contain sensitive information of some users. However, IoT networks are becoming more vulnerable to various web attacks [12]. Obviously, once they "share" these IoT data with the same field, they are likely to lose control of this data. If these data containing private information are leaked, and there is a lack of effective protection mechanism

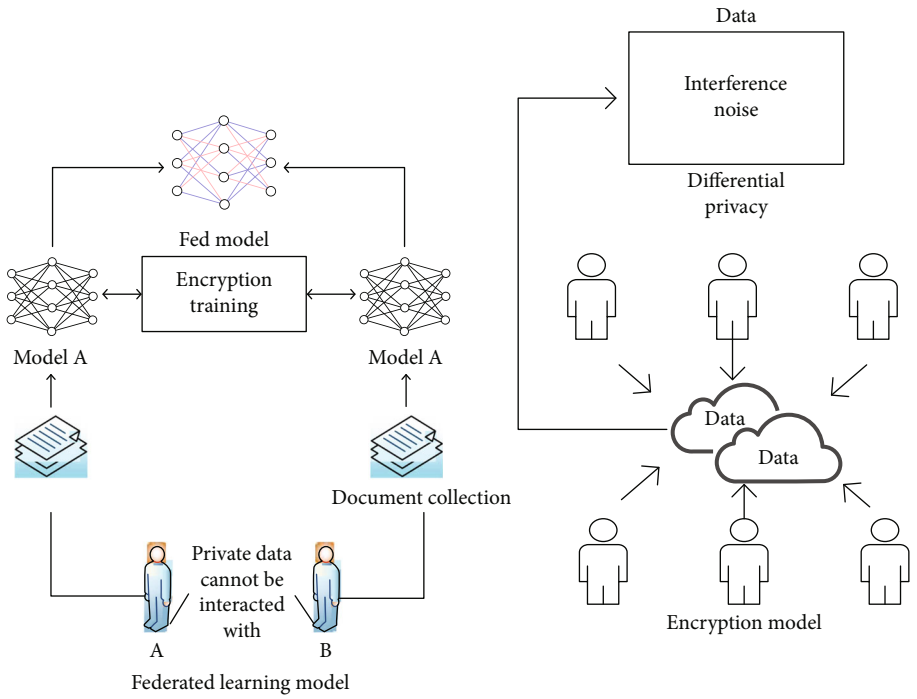

Figure 1: Three different types of working principles.

in the process of IoT search [13], it may cause irreversible harm to the people whose information is leaked. For example, in the field of healthcare, human physiological data collected by wearable IoT devices are put into deep learning models, which can predict the physical condition of patients [14–17]. Once these data are leaked, it will not only cause a patient's economic loss but also endanger life [18]. In the field of autonomous driving, the prediction system of deep learning may be maliciously interfered. Once location privacy data is obtained maliciously, it may cause traffic safety problems and bring troubles to society [19]. It can be seen that how to protect users' private data still faces severe challenges for projects that use deep learning to assist IoT applications, and it is a problem that must be solved.

At present, many solutions have been proposed to solve the big data privacy protection problem in machine learning [20] or deep learning. Generally, these schemes are divided into three categories: federated learning [21, 22], encryption-based technologies [23–26], and differential privacy technologies [27, 28], as shown in Figure 1. Figure 1 shows the working principles of three different types of privacy protection. Among them, encryption-based technologies mainly use direct encryption of data, such as using homomorphic encryption algorithms or setting access control on data uploaded to cloud servers. However, in actual situations, data owners not only want to share training data with others but also want to guarantee data security. Although homomorphic encryption solution realizes the encryption of data, it cannot meet the needs of multiuser data sharing when sharing data in the same field, and it cannot achieve one-to-many fine-grained communication. In attribute-based encryption, only users who meet the access strategy set by the owner can obtain the data, which can achieve more flexible access control. Therefore, to handle the problem of the incompatibility of secure storage and fine-grained sharing of IoT big data in deep learning, an attribute-based encryption solution can be introduced. Among them, the encryption of the ciphertext strategy is more suitable to be used in this scenario than the key-based encryption due to the characteristics of the ciphertext contact access strategy and key contact access structure.

In the actual data sharing scenario, due to the numerous attributes of the visitor, there are many departments in the enterprise engaged in the IoT, so the attribute fluidity is relatively large. Access users obtain the key through their own identity attribute information. If the attribute used to represent the identity does not have a valid period, it means that even if an employee resigns or a department merges, it will not affect the access rights of the resigned employee or the original department staff, and these employees can still obtain data through their own identity attributes. If a resigned employee sells IoT big data in exchange for economic benefits, it will not only endanger the interests of the company but also harm people's personal safety. This shows that it is necessary to set the validity period for each user attribute. The attribute will be cancelled when it expires. Moreover, many current solutions allow users to access unlimited times within the set time. To prevent the number of visits from being abused, it is necessary to limit the number of visits within the set time. By limiting the user's access period and access frequency, to a certain extent, it is possible to reduce the occurrence of data leakage caused by the sale of data information by employees or outsiders using decryption attributes to access big data of the Internet of Things.

We consider the data privacy problems of big data generated in the field of IoT for mobile computing and use attribute revocation idea [29, 30], then propose an IoT big data

privacy protection scheme based on time and the number of decryption restrictions. This scheme combines homomorphic encryption and attribute-based encryption. In summary, the main contributions of this paper are as follows:

(1) We propose a scheme that limits attribute usage time and user decryption frequency. By setting the attribute version number for each attribute as a mark, it is compared with the local time to determine whether the time has expired and realize the revocation. Besides, it limits the number of user accesses by establishing a user decryption frequency table and setting access tokens.

(2) We combine homomorphic encryption with ciphertext-based attribute-based encryption technology, which makes this solution more effective in improving data confidentiality without affecting neural network model training.

(3) We analyse the security of the scheme in a real deployment.

The remainder of the paper is organized as follows. After introducing the related work in Section 2, we provide related technologies used in this paper in Section 3. Section 4 describes the design of our scheme. We analyse security and effectiveness of our scheme in Section 5. Finally, Section 6 concludes this study.

2. Related Work

Although deep learning has brought great convenience to human life, its application is inseparable from data. If some IoT data involves the user's private information, once it is leaked, it will cause property and life safety issues. More and more solutions [31–34] are proposed to solve data security issues, which are implemented by not directly processing data. In addition, people can also protect their privacy by processing data. Lv et al. [35] proposed a secure transaction framework based on the blockchain, which uses the encryption mechanism of the blockchain to ensure information security, but it does not achieve fine-grained access control. Lindell et al. [36] proposed that two parties can process data sets collaboratively without revealing their privacy. Agrawal et al. [37] proposed a scheme that implements the function of outsourcing data to others for data mining tasks. This scheme is confirmed that it does not reveal the data owner's private information during the outsourcing process. Homomorphic encryption technology is considered to be the most effective and most direct means of protecting user privacy [38]. It can directly perform operations, and the results can be consistent with the results of plaintext operations. In 2007, Orlandi et al. [39] introduced homomorphic encryption technology and multiparty secure computing technology to feed the encrypted data into the neural network model for training, which not only ensured the consistency of the plaintext and ciphertext calculation results but also considered security. In [40], the authors proposed a neural network model that uses encrypted data for training. At the same time, in this scheme, it is also proved that cloud services can be used to put encrypted data into the neural network for prediction operations, and the results are returned from the cloud in the form of ciphertext. In [41], the authors improved the scheme [40] and proved that encrypted data can also train neural networks.

In addition to directly encrypting big data, there are also many solutions for setting access control to the data protection layer. In [42], the author created the first CP-ABE solutions, the access policy and ciphertext are sent to the receiver together. Due to the existence of user or attribute revocation problems, research on revocation of ABE has always received extensive attention. Shi et al. [43] proposed a scheme under a hierarchical cryptosystem. Once the attributes are revoked, the public key, private key, and ciphertext of the scheme need to be updated, so the revoking efficiency of this scheme is not high. In [44, 45], the authors pointed out that the private key can be divided into two parts. If the attribute is revoked, the two keys need to be updated, and it is necessary to reencrypt the ciphertext and header files, so the cost of revocation is relatively large. In [46], the authors proposed a user revocation scheme based on a time limit, but it did not achieve fine-grained attribute revocation. In [47], the authors proposed a scheme for using smart contracts to revoke attributes. In addition to these revocation schemes, the purpose of revocation can also be realized by limiting the number of user visits. In [48], the authors proposed a scheme that decryption frequency can be limited. But the function of this scheme is a bit single. While sharing IoT big data that can be used for neural network training, users can adopt a scheme that combines homomorphic encryption and CP-ABE. The solution proposed in [49] has proved that combining the two technologies in such scenarios can not only reduce the risk of data leakage but also reduce the number of key communications. However, in the field of deep learning-assisted IoT applications, there are very few solutions that can combine these technologies to limit user access time and specify the number of user accesses.

3. Preliminaries

3.1. Bilinear Maps. Suppose there is a large prime number p and two cyclic groups G_1 and G_2, their orders are both p, and g is a generator of G_1. Then, there is a mapping $e : G_1 \times G_1 \longrightarrow G_2$ from G_1 to G_2, and it has the following properties [50]:

(1) Bilinearity: $e(g^a, g^b) = e(g^b, g^a) = e(g, g)^{ab}$ for $\forall a, b \in Z_p^*$ and $\forall u, v \in G_1$

(2) Nondegeneracy: there exists $x, y \in G_1$, such that $e(x, y) \neq 1$, where 1 is the identity element of group G_2

(3) Computability: for $\forall u, v \in G_1, e(u, v)$ can be calculated by an effective algorithm.

Then, we call the above mapping e a bilinear mapping. In general, the cyclic group G_1 is an additive cyclic group, and the cyclic group G_2 is a multiplicative cyclic group.

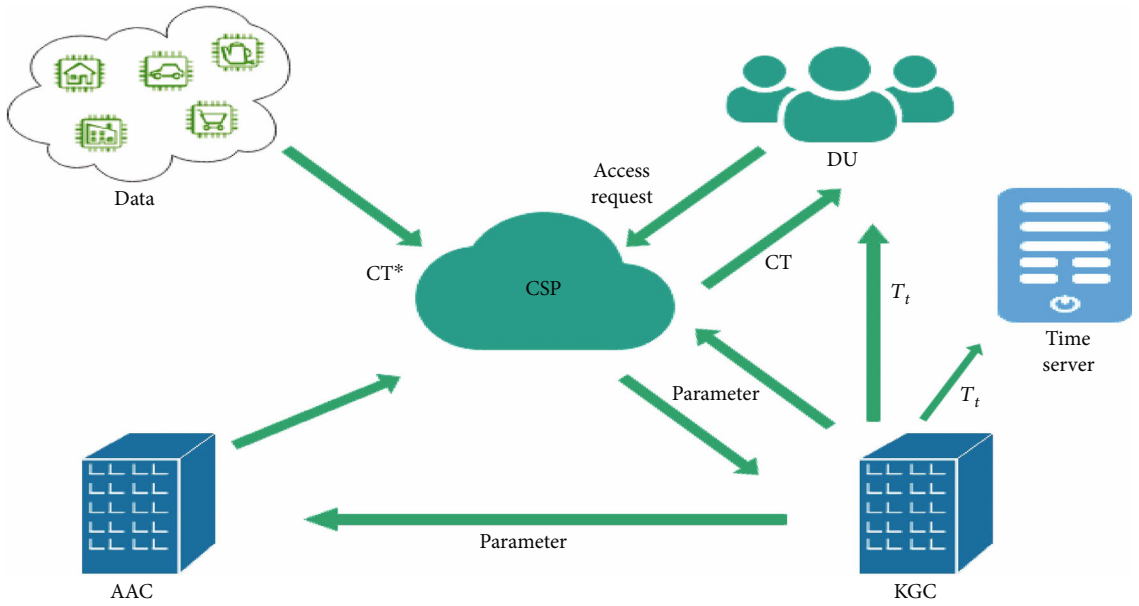

Figure 2: System model.

3.2. Diffie-Hellman Problem. For the additive cyclic group G_1 in the above bilinear map e, there are the following difficult problems in cryptography and discrete mathematics, various cryptosystems based on bilinear mapping are built on the basis of these difficult problems.

Definition 1 (discrete logarithm problem (DL)). If there are any two elements g and Y, $g \in G_1$, $Y \in G_1$, and satisfy $Y = g^k$, where $k \in Z_p^*$, it is difficult to calculate the value of k.

Definition 2 (computational Diffie-Hellman problem (CDH)). Given that a triplet is (g, g^a, g^b), where g is a generator of group G_1, $a, b \in Z_p^*$, it is difficult to calculate the value of g^{ab}.

Definition 3 (decisional Diffie-Hellman problem (DDH)). If there is a four-tuple (g, g^a, g^b, g^c), where g is a generator, $a, b, c \in Z_p^*$, it is difficult to determine whether $c = ab \bmod p$ is true.

Because the above three types of problems are based on group G_1, they are all regarded as group G_1 problems.

3.3. DBDH Assumption. Given that a five-tuple is $[g, g^a, g^b, g^c, \mathcal{Z}]$, where g is a generator of group G_1, $a, b, c \in_R Z_P$, $\mathcal{Z} \in G_2$, it is difficult to determine whether $\mathcal{Z} = e(g, g)^{abc}$ is true.

3.4. Access Structure. The structure is a set of judgment conditions, usually expressed as Γ, which contains several attribute elements in the attribute set A and threshold logic operators (such as OR and AND). If there is an attribute set that satisfies the judgment condition, this attribute set is called an authorized set, otherwise, we called it an unauthorized set. Let $P = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}$ be the entity set of n participants. For $\forall \mathcal{B}, \mathcal{C}$, if $\mathcal{B} \in \mathcal{C}$ and $\mathcal{B} \subseteq \mathcal{C}$, there is $\mathcal{C} \in \mathcal{A}$, then, the set $\mathcal{A} \subseteq 2^{\{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}}$ is monotonous. An access structure is a nonempty subset of $\{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}$, namely, $\mathcal{A} \subseteq 2^{\{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}} \backslash \{\phi\}$. In this proposed solution, the identity information of each user can be described by multiple attributes, such as company, department, and position, which are all his attributes.

3.5. Secure Two-Party Computing Protocol. A secure two-party computing protocol [51–53] means that in a network environment with a low safety factor, two participants can obtain the value of a function after collaborative calculation. Then, they can also obtain the desired value from each other according to this agreement. However, apart from knowing the value of oneself, other information cannot be derived. Through this agreement, it can be ensured that the privacy of the participants themselves will not be leaked when they do not trust each other, which improves program security.

3.6. Homomorphic Encryption. Definition $E(k, a)$ means using an encryption algorithm to encrypt a, the key is k, and F means a certain algorithm of homomorphic encryption, if there is an effective algorithm I, it can be satisfied: $E(k, F(a_1, a_2, \cdots, a_n)) = I(k, F(E(a_1), E(a_2), \cdots, E(a_n)))$. It means that E is homomorphic to F.

4. The Proposed System

4.1. System Solution. In our proposed solution, there exist six types of entities: IoT device, cloud server, data user, attribute authorization centre, key generation centre, and time server. The scheme model is shown in Figure 2.

From Figure 2, we can know that the data owner can encrypt all kinds of data from IoT devices and upload the data to CSP. The access user makes an access request to the cloud server. Legitimate users can download document set

from the cloud server and decrypt it. CSP and KGC jointly generate keys for users through continuous interaction. The time server is responsible for detecting whether the time sent to it by other entities has expired or has been forged or tampered with.

4.2. System Algorithms. We let group $\mathbb{G}$ be a bilinear group, let g be a generator in group $\mathbb{G}$. Let $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_1$ be a bilinear mapping. We choose three hash functions in this scheme: $H : \{0,1\}^* \longrightarrow \mathbb{G}$, so that each attribute can be mapped to the group, $H_1 : \mathbb{G}_1 \longrightarrow Z_P^*$, and $H_2 : \{0,1\}^* \longrightarrow \{0,1\}^*$. In addition, for any $i \in Z_P^*$, an attribute set $\mathbb{A}$, the Lagrangian coefficient is defined as $\Delta_{i,\mathbb{A}}(x) = \prod_{j\in\mathbb{A}, j\neq i}(x - j)/(i - j)$.

(1) Setup$(\lambda) \longrightarrow (\mathrm{PK_{KGC}}, \mathrm{MK_{KGC}}), (\mathrm{PK_{CSP}}, \mathrm{MK_{CSP}}), (\mathrm{PK_{sign}}, \mathrm{MK_{sign}})$. First, the security parameter λ is used to generate three pairs of public and private keys, which are the key generation centre's key pair $\mathrm{PK_{KGC}}, \mathrm{MK_{KGC}}$, the cloud server's key pair $\mathrm{PK_{CSP}}$, $\mathrm{MK_{CSP}}$, and the public and private key pair for digital signature $\mathrm{PK_{sign}}, \mathrm{MK_{sign}}$. KGC randomly selects $\beta \in_R Z_P^*$ and sets $h = g^\beta$, so $(\mathrm{PK_{KGC}} = h, \mathrm{MK_{KGC}} = \beta)$. At the same time, KGC also selects a random number $\gamma \in_R Z_P^*$, so that the public and private key pair used for digital signature is $(\mathrm{PK_{sign}} = g^\gamma, \mathrm{MK_{sign}} = \gamma)$. CSP randomly selects $\alpha \in_R Z_P^*$, it sets $(\mathrm{PK_{CSP}} = e(g,g)^\alpha, \mathrm{MK_{CSP}} = g^\alpha)$. Second, CSP allocates initialization information other than public and private keys for users accessing IoT data, including setting the unique identity of the ith user as u_i, where $u_i \in Z_P^*$. A list L is stored in the cloud server, which contains the user's unique mark u_i, the number of user visits σ, and the state-related mark K_c. Third, KGC selects a random secret value $r_j \in Z_P^*$ for the user, and AAC selects a mark $v_i \in Z_P^*$ for each attribute. Therefore, the system public key is $\mathrm{PK} = \{G, g, h, f = g^{1/\beta}, e(g,g)^\alpha\}$, and the master key is $\mathrm{MK} = (\alpha, \beta)$. The initial value of σ is set to 0.

(2) KeyGen$(\mathrm{PK}, \mathrm{MK}, \mathrm{MK_{sign}}, \mathscr{A}, U_i, \mathrm{st}) \longrightarrow (\mathrm{SK}_{u_t})$. In this part, the digital signature private key $\mathrm{MK_{sign}}$, the user's attribute set $\mathscr{A}$, and the attribute version key U_i, and outputs the user's decryption key. The following four parts are included:

(a) Generate attribute version key. This part is executed by AAC. AAC randomly selects any value $t_i \in Z_P^*$ for each attribute, and t_i is used as a parameter for subsequent use, so the attribute version key is set to $U_i = v_i t_i$, and the attribute version key is generated and sent to CSP.

(b) Generate partial user keys. This part is formed by the simultaneous operation of KGC and CSP via introducing a secure two-party computing protocol. First, KGC takes the parameters (r_j, β) as input, and CSP takes the parameter α as input. Through calculation, $x = (\alpha + r_j)\beta$ is obtained, and the result is output to CSP. CSP selects a random number $\delta \in Z_P^*$, calculates $A = g^{x\delta} = g^{(\alpha+r_j)\beta\delta}$, and sends the calculation result to KGC. When KGC receives the result, calculate $B = A^{1/\beta^2} = \left(g^{(\alpha+r_j)\beta\delta}\right)^{1/\beta^2} = g^{(\alpha+r_j)\delta/\beta}$, and finally, the result B is sent to CSP. CSP calculates $\mathrm{SK_C} = B^{1/\delta} = g^{(\alpha+r_j)/\beta}$ from the received result B. KGC inputs the set attribute version key and outputs partial user's private key $(\mathrm{SK}_k = (\forall\lambda \in \mathscr{A}, D_\lambda = g^{r_j} H(\lambda)^{U_i}, D_\lambda^* = g^{U_i}))$. The partial user's decryption key is composed of a combination of the private key generated by CSP and KGC: $\mathrm{SK} = (\mathrm{SK_C}, \mathrm{SK}_k) = (D = g^{(\alpha+r_j)/\beta}, \forall\lambda \in \mathscr{A}, D_\lambda = g^{r_j} \cdot H(\lambda)^{U_i}, D_\lambda^* = g^{U_i})$

(c) In this part of the algorithm, $K_p = g^{1/(H_2(st)+u_i)}$, $K_c = E^{1/(H_2(st)+u_i)}, K_c$ is the output value of the algorithm VRF [54], K_p, K_c refer to the calculation and detection scheme of the algorithm VRF. Therefore, the final decryption key is $\mathrm{SK}_{u_t} = \{\mathrm{SK}, \mathrm{st}, \mathrm{K}_c, \mathrm{K}_p\} = \{D = g^{(\alpha+r_j)/\beta}, \forall\lambda \in \mathscr{A}, D_\lambda = g^{r_j} \cdot H(\lambda)^{U_i}, D_\lambda^* = g^{U_i}, \mathrm{st}, K_c, K_p\}$, the generated decryption key is sent to the user.

(d) Set the expiration time T_t for each attribute and digitally sign $T_t\xi = g^{1/(H_2(T_t)+\gamma)}$.

(3) HKeyGen (). This algorithm generates the key of a homomorphic encryption algorithm. This scheme uses the DGHV encryption algorithm. In this algorithm, the key is selected as follows: we choose a randomly generated positive prime number as the key p, where $p \in [2^{\eta^2-1}, 2^{\eta^2})$.

(4) Encryption$(\mathrm{PK}, \Gamma, M, p) \longrightarrow (\mathrm{CT}^*)$. This algorithm first inputs the system public key and access policy tree Γ, homomorphic encryption key p, and plaintext message M. Then, this algorithm outputs encrypted ciphertext CT^*. First, the data owner uses the homomorphic encryption key p to encrypt the plaintext M. The specific operation is as follows: they choose two random numbers q, r, where $p \in [2^{\eta^s-1}, 2^{\eta^s}), r \in [2^{\eta-1}, 2^\eta), p/2 > |2r|, q \gg p$. The ciphertext of the document set is calculated by formula $pq + 2r + M$, M is expressed in binary, and the generated ciphertext is uploaded. Second, the data owner encrypts the homomorphic key and uploads the key with attribute access control to the cloud. The data owner regards the attributes as leaf nodes, the root node of the tree is R, and the other nodes are threshold logic operators. The encryption operation performs from the root node and, from top to bottom, produces a linked order for each node, which is d_x polynomial q_x. If n_x is the threshold of nonleaf nodes, then there is a relation $d_x = n_x - 1$. Then, they select a random value s

$\in Z_P^*$, set the polynomial on the root node to $q_R(0) = s$, and use the homomorphic encryption key p to encrypt the plaintext, and use the encryption result to calculate $C = \mathrm{Enc}(p) \cdot e(g,g)^{\alpha s}$, $\widehat{C} = h^s$. Let the polynomial of other nodes be $q_x(0) = q_{p(x)}(\mathrm{index}(x))$, where $\mathrm{index}(x)$ represents the number associated with any node x. The order of nodes is indicated from left to right. In the entire access policy tree, the information carried by each leaf node must be calculated, $C_\lambda = g^{q_\lambda(0)}$, $C_\lambda^* = H(\lambda)^{q_\lambda(0)}$. Then, the final CT* is $\mathrm{CT}^* = \{\Gamma, C = \mathrm{Enc}(p) \cdot e(g,g)^{\alpha s}, \widehat{C} = h^s, \forall \lambda \in \mathcal{J} : C_\lambda = g^{q_\lambda(0)}, C_\lambda^* = H(\lambda)^{q_\lambda(0)}\}$.

(5) $\mathrm{TimeCheck}(\mathrm{SK}_{u_t}, S, T_t, \xi, \mathrm{PK}_{\mathrm{sign}})$. In this part of the algorithm, after the time server receives the validity period of the attribute, it first needs to verify it with digital signature technology to check whether it has been forged or tampered with and verify it with the following calculation method:

$$\begin{aligned} e\left(g^{H_2(T_t)} \cdot \mathrm{PK}_{\mathrm{sign}}, \xi\right) &= e\left(g^{H_2(T_t)} \cdot g^{\gamma}, g^{1/H_2(T_t)+\gamma}\right) \\ &= e\left(g^{H_2(T_t)+\gamma}, g^{1/H_2(T_t)+\gamma}\right) \\ &= e(g,g). \end{aligned} \tag{1}$$

If the verification is successful, it means that the attribute has not been forged or tampered with. The time server compares the validity period T_t with the present time to determine whether the attribute has exceeded the validity period. If it has not expired, you need to continue to execute step 6. If it expires, the attribute needs to be revoked. On the contrary, if the verification fails, it means that the validity period T_t has been maliciously modified, then return $\perp$.

(6) $\mathrm{GenToken}(u_i, \mathrm{st}, \mathrm{ctr}_{\max}) \longrightarrow (B_T)$: after verifying the attribute validity period T_t, it also needs to verify the user's access times, but the difference is that even if a certain attribute fails, the user still has the possibility of access rights, but if the access times exceed the set threshold, then the user does not have the right to access IoT resource data. This algorithm makes the user's unique identity u_i, the user's current state st, and the maximum allowed number of decryption $\mathrm{ctr}_{\max}$ into a token and sends the token to the cloud server.

(7) $\mathrm{Predecryption}(\mathrm{SK}_{u_t}, B_T) \longrightarrow (\mathrm{timeindex})$. In this part, the cloud server first detects $e(g^{H_2(st)} * g^{u_i}, K_p) = E$ and $K_c = e(g, K_p)$ after receiving the token with information. If it meets the verification conditions, CSP will detect the number of decryption $\sigma + 1 \le \mathrm{ctr}_{\max}$ in the list L, if it is satisfied, let $\sigma = \sigma + 1$, update K_c at this time and store it in the list L, and then, the user and CSP continue to perform step 8. Then, let timeindex = 1, otherwise, timeindex = $\perp$. If timeindex = $\perp$, it means accessing users can no longer access IoT big data even if they have access rights.

(8) $\mathrm{Decryption}(\mathrm{SK}_{u_t}, \mathrm{CT}^*) \longrightarrow (p)$. This part of the algorithm is executed by the decryption user and is divided into the following four parts:

(a) When the node x in the access policy tree belongs to the leaf node in the access policy tree, let $i = \mathrm{att}(x)$, it means that the attribute corresponding to the node x computes

$$\begin{aligned} \mathrm{DecryptNode}\left(\mathrm{SK}_{u_t}, \mathrm{CT}^*, x\right) &= \frac{e(D_i, C_x)}{e(D_i^*, C_x^*)} = \frac{e(g,g)^{q_x(0)\left(r_j+U_i\right)}}{e(g,g)^{q_x(0)(U_i)}} \\ &= e(g,g)^{r_j q_x(0)}. \end{aligned} \tag{2}$$

If the attribute is not in the user's attribute set, return $\perp$.

(b) When λ belongs to a nonleaf node in the structure tree, we let S_x be the set of child nodes of each node z of size k_x. When F_z exists and the user's current decryption frequency meet the requirements, then compute

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{(\mathrm{timeindex})\Delta_i, S_x'} = \prod_{z \in S_x} \left(e(g,g)^{r_j q_z(0)}\right)^{\Delta_i, S_x'} \\ &= \prod_{z \in S_x} \left(e(g,g)^{r_j q_{\mathrm{parent}(z)}(\mathrm{index}(z))}\right)^{\Delta_i, S_x'} \\ &= e(g,g)^{r_j q_\lambda(0)}. \end{aligned} \tag{3}$$

If the root node R in this structure tree is replaced by the x node in the above formula, it can be computed as $A = \mathrm{DecryptNode}(\mathrm{SK}_{u_t}, \mathrm{CT}^*, R) = e(g,g)^{r_j s}$.

(c) When the user's attribute set meets the requirements, decryption is performed:

$$\begin{aligned} \mathrm{Dec}\left(\frac{C}{e\left(\widehat{C}, D\right)/A}\right) &= \mathrm{Dec}\left(\frac{e(g,g)^{\alpha s} \cdot \mathrm{Enc}(p)}{e\left(h^s, g^{(\alpha+r_j)/\beta}\right)/e(g,g)^{r_j s}}\right) \\ &= \mathrm{Dec}\left(\frac{e(g,g)^{\alpha s} \cdot \mathrm{Enc}(M)}{e\left(h^s, g^{(\alpha+r_j)/\beta}\right)/e(g,g)^{r_j s}}\right) \\ &= \mathrm{Dec}\left(\frac{e(g,g)^{\alpha s} \cdot \mathrm{Enc}(M)}{e(g,g)^{\alpha s}}\right) = p \end{aligned} \tag{4}$$

(d) After the data visitor obtains the homomorphic key p, users can obtain the document set by using the homomorphic key p.

(9) Revocation(). When the attribute is revoked, this algorithm is executed. In the algorithm, it consists of three parts:

(a) First, KGC randomly selects a reencryption parameter ψ, which is assigned to AAC, CSP, and users whose attributes have been revoked, so that they can update relevant component information in time. Receiving the update information, AAC updates the attribute version keys U_i' of the revoked attributes that it manages, $U_i' = v_i t_i'$.

(b) The next step is to update the user key. CSP obtains the reencryption parameters allocated in the previous step and regenerates the user's latest version key together with KGC. The updated user key is $\mathrm{SK}_{u_t} = \{D = g^{(\alpha+r_j)/\beta}, D_{\lambda'} = g^{r_j} \cdot H(\psi\lambda)^{U_i'}, D_{\lambda'}^* = g^{U_i'}, \forall\lambda \in \mathscr{A} \setminus \{\lambda'\}: D_\lambda = g^{r_j} \cdot H(\lambda)^{U_i}, D_\lambda^* = g^{U_i}, \mathrm{st}, K_c, K_p\}$.

(c) The third step is to update the ciphertext. In this part, CSP first selects a random cipher value $s' \in Z_P^*$ to ensure forward security and then updates the relevant components of the ciphertext after receiving the reencryption parameters. The updated ciphertext is

$$\begin{aligned}\mathrm{CT}^* = \Big\{ & \Gamma, C = e(g,g)^{\alpha(s+s')} \cdot \mathrm{Enc}_k(M), \widehat{C} \\ & = h^{(s+s')}, \forall\lambda \in \mathscr{J} : C_\lambda = g^{q_\lambda(0)+s'}, C_\lambda^* \\ & = H(\psi\lambda)^{q_\lambda(0)+s'} \left(\lambda = \lambda'\right), C_\lambda^* \\ & = H(\lambda)^{q_\lambda(0)+s'} \left(\lambda \neq \lambda'\right) \Big\}\end{aligned} \tag{5}$$

5. Safety and Efficiency Analysis

5.1. Solution Security Analysis

5.1.1. Confidentiality. The confidentiality of this scheme is achieved through two aspects. On the one hand, the attributes of the user must be able to meet the policy set by data owner. If the access policy is not met, then the attributes cannot be used to calculate $e(g,g)^{r_j s}$, so it can prevent unauthorized users from stealing sensitive data. On the other hand, while generating the user's key, to reduce the condition impact of low safety factor and untrustworthy, a secure two-party computing protocol is used to protect the related information of the private key from being obtained by anyone other than itself.

5.1.2. Forward Security. Since each user is set to limit decryption frequency, when users access data, if they meet the requirements of the access policy, they also need to send a token carrying the number of times of decryption to the cloud server. If the number of accesses exceeds the limit, then the user can no longer be decrypted, which ensures forward security.

5.1.3. Collusion Resistance. Users need to use their own attributes to calculate $e(g,g)^{r_j s}$. If users with different permissions want to create a conspiracy attack, then KGC and CSP will generate partial decryption keys through a secure two-party calculation protocol $D = g^{(\alpha+r_j)/\beta}, D_\lambda = g^{r_j} \cdot H(\lambda)^{U_i}, D_\lambda^* = g^{U_i}$, where u_i is a unique random value for each user, so even if the attackers collude, they cannot calculate the value of $e(g,g)^{r_j s}$.

5.1.4. Chosen-Plaintext Attack

Proof. We consider that there exists a polynomial adversary A that is able to break this solution and algorithm B that can overcome the DBDH problem with the advantage of ε.

Initialization: adversary A selects an access structure tree T and sends this access strategy tree to challenger B, and challenger B executes the Setup () initialization algorithm. This part of the process is as follows:

Randomly select four values to calculate $e(g,g)^{ab} e(g,g)^{x} = e(g,g)^{\alpha}$, where $a, b, c, x \epsilon Z_P^*$.

For each attribute $\lambda \epsilon \mathscr{A}$, select a random value $\ell_i \epsilon Z_P^*$, when the attribute does not exist in the access structure tree $\mathbb{T}$, we set $Y_i = H^{1/\ell_i}$, $y_i = b/\ell_i$, if the attribute exists in the access structure tree $\mathbb{T}$, we let $Y_i = g^{\ell_i}$ and $y_i = \ell_i$.

The public key $\mathrm{PK} = \{G, h, g, f = g^{1/\beta}, e(g,g)^{\alpha}\}$ is published, and challenger B keeps the private key $\mathrm{MK} = (\alpha, \beta)$.

Phase 1: after challenger B obtains the public key, adversary A can issue a query request. Adversary A selects an attribute set $s = \{\lambda_i \mid \lambda_i \in \mathbb{T}\}$ and u_i and submits the information to challenger B to apply for a private key. Challenger B randomly selects r_i, U_i generates the corresponding private key. The calculation process is as follows:

$$\mathrm{SK} = \left(D = g^{(ab+x+r_j)/\beta}, D_\lambda = g^{r_j} \cdot H(\lambda)^{U_i}, D_\lambda^* = g^{U_i}\right). \tag{6}$$

If the number of decryptions meets the requirements, st, K_c, K_p will not affect the final decryption effect.

Challenge: adversary A has obtained the access control tree T at this time and then submits two plaintexts of the same length to challenger B. By comparing the attribute sets, if the attribute set sent in the previous step does not meet the structure tree T, then the two plaintexts are set to m_0, m_1, and the two plaintexts are sent to challenger B along with the access strategy tree. Then, B randomly selects $\rho \in \{a, b\}$, calculate:

$$\begin{aligned} C_0 &= \mathrm{Enc}\left(M_\rho\right) \cdot e(g,g)^{\alpha s} = \mathrm{Enc}\left(M_\rho\right) \cdot e(g,g)^{(ab+x)s} \\ &= \mathrm{Enc}\left(M_\rho\right) \cdot e(g,g)^{abs} e(g,g)^{xs} = \mathrm{Enc}\left(M_\rho\right) \\ &\quad \cdot e(g,g)^{abs} \cdot e(g^x, g^s), \\ \widetilde{C_0} &= h^s, \forall\lambda \in \mathscr{J} : \widetilde{C_\lambda} = g^{q_\lambda(0)}, \widetilde{C_\lambda^*} = H(\lambda)^{q_\lambda(0)}. \end{aligned} \tag{7}$$

Challenger B sends this information to A.

Table 1: Functional comparison.

Schemes	Revocability	Time	Number	Collusion	Ciphertext operability
[55]	User and attribute	×	×	✓	×
[46]	User	✓	×	✓	×
[48]	User	×	✓	✓	×
[47]	Attribute	✓	×	✓	×
[49]	None	×	×	✓	✓
[56]	Attribute	×	×	✓	×
Our scheme	User and attribute	✓	✓	✓	✓

Table 2: Cost comparison.

Schemes	Secret key cost	Decryption cost		Revocation	
		User	CSP	Attribute-cost	User-cost
[55]	$3e$	3p	×	$3(n+1)p$	np
[56]	$4e$	×	e	e	×
Our scheme	$O(5n)$	$O(n)$	$O(n)$	$O(5n)$	$O(1)$

Phase 2: A can always ask B for private key-related information, and then, A guesses the ciphertext and needs to give his own guess value $\wp'$.

Guess: if $\wp' = \wp'$, then DBDH is established, the advantage is $p_r[\wp' = \wp' \mid e(g,g)^{abc}] = \varepsilon + 1/2$, if $\wp' \neq \wp'$, the ciphertext cannot be judged, and the advantage is $p_r[\wp' \neq \wp' \mid e(g,g)^{\vartheta}] = 1/2$. In summary, $p_r[(g, g^a, g^b, g^c, e(g,g)^{abc}) = 1] - p_r[(g, g^a, g^b, g^c, e(g,g)^{\vartheta}) = 1] \geq \varepsilon$. It shows that this scheme can realize that no adversary can break the scheme with a nonnegligible advantage in polynomial time.

5.2. Theoretical Comparison. Our scheme is compared with other schemes in terms of revocation mechanism, time limit, number of decryption limits, and anticollusion. The comparison results are shown in Table 1.

From Table 1, it can be seen that in [46–49, 56], the revocation schemes proposed by the authors do not fully meet the revocation needs. Although in [55] the authors proposed a scheme that can support user revocation and attribute revocation, in the scenario we mentioned, it is also a requirement that the ciphertext can be operated. This scheme in [49] realizes that users can operate on ciphertext, but it is not suitable for scenarios where attributes need to be revoked. Our scheme realizes two revocation functions, solves the basic system security problem, and achieves the ciphertext operable function. What is more, we also consider two factors: time and frequency of decryption.

Our scheme is compared with other schemes in terms of key generation efficiency, decryption efficiency, and revocation efficiency. e is the exponential calculation cost, and p is the bilinear pair calculation cost. The comparison results are shown in Table 2.

It can be seen that in [55] only the user performs the decryption operation and in [56] only CSP performs the decryption operation, which will cause one-side pressure. Our scheme can effectively reduce the amount of user tasks by placing part of the decryption task on the cloud server. Also, in [55], while realizing user revocation, the cost is np. However, in our scheme, if the user is revoked after judgment, the user only needs to be removed from the list L, thus, its computational complexity is better than the schemes [55, 56]. Although the cost of generating the key is relatively high due to the use of a two-party security protocol, the security of the key is guaranteed through this multiparty cooperation method.

6. Conclusions

Since important personal privacy may be leaked while storing and sharing IoT big data on the cloud, we have proposed an IoT big data privacy protection scheme based on time and decryption frequency limitation, the solution realizes the revocation within the time range and the revocation within the range of decryption times. The access control is set by the combination of homomorphic encryption and attribute-based encryption. In our scheme, legitimate users with a homomorphic encryption key can obtain the original data, and users without a homomorphic encryption key can perform operation training on the homomorphic ciphertext. Our scheme does not only affect the training of the neural network model but also improves the confidentiality of the data. At the same time, the security of the system is improved by introducing a secure two-party agreement. Through theoretical analysis, we found that our scheme realizes two revocation functions, solves the basic system security problem, and achieves the ciphertext operable function. While realizing user revocation, the computational complexity is preferable to other schemes. Besides, our scheme can effectively reduce the amount of user tasks by placing part of the decryption task on the cloud server. Therefore, our scheme can not only ensure safety but also improve efficiency. In the next step, we plan to combine the advantages of decentralization and anonymity of blockchain to protect big data in the Internet of Things in a distributed storage environment.

Data Availability

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

Conflicts of Interest

There is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China Project (Nos. 61701170 and U1704122), the Key Scientific and Technological Project of Henan Province (Nos. 202102310340 and 202102210352), the Young Elite Scientist Sponsorship Program by Henan Association for Science and Technology (No. 2020HYTP008), the Foundation of University Young Key Teacher of Henan Province (Nos. 2019GGJS040 and 2020GGJS027), and the Key Scientific Research Project of Colleges and Universities in Henan Province (No. 21A110005).

References

[1] W. Huang, Y. Xu, X. Hu, and Z. Wei, "Compressive hyperspectral image reconstruction based on spatial–spectral residual dense network," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 5, pp. 884–888, 2019.

[2] W. Huang, Y. Huang, H. Wang, Y. Liu, and H. J. Shim, "Local binary patterns and superpixel-based multiple kernels for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4550–4563, 2020.

[3] L. Peng, H. Zhang, H. Hassan, Y. Chen, and B. Yang, "Accelerating data gravitation-based classification using GPU," *Journal of Supercomputing*, vol. 75, no. 6, pp. 2930–2949, 2019.

[4] X. Zhang, F. Ding, and E. Yang, "State estimation for bilinear systems through minimizing the covariance matrix of the state estimation errors," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 7, pp. 1157–1173, 2019.

[5] X. Zhang, L. Xu, F. Ding, and T. Hayat, "Combined state and parameter estimation for a bilinear state space system with moving average noise," *Journal of the Franklin Institute*, vol. 355, no. 6, pp. 3079–3103, 2018.

[6] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in Internet of Things with privacy preserving: challenges, solutions and opportunities," *IEEE Network*, vol. 32, no. 6, pp. 144–151, 2018.

[7] E. Park, Y. Cho, J. Han, and S. J. Kwon, "Comprehensive approaches to user acceptance of Internet of Things in a smart home environment," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2342–2350, 2017.

[8] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[9] P. Gope, R. Amin, S. K. Hafizul Islam, N. Kumar, and V. K. Bhalla, "Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment," *Future Generation Computer Systems*, vol. 83, pp. 629–637, 2017.

[10] Y. Tian, K. Pei, S. Jana, and B. Ray, "DeepTest: automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th International Conference on Software Engineering*, pp. 303–314, Gothenburg, Sweden, May 2018.

[11] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, "Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6266–6278, 2020.

[12] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, and Z. Tian, "A novel web attack detection system for Internet of Things via ensemble classification," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5810–5818, 2020.

[13] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, "A survey on access control in the age of internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4682–4696, 2020.

[14] R. Poplin, A. V. Varadarajan, K. Blumer et al., "Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning," *Nature Biomedical Engineering*, vol. 2, no. 3, pp. 158–164, 2018.

[15] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 3, pp. 312–325, 2016.

[16] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: remote sensing image scene classification via learning discriminative CNNs," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 5, pp. 2811–2821, 2018.

[17] A. Rachedi and A. Benslimane, "Multi-objective optimization for security and QoS adaptation in wireless sensor networks," in *2016 IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 2016.

[18] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of IoT new features on security and privacy: new threats, existing solutions, and challenges yet to be solved," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1606–1616, 2018.

[19] M. Amoozadeh, A. Raghuramu, C.-n. Chuah et al., "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 126–132, 2015.

[20] M. Shafiq, Z. Tian, A. K. Bashir, X. du, and M. Guizani, "CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2021.

[21] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2019.

[22] L. Jiang, X. Lou, R. Tan, and J. Zhao, "Differentially private collaborative learning for the IoT edge," *In Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks (EWSN '19)*, Junction Publishing, USA, 2019.

[23] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6532–6542, 2019.

[24] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, "Enabling efficient and geometric range query with access control over encrypted spatial data," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 870–885, 2019.

[25] H. Li, D. Liu, Y. Dai, T. H. Luan, and S. Yu, "Personalized search over encrypted data with efficient and secure updates in mobile clouds," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 97–109, 2018.

[26] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. P. C. Rodrigues, "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4755–4763, 2018.

[27] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and U. Erlingsson, *Scalable Private Learning with PATE*, 2018.

[28] C. Xu, J. Ren, L. She, Y. Zhang, Z. Qin, and K. Ren, "EdgeSanitizer: locally differentially private deep inference at the edge for mobile data analytics," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5140–5151, 2019.

[29] J. Hur, "Improving security and efficiency in attribute-based data sharing," *Transactions On Knowledge And Data Engineering*, vol. 25, no. 10, pp. 2271–2282, 2013.

[30] J. Y. Wang and X. J. Zhou, "An attribute-based encryption scheme for ciphertext policy that supports attribute revocation," *Computer Engineering*, pp. 1–7, 2020.

[31] Y. Sun, Z. Tian, M. Li, S. Su, X. Du, and M. Guizani, "Honeypot identification in softwarized industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5542–5551, 2020.

[32] Y. Pang, L. Peng, Z. Chen, B. Yang, and H. Zhang, "Imbalanced learning based on adaptive weighting and Gaussian function synthesizing with an application on android malware detection," *Information Sciences*, vol. 484, pp. 95–112, 2019.

[33] V. Ravindranath, S. Ramasamy, R. Somula, K. S. Sahoo, and A. H. Gandomi, "Swarm intelligence based feature selection for intrusion and detection system in cloud infrastructure," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–6, Glasgow, UK, July 2020.

[34] D. Xu, J. Pan, X. Du, B. Wang, M. Liu, and Q. Kang, "Massive fishing website URL parallel filtering method," *IEEE Access*, vol. 6, pp. 2378–2388, 2018.

[35] L. Lv, Z. Yang, L. Zhang, Q. Huang, and Z. Tian, "Multi-party transaction framework for drone services based on alliance blockchain in smart cities," *Journal of Information Security and Applications*, vol. 58, no. 4, p. 102792, 2021.

[36] Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, Springer, Berlin, Heidelberg, 2000.

[37] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00*, Dallas, Texas, USA, 2000.

[38] T. Plantard, W. Susilo, and Z. Zhang, "Fully homomorphic encryption using hidden ideal lattice," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 2127–2137, 2013.

[39] C. Orlandi, A. Piva, and M. Barni, "Oblivious neural network computing via homomorphic encryption," *EURASIP Journal on Information Security*, vol. 2007, no. 1, Article ID 037343, 2007.

[40] N. Dowlin, G. B. Ran, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: applying neural networks to encrypted data with high throughput and accuracy," in *International Conference on Machine Learning*, pp. 201–210, NY, USA, 2016.

[41] E. Hesamifard, H. Takabi, and M. Ghasemi, "Privacy-Preserving Machine Learning in Cloud," in *Proceedings of the 2017 on cloud computing security workshop*, pp. 39–43, 2017.

[42] M. Chase, "Multi-authority attribute based encryption," in *Conference on Theory of Cryptography*, Springer-Verlag, 2007.

[43] J. Shi, C. Huang, K. He, and X. Shen, "ACS-HCA: an access control scheme under hierarchical cryptography architecture," *Chinese Journal of Electronics*, vol. 28, no. 1, pp. 52–61, 2019.

[44] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1767–1777, 2017.

[45] S. Wang, K. Guo, and Y. Zhang, "Traceable ciphertext-policy attribute-based encryption scheme with attribute level user revocation for cloud storage," *PLoS One*, vol. 13, no. 9, article e0203225, 2018.

[46] G. Dilxat, S. Y. Han, A. Gulmira, and H. Nurmamat, "Time-based user revocation CP-ABE scheme," *Journal of Xinjiang University(Natural Science Edition)*, vol. 36, no. 3, pp. 324–329, 2019.

[47] X. Qin, Y. Huang, Z. Yang, and X. Li, "An access control scheme with fine-grained time constrained attributes based on smart contract and trapdoor," in *2019 26th International Conference on Telecommunications (ICT)*, Hanoi, Vietnam, April 2019.

[48] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma, and L. Wei, "Auditable σ-time outsourced attribute-based encryption for access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 94–105, 2018.

[49] Y. Tan, L. Lu, and J. Y. Wang, "Ciphertext-policy attribute encryption scheme based on homomorphic encryption," *Computer Engineering and Applications*, vol. 55, no. 19, pp. 115–120, 2019.

[50] J. Li, Y. Wang, Y. Zhang, and J. Han, "Full verifiability for outsourced decryption in attribute based encryption," *IEEE Transactions on Services Computing*, vol. 13, no. 3, pp. 478–487, 2020.

[51] Y. Tang and D. Y. Xu, "A secure two-party computation problem based on the convolution," *Journal of Guizhou University(Natural Ences)*, vol. 33, no. 1, pp. 52–57, 2016.

[52] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security - CCS '09*, pp. 121–130, Chicago, Illinois, USA, 2009.

[53] S. S. M. Chow, "Removing escrow from identity-based encryption," in *International Workshop on Public Key Cryptography*, Springer, Berlin, Heidelberg, 2009.

[54] Y. Dodis and A. Yampolskiy, "A verifiable random function with short proofs and keys," in *Public Key Cryptography - PKC 2005*, Springer, 2005.

[55] X. Li, S. Tang, L. Xu, H. Wang, and J. Chen, "Two-factor data access control with efficient revocation for multi-authority cloud storage systems," *IEEE Access*, vol. 5, pp. 393–405, 2017.

[56] Z. T. Jiang, J. Huang, S. Hu, and Z. Xu, "Fully-outsourcing CP-ABE scheme with revocation in cloud computing," *Computer Science*, vol. 46, no. 7, pp. 114–119, 2019.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5549109, 9 pages
https://doi.org/10.1155/2021/5549109

Research Article

A Model Study on Collaborative Learning and Exploration of RBAC Roles

Jiyong Yang,[1,2] Xiajiong Shen,[1,2] Wan Chen,[1,2] Qiang Ge,[1,2] Lei Zhang,[1,2,3] and HaoLin Chen[1,2]

[1]*Henan Key Laboratory of Big Data Analysis and Processing, Henan University, 475000 Kaifeng, China*
[2]*School of Computer and Information Engineering, Henan University, 475000 Kaifeng, China*
[3]*Institute of Data and Knowledge Engineering Henan University, 475000 Kaifeng, China*

Correspondence should be addressed to Lei Zhang; zhanglei@henu.edu.cn

Received 28 February 2021; Accepted 2 June 2021; Published 25 June 2021

Academic Editor: Lihua Yin

Role-based access control (RBAC) can effectively guarantee the security of user system data. With its good flexibility and security, RBAC occupies a mainstream position in the field of access control. However, the complexity and time-consuming of the role establishment process seriously hinder the development and application of the RBAC model. The introduction of the assistant interactive question answering algorithm based on attribute exploration (semiautomatic heuristic way to build an RBAC system) greatly reduces the complexity of building a role system. However, there are some defects in the auxiliary interactive Q&A algorithm based on attribute exploration. The algorithm is not only unable to support multiperson collaborative work but also difficult to find qualified Q&A experts in practical work. Aiming at the above problems, this paper proposes a model collaborative learning and exploration of RBAC roles under the framework of attribute exploration. In this model, after interactive Q&A with experts in different permissions systems by using attribute exploration, the obtained results are merged and calculated to get the correct role system. This model not only avoids the time-consuming process of role requirement analysis but also provides a feasible scheme for collaborative role discovery in multidepartment permissions.

1. Introduction

With the development of the information system, information sharing among people becomes more and more convenient and fast. However, the "explosive" growth of the information system brings people convenient and quick access to information, and it also brings the problem of information security. It is not only the sharing of information between people that needs to be protected but also the information between industrial systems. For example, when computing matrix in the research field of Kalman filtering, multiple computing contents need to be encrypted [1, 2].

To prevent the intrusion of illegal users or leakage caused by the careless operation of legal users, many solutions have been proposed [3, 4]. For example, Lihua proposes a new privacy protection scheme, which plays a good role in protecting privacy [5]. Access control allows users to access system resources only according to their permissions setting and may not exceed their permissions. To ensure flexibility and security, role-based access control (RBAC) [6] has been widely studied and applied due to its good applicability and occupies a mainstream position in the access control model [7]. The RBAC model introduces roles between users and permissions; connects users and permissions with roles and grants and revokes access permissions to users by assigning and canceling roles to users; and realizes the logical separation of users and access permissions [8]. Flexibility in permission management and its high correlation with an enterprise's organizational structure greatly facilitate permission management [9].

However, the increasing complexity of the information system leads to the increasing complexity of the RBAC model system construction [10]. In the design and use of a traditional RBAC system, the relationship between "users and roles" and

"roles and permissions" is dependent on the acquisition of system requirement information and the personal experience of administrators. With the increasing complexity and diversification of the information system, the number of users, resources, and permissions in access control is increasing, and the business process and related domain knowledge of information systems are becoming complex. As a result, designing and managing an RBAC system that meets the functional and security needs of users solely relying on human beings is challenging [11]. With the development and prosperity of machine learning has given us more ways and methods to solve problems, machine learning is applied in various fields [12]. Many scholars have also applied machine learning to information security, Sun proposes an ESS-based algorithm of balancing the QoS and privacy risk, which reaches a stable state of maintaining long-term service by multiple iterations [13], and Yin uses a recursive neural network for intrusion detection [14]. In addition, machine learning is also applied to various fields, such as hyperspectral image processing and classification [15, 16]. With the prosperity and development of information system, information security combined with many research fields has been widely discussed and studied [17, 18].

Among them, zhang Lei [19] proposed an auxiliary interactive question answering algorithm based on attribute exploration and used the attribute exploration algorithm to interact with experts to get the required roles and the partial order relationship between roles in the RBAC system. The reason why the attribute exploration algorithm [20] can obtain the roles and the partial order relationship between roles is that the attribute exploration algorithm is an important tool in the formal concept analysis [21]. Formal concept analysis is considered as a favorable tool for data analysis and knowledge description and has been widely used in data analysis [22], knowledge discovery [23], rule extraction [24], concept cognitive learning [25], and other fields. Among them, the important data structure-concept lattice [26] can well represent the partial order structure among data. Each lattice node on the concept lattice is composed of a group of intent and extent which have a natural correspondence with roles and permissions in role engineering. The role system mined by the concept lattice theory can not only reflect the hierarchical relationship between the roles but also ensure the correctness of the roles mined [27].

Although the auxiliary interactive question answering algorithm based on attribute exploration can accomplish the role design of RBAC with heuristic assistance, the traditional attribute algorithm relies on the complete system permissions knowledge. In practice, it is difficult to find people who have a good knowledge of all permissions, especially when the permissions are involved in multiple departments. For example, it is difficult to find an expert who knows all the permissions information well when constructing the role of the administration system in conjunction with the faculty system. This defect severely limits the development and application of the RBAC model.

In this paper, it is found that the Duquenne–Guigues and the set of roles obtained by the auxiliary interactive question answering algorithm based on attribute exploration have a close relationship with the whole system, but also have a close relationship with the local subsystem. Therefore, we can find an interactive domain expert in each one and merge the roles and Duquenne–Guigues of each system after the interaction of multiple systems is completed, to obtain the set of the Duquenne–Guigues and roles of the entire system.

Therefore, this paper proposes a model collaborative learning and exploration of RBAC roles (*RCLE*). Under the framework of interactive Q&A of property exploration, a method is designed to support the role discovery of the same group of users under different permission systems. This model not only avoids the time-consuming process of role demand analysis and questionnaire survey in the process of role construction but also avoids the defects of the auxiliary interactive question and answer algorithm of attribute exploration in the construction of role system across departments.

2. Basic Definition

The relevant definitions used in this article are as follows [23, 25, 26]:

Definition 1. An access security context $K=(U, M, I)$ is composed of two sets U, M, and I (the relationship between U and M). The element of U is called user (object), and the element of M is called permission (attribute). $(u, m) \in I$ or uIm means that user u has permission m. We use $(u, m) \notin I$, which means that user u does not have permission m.

Definition 2. Set $K=(U, M, I)$ that is an access security context, if $A \subseteq U, B \subseteq M$, then write

$$f(A) = \{m \in M | \forall u \in A \in \mathrm{A}, (u, m) \in I\}, \tag{1}$$

$$g(B) = \{u \in U | \forall \forall m \in B, (u, m) \in I\}. \tag{2}$$

If A and B satisfy that $f(A) = B$ and $g(B) = A$, then we call the binary group (A, B) a concept. A is the extent of the concept (A, B), and B is the intent of the concept (A, B).

The computation of Definition 2 is carried out throughout the text. Definition 2 shows how to compute concepts in a given access security context. Since more than one formal context will be involved in the following paragraphs, for the convenience of distinguishing, $f_1(A)$ and $g_1(B)$ represent the calculation of $f(A)$ and $g(B)$ on the formal context K_1.

The concept of access security context $K=(U, M, I)$ has the following basic properties ($\forall A, A_1, A_2 \subseteq U, \forall B, B_1, B_2 \subseteq M$):

Property 3. $A_1 \subseteq A_2 \Rightarrow f(A_2) \subseteq f(A_1)$; $B_1 \subseteq B_2 \Rightarrow g(B_2) \subseteq g(B_1)$; $A \subseteq g(f(A))$; $B \subseteq f(g(B))$; if $B=f(g(B))$, then B is intent on the access security context K.

Definition 4. Set (U, M, I) is an access security context, $Y \subseteq M$, and satisfies

(1) $Y \neq f(g(Y))$ $(Y \subseteq f(g(Y)))$,

(2) Each pseudointent $Y_1 \subset Y$ has $f(g(Y_1)) \subseteq Y$. Then, Y is a pseudointent.

Definition 4 provides the conditions for the establishment of pseudointent. To prove whether an attribute set is a pseudointent, we only need to verify whether it meets the two conditions of Theorem 14.

Definition 5. Set $K=(U, M, I)$ is an access security context, Y_1, $Y_2 \subseteq M$. if $g(Y_1) \subseteq g(Y_2)$, then $Y_1 \longrightarrow Y_2$ is true in K.

Definition 6. If $K=(U, M, I)$ is an access security context, then the value dependency set $\{X \longrightarrow f(g(X)) \mid X$ is the pseudointent of $K\}$ which is the Duquenne–Guigues of K.

Definition 7. Given access security context $K=(U, M, I)$, implication set $J(K)$, and implication formula $C \longrightarrow D \in J(K)$, the attribute set if and only if $T \subseteq MC \notin T$ or $D \subseteq T$, called T, is associated with $C \longrightarrow D$. If T is related to all the implication forms in $J(K)$, then T is related to $J(K)$.

According to the value dependence theory of concept lattice, the Duquenne–Guigues can produce all value dependence held in an access security context, namely, the implication relation of an attribute. It can be seen from definition 6 that the Duquenne–Guigues of access security context can be obtained as long as all pseudointents are found. The correlation judgment between attribute set and implication set in Definition 7 can be used in the calculation of pseudointent.

Definition 8. Let $K=(U, M, I)$ be an access security context, $M = \{m_1, m_2 \cdots m_n\}$, and the permission (attribute) in M satisfies the basic linear order relationship $(m_1 < m_2 < \cdot\cdot < m_n)$. For any $Y_1, Y_2 \subseteq M$ if and only if there is $m_i \in Y_2 - Y_1$ and $Y_1 \cap \{m_1, \cdots, m_{i-1}\} = Y_2 \cap \{m_1, \cdots, m_{i-1}\}$, the lexicographical order of attribute set Y_1 is less than the lexicographical order of permission (attribute)set Y_2, denoted as $Y_1 < Y_2$.

Definition 8 describes the lexicographical order relation of the property set < which is a linear order relation of 2^M. All property sets can be generated one by one according to the lexicographical order and tested one by one to see if the property set is a pseudointent or intent.

3. A Model for Collaborative Learning and Exploration of RBAC Roles

The attribute exploration algorithm interacts with domain experts by asking questions, traverses the attribute set in lexicographical order, and tests whether the set is pseudointent or intent. The use is the attribute set of the pseudointent to produce the implication, so as to construct the Duquenne–Guigues of the access security context and obtain the relevant context knowledge. Lexicographical order < is a linear order on the power set of all permission (attribute), which guarantees the completeness of the attribute exploration algorithm. In other words, the set of roles obtained by the traditional role discovery algorithm based on attribute exploration is complete. However, due to the lack of cooperation mechanism, traditional role discovery algorithms based on attribute exploration cannot build a role system across departments.

The key to the above problem is how to discover the set of roles and the implication relationship between permissions under multiple permission systems (Duquenne–Guigues). In this paper, we found that after the attribute exploration among different departments, we further analyzed and summarized the roles and Duquenne–Guigues under different permissions systems, so as to obtain the role construction of the crossdepartment permission system.

3.1. Basic Theorem. To facilitate the elaboration, we first make the following definition.

Definition 9. Given an access security context $K_1 = (U_1, M_1, I_1)$ and $K_2 = (U_2, M_2, I_2)$, $\forall g \in U_1 \cap U_2$, and $\forall b \in M_1 \cap M_2$ that meet $gI_1b \Leftrightarrow gI_2b$ then called K_1 is consistent with K_2.

Definition 10. A model for collaborative learning and exploration of RBAC roles RCLE $= (K_1, K_2, J(K_1), C(K_1), J(K_2), C(K_2))$, $K_1 = (U_1, M_1, I_1)$, $K_2 = (U_2, M_2, I_2)$, $U_1 = U_2 = \{g_1, g_2, g_3, g_4 \cdots\}$, and $M_1 = (a_1, b_1, c_1, d_1 \cdots)$, $M_2 = (a_2, b_2, c_2, d_2 \cdots)I$, represents the relationship between U_i and M. M and I are consistent in K_1 and K_2, $J(K_1)$, $J(K_2)$, $C(K_1)$, and $C(K_2)$, respectively, and represent the Duquenne–Guigues and intent of K_1 and K_2.

Based on the above definition, we have the following findings, which can be used as the theoretical basis of the RCLE model.

Theorem 11. *Given an access security context $K = (U, M, I)$, the Duquenne–Guigues $J(K)$, and implication formula $A \longrightarrow B \in J(K)$, if the attribute set is $T \subseteq M$ and if $A \subseteq T$, $B \notin T$, the attribute set T in the access security context K is neither intent nor pseudointent.*

Proof. Firstly, proof T is not intent. $A \subseteq T$, $B \notin T$, we knew from property 3 that $T \subseteq f(g(T))$, then $A \subseteq T \subseteq f(g(T))$, $f(g(A)) \subseteq f(g(T))$. Subtract A from both ends of $f(g(A)) \subseteq f(g(T))$ and get $f(g(A)) - A \subseteq f(g(T)) - A$. And because $A \longrightarrow B \in J(K)$, then $f(g(A)) - A =$ B. Because $D \notin T$, then $f(g(T)) - A \notin T$. Add A to both ends of $f(g(T)) - A \notin T$, get $f(g(T)) \notin T \cup A$. Because $A \subseteq T$, therefore, $f(g(T)) \notin T$, T, is not intent.

(2) Lastly, proof T is not pseudointent. If T satisfies the definition of the pseudointent (2), then each pseudointent $Y_1 \subset T$ must meet $f(g(Y_1)) \subseteq T$, because $A \longrightarrow B \in J(K)$. Thus, in K, A is a pseudointent. Because of $A \subseteq T$, $f(g(A)) = B \notin T$. Therefore, T does not satisfy the definition of pseudointent (2) that T is not a pseudointent in K.

Theorem 11 shows that the set of permissions (attributes) is neither intent nor pseudointent if it is not related to any implication in the Duquenne–Guigues. Because in the attribute exploration, only the set of permissions (attributes) that are intent or pseudointent are considered, and the set of permissions (attributes) that satisfy theorem 11 can be ignored and not calculated.

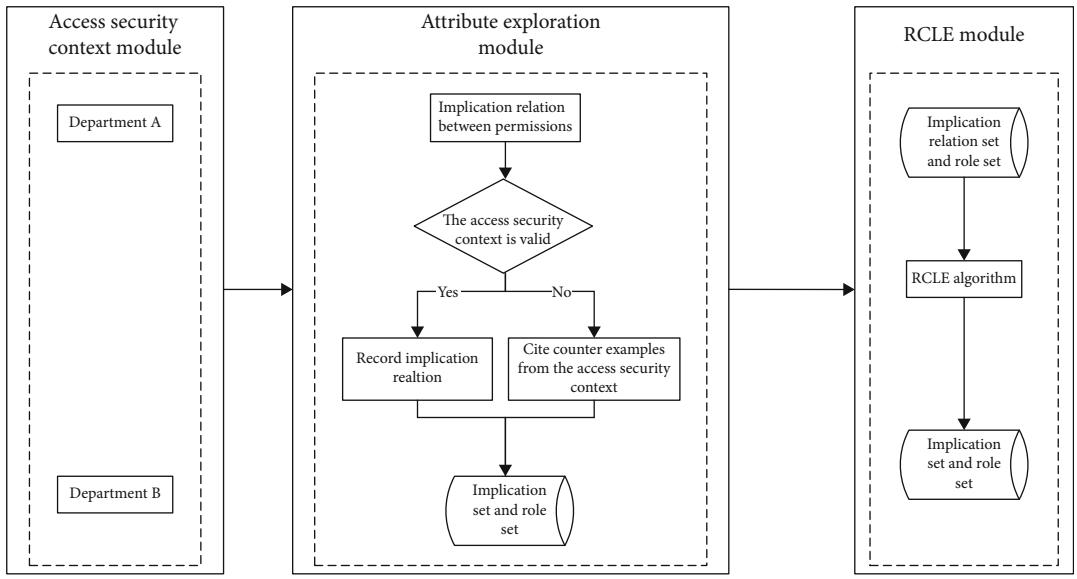

Figure 1: *RCLE* model framework.

Theorem 12. *RCLE* $= (K_1, K_2, J(K_1), C(K_1), J(K_2), C(K_2))$, $K_1 = (U_1, M_1, I_1)$, $K_2 = (U_2, M_2, I_2)$, *access security context* $K = K_1 + K_2$, *and the Duquenne–Guigues of* K *is* $J(K)$. *The permission set* D *is related to* $J(K)$, $D \longrightarrow f_1(g_1(D)) - D \in J(K_1)$. *If* $f_1(g_1(D)) \cup f_2(g_1(D)) \neq D$ *then* $D \longrightarrow f_1(g_1(D)) \cup f_2(g_1(D)) - D \in J(K)$. *If* $f_1(g_1(D)) \cup f_2(g_1(D)) = D$, *then* $f_1(g_1(D)) \cup f_2(g_1(D)) \in C(K)$.

Proof. Because D is related to $J(K)$, so by definition 7, we know that D is intent or pseudointent. $D \longrightarrow f_1(g_1(D)) - D \in J(K_1)$, then in K_1, the users that coown the permission set D are $g_1(D)$. According to definition 10, $U_1 = U_2 = U$. Because $K = K_1 + K_2$, so $f(g(D)) = f_1(g_1(D)) \cup f_2(g_1(D))$. If $f_1(g_1(D)) \cup f_2(g_1(D)) \neq D$, then D is a pseudointent; so, $D \longrightarrow f_1(g_1(D)) \cup f_2(g_1(D)) - D \in J(K)$. If $f_1(g_1(D)) \cup f_2(g_1(D)) = D$, then D is an intent, so $f_1(g_1(D)) \cup f_2(g_1(D)) \in C(K)$.

Inference 13. RCLE $= (K_1, K_2, J(K_1), C(K_1), J(K_2), C(K_2))$, $K_1 = (U_1, M_1, I_1)$, $K_2 = (U_2, M_2, I_2)$, access security context $K = K_1 + K_2$, and the Duquenne–Guigues of K are $J(K)$. The permission set D is related to $J(K)$, $D \longrightarrow f_2(g_2(D)) - D \in J(K_2)$. If $f_2(g_2(D)) \cup f_1(g_2(D)) \neq D$, then $D \longrightarrow f_2(g_2(D)) \cup f_1(g_2(D)) - D \in J(K)$. If $f_2(g_2(D)) \cup f_1(g_2(D)) = D$, then $f_2(g_2(D)) \cup f_1(g_2(D)) \in C(K)$.

Theorem 14. *RCLE* $= (K_1, K_2, J(K_1), C(K_1), J(K_2), C(K_2))$, $K_1 = (U_1, M_1, I_1)$, $K_2 = (U_2, M_2, I_2)$, *access security context* $K = K_1 + K_2$, *and the Duquenne–Guigues of* K *are* $J(K)$. *The permission set* D *is related to* $J(K)$, $D \in C(K_1)$. *If* $D \cup f_2(g_1(D)) = D$, *then* $D \cup f_2(g_1(D)) \in C(K)$. *If* $D \cup f_2(g_1(D)) \neq D$, *then* $D \longrightarrow D \cup f_2(g_1(D)) \in J(K)$.

Proof. Because D is related to $J(K)$, so by definition 7, we know that D is intent or pseudointent. $D \in C(K_1)$, and then in K_1, the users that coown the permission set D are $g_1(D)$. According to definition 10, $U_1 = U_2 = U$. Because $K = K_1 + K_2$, so $f(g(D)) = f_1(g_1(D)) \cup f_2(g_1(D))$. If $f_1(g_1(D)) \cup f_2(g_1(D)) \neq D$, then D is a pseudointent, so $D \longrightarrow D \cup f_2(g_1(D)) = D - D \in J(K)$. If $D \cup f_2(g_1(D)) = D$, then D is an intent, so $f_1(g_1(D)) \cup f_2(g_1(D)) \in C(K)$.

Inference 15. RCLE $= (K_1, K_2, J(K_1), C(K_1), J(K_2), C(K_2))$, $K_1 = (U_1, M_1, I_1)$, $K_2 = (U_2, M_2, I_2)$, access security context $K = K_1 + K_2$, and the Duquenne–Guigues of K are $J(K)$. The permission set D is related to $J(K)$, $D \in C(K_2)$. If $D \cup f_1(g_2(D)) = D$, then $D \cup f_1(g_2(D)) \in C(K)$. If $D \cup f_1(g_2(D)) \neq D$, then $D \longrightarrow D \cup f_1(g_2(D)) \in J(K)$.

Theorems 12 and 14 show that in the *RCLE* model, if a permission set is related to the Duquenne–Guigues of an access security context, then we can use the results obtained in the subdivision to carry out the union operation with the results calculated by other departments and judge whether the obtained results are intent or pseudointent.

4. RCLE Model Framework

Based on the above definitions and theorems, this section designs a model of RBAC role collaborative learning and exploration (RCLE) by referring to the framework of traditional attribute exploration algorithm and expert questions. the algorithm uses the traditional attribute exploration framework to discover the roles of different permissions system, then automatically revises the set of roles and the Duquenne–Guigues according to the obtained knowledge and the proposed theorem. In this way, we can get the required roles and the implication relationship between permissions and permissions of the system after the fusion of multiple systems. The model architecture is shown in Figure 1.

Input: two access security contexts $K_1 = (U_1, M_1, I_1)$, $K_2 = (U_2, M_2, I_2)$; Duquenne–Guigues $J(K_1)$, $J(K_2)$; intent set $C(K_1), C(K_2)$
Output: access security context K, $J(K)$, $C(K)$
BEGIN
1. $J(K)=\varnothing$, $C(K)=\varnothing$
2. $K=K_1 \cup K_2$
3. WHILE ($B \neq M$)
4. IF ($B \in C(K_1)$) THEN
5. IF ($B \cup f_1(g_2(B)) == B$) THEN
6. Add $B \cup f_1(g_2(B))$ to $C(K)$
7. ELSE
8. Add $B \longrightarrow B \cup f_1(g_2(B)) - B$ to $J(K)$
9. END IF
10. B =FindNextB($J(K), B$)
11. Continue
12. END IF
13. IF ($B \in C(K_2)$) THEN
14. IF ($B \cup f_2(g_1(B)) == B$) THEN
15. Add $B \cup f_2(g_1(B))$ to $C(K)$
16. ELSE
17. Add $B \longrightarrow B \cup f_2(g_1(B)) - B$ to $J(K)$
18. END IF
19. B=FindNextB($J(K), B$)
20. Continue
21. END IF
22. IF ($B \in J(K_1)$) THEN
23. IF ($f_1(g_1(B)) \cup f_2(g_1(B)) == B$) THEN
24. Add $f_1(g_1(B)) \cup f_2(g_1(B))$ to $C(K)$
25. ELSE
26. Add $B \longrightarrow f_1(g_1(B)) \cup f_2(g_1(B))$ to $J(K)$
27. END IF
28. B=FindNextB($J(K), B$)
29. Continue
30. END IF
31. IF ($B \in J(K_2)$) THEN
32. IF ($f_2(g_2(B)) \cup f_1(g_2(B)) == B$) THEN
33. Add $f_2(g_2(B)) \cup f_1(g_2(B))$ to $C(K)$
34. ELSE
35. Add $B \longrightarrow f_2(g_2(B)) \cup f_1(g_2(B))$ to $J(K)$
36. END IF
37. B=FindNextB($J(K), B$)
38. Continue
39. END IF
40. IF ($f(g(B)) \neq B_i$)
41. $J(K)=J(K) \cup (B \longrightarrow f(g(B)) - B)$
42. ELSE
43. $C(K)=C(K) \cup (B)$
44. END IF
45. END WHILE
46. END

Algorithm 1: RCLE algorithm description.

Using the attribute exploration algorithm, the role discovery algorithm interacts with system security managers in different departments to obtain the required set of roles (intent) and the set of implications between permissions (Duquenne–Guigues) in each department. The following is the specific process of the attribute exploration role discovery algorithm:

Input: Attribute set B, implies set $J(K)$
Output: The next attribute set NextB
BEGIN
1. B'=Find the lexicographic order of B s next
2. Flag = TRUE
3. WHILE(Flag)
4. FOR each $a_1 \longrightarrow b_1 \in J(K)$
5. IF ($a_1 \subseteq B' \&\& b_1 \notin B'$)THEN
6. B'=Find the lexicographic order of B' next
7. BREAK
8. ELSE
9. RETURN B'
10. END IF
11. END FOR
12. END

Algorithm 2: findNextB algorithm description.

At the beginning of the algorithm, the access security context is empty, the Duquenne–Guigues is empty, and the intent set is empty. Then, the set of attributes to be tested is continuously generated in lexicographic order, and an expert is asked if the implication with the attribute set as the preceding is true. If not, add a counterexample to the access security context and recalculate. If true, the attribute set is judged to be intent or pseudointent. If it is a pseudointent, then an implication form with the pseudointent added to the Duquenne–Guigues. If it is not a pseudointent, according to the value dependence of the concept lattice and the correlation theory of the attribute set, it must be intent, and then the attribute set is added to the intent set.

The set of roles and the Duquenne–Guigues obtained are substituted into the RCLE model, and the set of roles and the Duquenne–Guigues required in the system after multidepartment system fusion are calculated. At the initial stage of the algorithm, the access security context is the union of multiple access security contexts, the Duquenne–Guigues is empty, and the set of roles is empty. In line 1 of the algorithm to determine whether the algorithm has reached the end state. Inline 4-8 of the algorithm, it means that the permissions set belongs to the role set of departments 1; so, the permissions jointly owned by users in department 1 and department 2 are calculated. Line 9-13 of the algorithm indicates that the permission set belongs to the role set of department 2; so, the permissions jointly owned by users in department 2 who have B permission set are calculated in department 1. Algorithm 14-23 is the processing process of the B permission set. Line 24-28 of the algorithm is the process where B does not exist in the set of roles of department 1 and department 2, nor in their Duquenne–Guigues, where the findNextB algorithm calculates the next permission set of B' according to the correlation definition.

5. Example of the RCLE Algorithm Process

This section illustrates the running process of the RCLE model with an example. Access security context $K_1 = (U_1, M_1, I_1)$ and $U_1 = (1, 2, 3, 4)$ represents (dean of faculty, dean of

Table 1: Access security context K_1.

	f	g	h	i
1	1	1	0	0
2	0	1	0	1
3	0	0	0	0
4	0	0	1	0

Table 2: Access security context K_2.

	a	b	c	d	e
1	0	0	1	1	1
2	0	0	0	0	0
3	0	0	1	1	1
4	1	1	1	1	0

teaching, dean of research, dean of academic affairs), and $M_1 = (f, g, h, i)$represents (student curriculum management, teacher information management, graduate employment information management, scientific research information management). The specific permission information is shown in Table 1. Access security context, $K_2 = (U_2, M_2, I_2)$, $U_2 = (1, 2, 3, 4)M_2 = (a, b, c, d, e)$ represents (student information management, student registration information management, student status management, student curriculum review, student curriculum development and modification). The specific permission information is shown in Table 2. The permissions $a < b < c < d < e < f < g < h < i$。

Get by using the attribute exploration role discovery algorithm $J(K_1) = \{e \longrightarrow cd, d \longrightarrow c, c \longrightarrow d, b \longrightarrow acd, a \longrightarrow bcd\}$, $C(K_1) = \{\varnothing, cd, cde, abcd, abcde\}$, $J(K_2) = \{i \longrightarrow g, gh \longrightarrow fi, f \longrightarrow g, fgi \longrightarrow h\}$, and $C(K_2) = \{\varnothing, h, g, gi, fg, fghi\}$and plug $J(K_1)$, $C(K_1)$, $J(K_2)$, $C(K_2)$ into the RCLE algorithm.

(1) The algorithm starts at $B=\varnothing$.

(2) Because $\varnothing \in C(K_1)$, calculate $\varnothing \cup f_2(g_1(\varnothing)) = \varnothing = B$, add $\varnothing$ to the set $C(K)$, calculate the next property set of B to be i, and make $B=i$.

(3) Because $i \in J(K_1)$, calculate $f_1(g_1(i)) \cup f_2(g_1(i)) - i = g \neq \varnothing$, add $i \longrightarrow g$ to the set J(K), calculate the next property set of B to be h, and make $B=h$.

(4) Because h does not exist in $J(K_1)$, $C(K_1)$, $J(K_2)$, and $C(K_2)$,. calculate $f(g(h)) - h = abcd \neq \varnothing$, add $h \longrightarrow abcd$ to the set $J(K)$, calculate the next property set of B to be g, and make $B=g$.

(5) Because g does not exist in$J(K_1)$, $C(K_1)$, $J(K_2)$, $C(K_2)$. calculate $f(g(g)) - g=\varnothing$, add g to the set $C(K)$, calculate the next property set of B to be gi, and make $B=gi$.

(6) Because gi does not exist in $J(K_1)$, $C(K_1)$, $J(K_2)$, $C(K_2)$, calculate $f(g(gi)) - gi=\varnothing$, add gi to the set $C(K)$, calculate the next property set of B is f, and make $B=f$.

(7) Because $f \in J(K_1)$, calculate $f_1(g_1(f)) \cup f_2(g_1(f)) - f = cdeg \neq \varnothing$, add $f \longrightarrow cdeg$ to the set $J(K)$, calculate the next property set of B which is e, and make $B=e$.

(8) Because $e \in J(K_1)$, calculate $f_1(g_1(e)) \cup f_2(g_1(e)) - e = cdg \neq \varnothing$, add $e \longrightarrow cdg$ to the set $J(K)$, calculate the next property set of B is d, and make $B=d$.

(9) Because $d \in J(K_2)$, calculate $f_2(g_2(d)) \cup f_1(g_2(d)) - d=c \neq \varnothing$, add $d \longrightarrow c$ to the set $J(K)$, calculate the next property set of B is c, and make $B=c$.

(10) Because $c \in J(K_2)$, calculate $f_2(g_2(c)) \cup f_1(g_2(c)) - c=d \neq \varnothing$, add $c \longrightarrow d$ to the set J(K), calculate the next property set of B is cd, and make $B=cd$.

(11) Because cd does not exist in $J(K_1)$, $C(K_1)$, $J(K_2)$,and $C(K_2)$ calculate $f(g(cd)) - cd=\varnothing$, add cd to the set $C(K)$, calculate the next property set of B is cdg, and make $B=cdg$.

(12) ……

(13) This article will not repeat the process because of the limited space.

At the end of the algorithm, $C(K) = \{\varnothing, g, gi, cd, cdeg, cdefg, abcdh, abcdefghi\}$, $J(K) = \{i \longrightarrow g, h \longrightarrow abcd, f \longrightarrow cdeg, e \longrightarrow cdg, d \longrightarrow c, c \longrightarrow d, cdg \longrightarrow e, cdegi \longrightarrow abfh, b \longrightarrow acdh, a \longrightarrow bcdh, abcdegh \longrightarrow fi\}$.

It can be seen from the above algorithm example process that the RCLE model utilizes the traditional attribute exploration role discovery algorithm to interact with the system managers of multiple departments, so as to obtain the set of roles and the implication relation between permissions under the combination of multiple departments.

6. Experiment and Analysis

6.1. Experimental Design. In order to verify the performance of the model proposed in this paper, the random function simulation in the JAVA language MATH library is used to generate two sets of access security context as test data. The experimental design is divided into two aspects. The first aspect is to observe the change in the number of the implication relation (Duquenne–Guigues) by changing the experimental conditions. The second aspect is to change the experimental conditions to observe the changes in the number of roles (intent).

In the experiment, the algorithm traverses the access security context to answer the questions instead of the experts. The algorithm takes the randomly generated access security context as the objective access security context and traverses the entire access security context when judging whether the implication relation is true. If all users in the access security context meet the implication relation of this implication, the implication is considered to be true. Otherwise, it is considered that this implication relation is not valid, and a user is taken from the access security context and provided to the

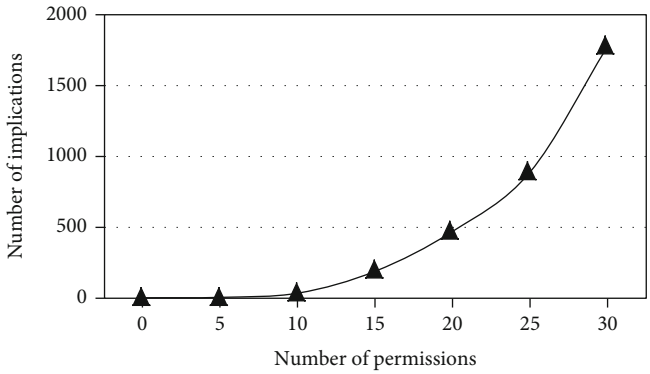

Figure 2: Number of implications (number of users: 30).

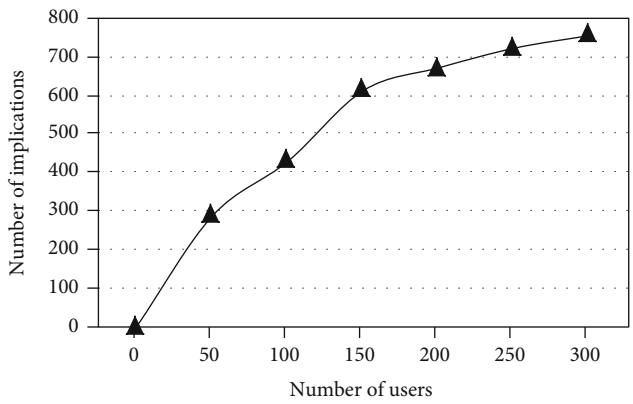

Figure 3: Number of implications (number of permissions: 15).

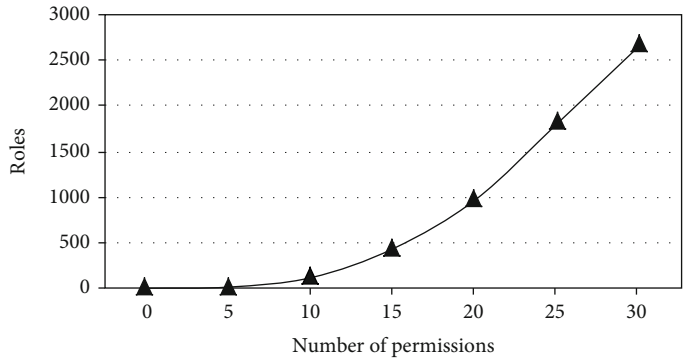

Figure 4: Number of roles (number of users: 30).

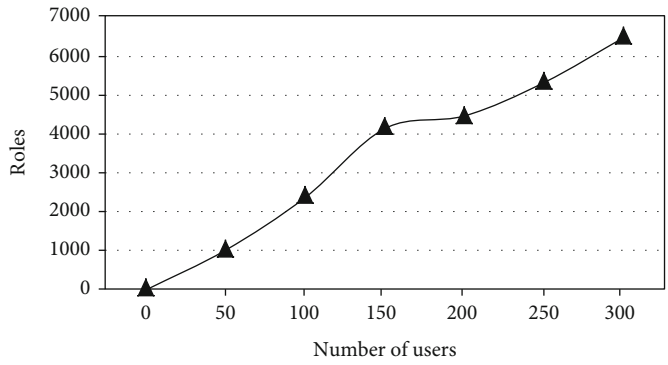

Figure 5: Number of roles (number of permissions: 15).

algorithm as a counterexample. The test platform hardware is 3.4GHZ CPU, and the 16GB memory operating system is Windows X10.

The first group of experiments sets the access security context with the same number of users (objects) and the number of permissions (attributes) from 0 to 30 at an interval of 5 to test. The purpose of the test is to fix the number of users to change the number of permissions and observe the change in the number of implications. The test results are shown in Figure 2.

The second group sets the number of access security context with the same number of permissions (attributes), and the number of users (objects) is tested from 0 to 300 at intervals of 50. The purpose of testing is to fix the number of permissions, change the number of users, and observe the change of the number of implications. The test results are shown in Figure 3.

The third group of experiments sets access security context with the same number of users (objects) and the number of permissions (attributes) from 0 to 30 at an interval of 5 to test. The purpose of the test is to fix the number of users, change the number of permissions, and observe the change in the number of roles. The test results are shown in Figure 4.

The fourth group sets the number of access security context with the same number of permissions (attributes), and the number of users (objects) is tested from 0 to 300 at intervals of 50. The purpose of the test is to fix the number of permissions, change the number of users, and watch the number of roles change. The test results are shown in Figure 5.

6.2. Experimental Analysis. The first, second, third, and fourth groups of experiments show that whether the number of fixed objects, changing the number of attributes, or the number of fixed attributes, changing the number of objects, the implication relationship, and role (intent) increase with the scale expansion of the access security context.

The RCLE model proposed in this paper not only avoids the time-consuming and labor-consuming process of role requirement analysis and questionnaire survey in the process of role construction but also solves the defects of the traditional auxiliary interactive question and answer algorithm based on attribute exploration, which does not support crossdepartments.

7. Conclusion

Because of the defect that the traditional semiautomatic heuristic method for constructing the RBAC system cannot construct a role system in different permission system departments, this paper proposes a model of RBAC role cooperative learning and exploration. Based on the local access security context, three theorems are summarized from the local point of view, and the proposed theorems are proved by mathematical rigor. Finally, a model of RCLE is given according to the theorems. The model uses the traditional attribute exploration role

discovery method to construct the role system of different permission systems, and then according to the theorem proposed in this paper, calculates the role system of the multiple departments. Because the RCLE model greatly saves the time-consuming steps in the process of role to formulate and has characteristic of the interdepartmental build role, and so here we will further the development of tools for easier operation and makes the model able to get the more extensive application and development.

Data Availability

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

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Scientific and Technological Project of Henan Province (Grant No. 202102310340), Foundation of University Young Key Teacher of Henan Province (Grant Nos. 2019GGJS040 and 2020GGJS027), and Key Scientific Research Projects of Colleges and Universities in Henan Province (Grant No. 21A110005).

References

[1] X. Zhang, F. Ding, L. Xu, and E. Yang, "Highly computationally efficient state filter based on the delta operator," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 6, pp. 875–889, 2019.

[2] X. Zhang, F. Ding, and E. Yang, "State estimation for bilinear systems through minimizing the covariance matrix of the state estimation errors," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 7, pp. 1157–1173, 2019.

[3] Y. Sun, M. Li, S. Su, Z. Tian, W. Shi, and M. Han, "Secure data sharing framework Via hierarchical greedy embedding in darknets," *Mobile Networks and Applications*, vol. 26, no. 2, pp. 940–948, 2021.

[4] L. Yin, B. Fang, Y. Guo, Z. Sun, and Z. Tian, "Hierarchically defining Internet of Things security: from CIA to CACA," *International Journal of Distributed Sensor Networks*, vol. 16, no. 1, 2020.

[5] L. Yin, L. I. Ran, J. Ding, L. I. Xiao, and L. I. Ang, "δ-calculus: a new approach to quantifying location privacy," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1323–1342, 2020.

[6] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[7] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST model for role based access control: Towards a unified standard," in *Proc ofthe 5th ACM Workshop on Role Based Access Control*, pp. 47–63, New York, NY:ACM Press, 2020.

[8] H.-C. Chen, "Collaboration IoT-based RBAC with trust evaluation algorithm model for massive IoT integrated application," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 839–852, 2019.

[9] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role-based access control (RBAC): Features and motivations," in *Proceedings of 11th annual computer security application conference*, pp. 241–248, New Orleans, Louisiana, United States, 1995.

[10] E. Bertino, "RBAC models – concepts and trends," *Computers& Security*, vol. 22, no. 6, pp. 511–514, 2003.

[11] A. Colantonio, R. Di Pietro, and A. Ocello, *Role Mining in Business:Taming Role-Based Access Control Administration*, World Scientific, 2012.

[12] D. Xu, Z. Tian, R. Lai, X. Kong, Z. Tan, and W. Shi, "Deep learning based emotion analysis of microblog texts," *Information Fusion*, vol. 64, pp. 1–11, 2020.

[13] Z. Sun, L. Yin, C. Li, W. Zhang, A. Li, and Z. Tian, "The QoS and privacy trade-off of adversarial deep learning: an evolutionary game approach," *Computers & Security*, vol. 96, article 101876, 2020.

[14] C. L. Yin, Y. F. Zhu, J. L. Fei, and X. Z. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[15] D. Xu, J. Pan, X. Du, B. Wang, M. Liu, and Q. Kang, "Massive fishing website URL parallel filtering method," *IEEE Access*, vol. 6, pp. 2378–2388, 2018.

[16] W. Huang, Y. Xu, X. Hu, and Wei, "Compressive hyperspectral image reconstruction based on spatial–spectral residual dense network," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 5, pp. 884–888, 2020.

[17] W. Huang, Y. Huang, H. Wang, Y. Liu, and H. J. Shim, "Local binary patterns and Superpixel-based multiple kernels for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4550–4563, 2020.

[18] Y. Pang, L. Peng, Z. Chen, B. Yang, and H. Zhang, "Imbalanced learning based on adaptive weighting and Gaussian function synthesizing with an application on Android malware detection," *Information Sciences*, vol. 484, pp. 95–112, 2019.

[19] L. Zhang, H.-l. Zhang, D.-j. Han, and X.-j. Shen, "Theory and algorithm of role minimization problem in RBAC model based on concept lattice," *Acta Electronics Sinica*, vol. 42, no. 12, pp. 2371–2378, 2014.

[20] R. W. Ganter, *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag, Berlin, 1999.

[21] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer Science & Business Media, 2012.

[22] L. Qin, J. Li, and Y. Wang, "Knowledge discovery based on concept lattice and its application in university employment data analysis," *Journal of Shandong University (Natural Science Edition)*, vol. 50, no. 12, pp. 58–64, 2015.

[23] X. Shen, J. Yang, and L. Zhang, "Attribute exploration algorithm based on uncorrelated attribute sets," *Computer Science*, vol. 48, no. 4, pp. 54–62, 2021.

[24] L. Wei, L. Lin, J. Qi, and T. Qian, "Rules acquisition of formal decision contexts based on three-way concept lattices," *Information Sciences*, vol. 516, pp. 529–544, 2020.

[25] Y. Mi, W. Liu, Y. Shi, and J. Li, "Semi-supervised concept learning by concept-cognitive learning and concept space," *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2020.

[26] J. Li, L. Wei, and Z. Zhang, "Concept lattice theory and method and their research prospect," *Pattern Recognition and Artificial Intelligence*, vol. 33, no. 7, pp. 619–642, 2020.

[27] C. A. Kumar, "Designing role-based access control using formal concept analysis," *Security and Communication Networks*, vol. 6, no. 3, 383 pages, 2013.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9935780, 11 pages
https://doi.org/10.1155/2021/9935780

Research Article

A Novel Privacy-Preserving Mobile-Coverage Scheme Based on Trustworthiness in HWSNs

Chunyang Qi,[1] **Jie Huang**,[1,2] **Bin Wang**,[3] **and Hongkai Wang**[4]

[1]*School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China*
[2]*Purple Mountain Laboratories, Nanjing 211111, China*
[3]*College of Electrical Engineering, Zhejiang University, Hangzhou 310058, China*
[4]*State Grid Zhejiang Electric Power Corporation Information & Telecommunication Branch, Hangzhou 310007, China*

Correspondence should be addressed to Jie Huang; jhuang@seu.edu.cn and Bin Wang; bin_wan@zju.edu.cn

Received 3 March 2021; Accepted 21 May 2021; Published 17 June 2021

Academic Editor: Lihua Yin

To solve the problem of security deployment in a hybrid wireless sensor network, a novel privacy-preserving mobile coverage scheme based on trustworthiness is proposed. The novel scheme can efficiently mitigate some malicious attacks such as eavesdropping and pollution and optimize the coverage of hybrid wireless sensor networks (HWSNs) at the same time. Compared with the traditional mobile coverage scheme, the security of data transmission and mobility are considered in the deployment of HWSNs. Firstly, our scheme can mitigate the eavesdropping attacks efficiently utilizing privacy-preserving signature. Then, the trust mobile protocol based on the trustworthiness is used to defend the pollution attacks and improve the security of mobility. In privacy-preserving signature, the hardness of discrete logarithm determines the degree of security of the privacy-preserving signature. The correctness and effectiveness of signature algorithm are proven by the probabilities of the native messages which can be recovered and forged which is negligible. Furthermore, a mobile scheme based on the trustworthiness (MSTW) is proposed to optimize the network coverage and improve the security of mobility. Finally, the simulation compared with a previous algorithm is carried out, in which the communication overhead, computational complexity, and the coverage are given. The result of the simulation shows that our scheme has roughly the same network coverage as the previous schemes on the basis of ensuring the security of the data transmission and mobility.

1. Introduction

As wireless sensor networks (WSNs) have been rapidly developing and growing popularity, the mobile deployment in an interested area with maximum coverage has become an important challenge in the field of research. However, the security of the data transmission and the mobility are generally not considered in the mobile deployment. Hybrid wireless sensor networks (HWSNs) usually include two types of nodes, such as mobile nodes and static nodes, where the mobile nodes have the ability of movement to increase the coverage in the monitored area. If there are eavesdropping and pollution attacks in the network, the innocent nodes can be eavesdropped and quickly polluted during the adjustment. For the mobile deployment in HWSNs, the previous researchers had done tremendous researches.

A large number of network coverage optimization schemes for HWSNs had been proposed in [1–6]. Unlike the previously proposed mobile coverage schemes, our mobile coverage scheme based on privacy-preserving signature and trustworthiness can mitigate the eavesdropping and pollution attacks effectively and ensures that the network coverage will not be significantly reduced at the same time.

Due to the limited resources of HWSNs, traditional symmetric or asymmetric cryptographic algorithms are not available. The trustworthiness is an emerging security technology based on network dynamic parameters, which can estimate the level of network security in HWSNs. The problems of node integrity and authentication are addressed utilizing the trustworthiness in [7–10]. The investigations and studies have shown that the recommended trust and the dynamic network information of the current node can effectively

evaluate the node security level. Moreover, another solution [11] to achieve evaluation is that the sensor nodes will be equipped with additional computing units to evaluate trust. For more related work in detail, please refer to Section 2.

1.1. Motivation and Contribution. In the past decade, the previous researchers had conducted tremendous studies for mobile coverage. However, many issues are not addressed. A genetic scheme utilized virtual force and a particle swarm to optimize network coverage for HWSNs were proposed in [12]. However, the security issues in mobile deployment also need to be considered. If there are eavesdropping and pollution attacks in the HWSNs, the eavesdropping attack can result in the disclosure of location information of a mobile node, thereby attackers launched a premeditated attack. Relatively, the pollution attack will quickly infect malicious information to its neighbor nodes in a communication phase. Figure 1 shows the probability that nodes are vulnerable to infection when there is an attack in the network. If a node is malicious, the area within the communication radius is marked as polluted region. While the Euclidean distance between the sensor nodes and the malicious node is less than $2r$, the sensor nodes will be marked as high-risk nodes. If the Euclidean distance is equal to $2r$, the sensor nodes are labeled secondary risk nodes. All other sensor nodes are low risk. During the deployment process, it is an arduous task against pollution attack by protecting the sensor nodes integrity and realizing sensor nodes authentication. Bao and Chen [13] proposed a scalable trust management protocol, in which the multidimensional trust attributes extracted from communication and social networks are considered. However, many subjective factors were added in the process of selecting trust attributes.

Considering that and the target of mobile coverage in HWSNs, the main contributions of this paper are summarized as follows:

(1) We proposed a novel trust evaluation and update scheme including information entropy theory to resist the pollution attacks, in which the probability of trust can be objectively evaluated and updated

(2) At the same time, a privacy-preserving signature scheme which can sign the data transmission during the lifetime of HWSNs is proposed to defend against eavesdropping attacks

(3) Moreover, a mobile scheme based on the trustworthiness (MSTW) including an optimized virtual force algorithm is proposed to reduce the probability of being polluted. Meanwhile, the maximum network coverage is obtained under the overall network trust

1.2. Organization. The other chapters of this paper are introduced as follows. Related researches regarding mobile coverage based on the trust and privacy-preserving signature technologies are introduced in Section 2. The network and attack model are demonstrated in Section 3, where the mobile framework is also given. In Section 4, a mobile scheme based on the trustworthiness (MSTW) is proposed to improve the network coverage and reduce the probability of the sensor nodes being polluted. In Section 5, a suitable privacy-preserving signature scheme is introduced to ensure the security of data transmission. In Section 6, the full construction of the privacy-preserving mobile coverage scheme based on the trustworthiness (PP-MSTW) and the virtual force are presented, and the results of performance analysis are also given. Finally, the conclusion is presented in Section 7.

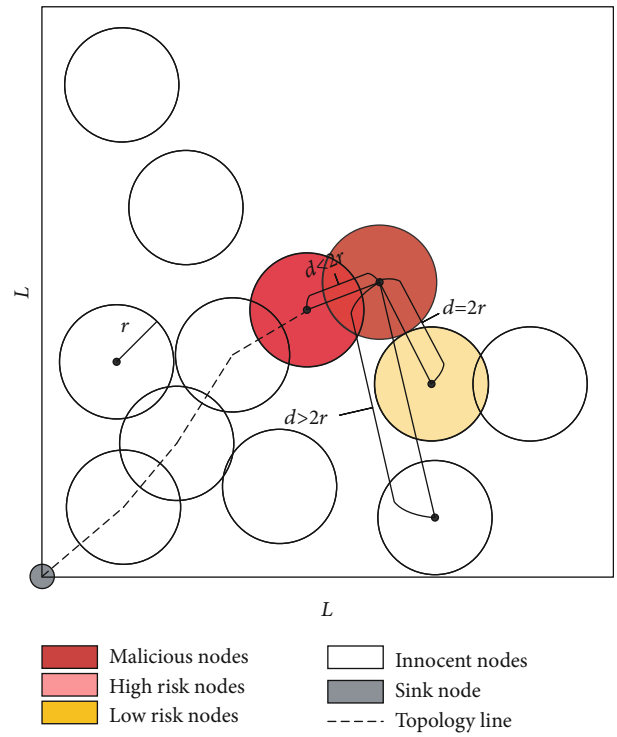

Figure 1: The distribution of different types of node security level.

2. Related Works

Mobile coverage in the deployment of HWSNs has always been a research hotspot, in which tremendous researches had been proposed. However, the security of HWSNs during the deployment is always not considered. The trust evaluation scheme can replace the traditional encryption mechanism with minimal resource. Some works in the field of mitigating eavesdropping attacks, trustworthiness for pollution attacks, and network coverage are discussed below.

2.1. Privacy-Preserving Signature. In HWSNs, attackers often obtain the dynamic information of the entire network by monitoring certain sensor nodes in the network. Yang et al. [14] studied several typical network structures and proved that as long as the eavesdropper monitors the data of one sensor node in each cycle, the entire network system can be completely observed.

There are two ways to defend against eavesdropping attacks including information theoretic and computational approaches. In information theoretic schemes, Zhang et al. [15] proposed a novel type of network $\boldsymbol{P}$-coding scheme,

which prevented the network from being eavesdropped globally by performing lightweight sorting encryption on each native message and its encoding vector. Furthermore, Chen and Wang [16] applied the fake signature to network coding in the environment of IOT devices, which can resist both external and internal attacks at the same time and can also achieve the highest security level. In computational approaches, Nikravan et al. [17] proposed a lightweight computing scheme based on identity online and offline information to resist eavesdropping attacks with high computing power. Huang and Zhu [18] used the method of strategic equilibrium game to capture deception or eavesdropping, which can achieve Bayesian Nash equilibrium under an iterative algorithm.

2.2. Trust Evaluation and Update. Currently, most traditional trust evaluation models focus on sensor radio and the number of successful data transmission. Sensor nodes build a trust model through the measurable parameters such as remaining energy and the recommendation from the neighboring nodes. Ganeriwal et al. [19] proposed a framework for evaluating the communication trust of neighbor nodes to ensure the security of sensor network. Though the framework had good robustness, the recommendations of other neighboring nodes were not considered. With the continuous development of trust evaluation framework, the trust model was trained by the neural network fuzzy inference [20], in which the parameter accuracy was optimized by the sorting genetic theory. Finally, they proposed a trust management framework with the higher security. However, the scheme neglected the recommendations from the neighboring nodes, which resulted in the trust values being not soundest. For this problem, an effective clustered trust model was proposed [21] to consider more external and human intervention factors, in which the trust weight between the subtrust sets is adaptive. However, the calculation cost will increase continuously due to weight updating, in which the method needs to learn repeatedly interaction information between sensor nodes.

2.3. Maximum Mobile-Coverage in HWSNs. For sensor network coverage issues, researchers had done a lot of research in the previous decades.

A novel distributed and centralized aggregation method is proposed in [22] to reduce the sensor density in a limited area, which can also prolong the sensor network life to the greatest extent. Chen et al. [23] proposed a supplementary solution for network coverage to dynamically maintain the coverage during the network life. When the nodes in the network are depleted or damaged due to the reasons such as energy exhaustion or damage, the sensor nodes with redundant coverage are dynamically adjusted to supplement the missing coverage. Naveen and Kumar [24] studied the problem of minimizing the cost of network deployment under the constraint of average vacancy, which calculated the density function of the relay nodes to preset the initial position of the nodes and verified the effectiveness of the scheme. In dynamic HWSNs, Cao et al. [25] proposed an improved social spider optimization strategy to reduce the network coverage blind spots and redundancy, which simulated the movement law of spiders and cooperation mechanism to achieve the optimal solution of network coverage.

3. Problem Statement

3.1. Hybrid Network Model. Suppose that hybrid sensor nodes with two groups of different attributes are randomly deployed in a two-dimensional space, and each node is assigned a unique identifier. The different two groups of nodes include static and mobile sensor nodes. The network topology of this HWSNs is $L - G = (L, V, E, r_s)$, in which L expresses the side length of the rectangular deployment region, V denotes the combination of two types of nodes, E means the topological connection combination between sensor nodes, and r_s shows the sensing radius.

Generally, the communication radius r_c of the sensor nodes is $r_c = 2r_s$ in the HWSNs. The characteristics of nodes and conversion rules are given as follows:

(1) Static sensor nodes: such nodes do not have the ability to move. After the deployment is completed, the position of the nodes is not allowed to be changed. At the same time, the energy consumption of nodes is mainly caused by data communication.

(2) Mobile sensor nodes: if there are no restrictions, mobile nodes can move freely within the deployment range. Such nodes usually have higher energy and computing resources than static nodes. Here, we specify the energy of mobile nodes as $E_m = 5 * E_s$.

(3) Node conversion rules: the mobile sensor nodes need to consume a lot of energy in the process of moving. When the current remaining energy of the mobile nodes drops below 50% of the average energy of the entire network, the mobility of nodes will be removed. At this time, the mobile node will participate in information perception as a static node.

Each WSN has its cluster selection scheme to gather information from sensors. Figure 2 shows the deployment structure in our hybrid WSNs. The hybrid network model including static nodes and mobile nodes will execute cluster formation (CF) and optimal cluster head (CH) selection algorithm after deployment. In a window period, each cluster only has one CH. The nodes within a cluster usually communicate with the cluster head. The cluster head nodes are generally mobile nodes or the higher energy static nodes in the cluster. In addition, when the nodes in the network can directly communicate with the BS, such nodes can directly send data to the BS.

3.2. Attack Model. The inherent characteristics of HWSNs including open deployment environment and limited resources make it vulnerable to various types of unknown attacks. Previous researches about the mobile coverage usually do not consider the network security during the movement. We assume that before the sensor nodes start to implement the mobile coverage scheme, there are already attacked nodes in the network. These malicious nodes launch

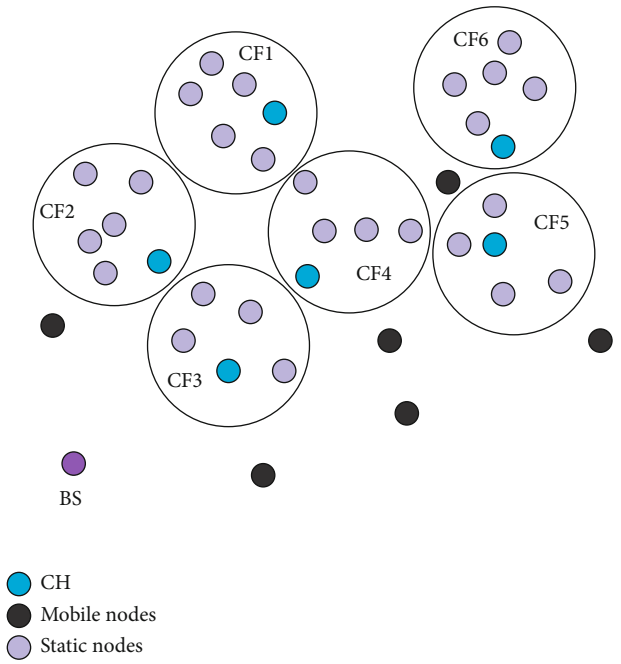

Figure 2: The deployment structure in Hybrid WSNs.

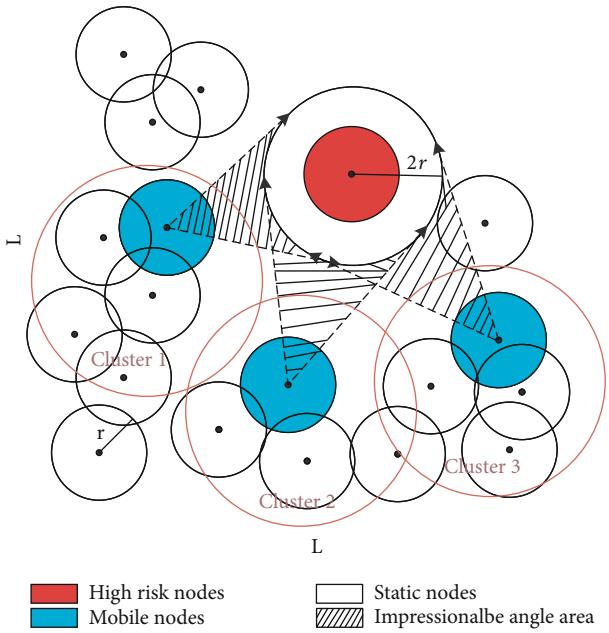

Figure 3: High risk moving angle and range.

attacks on the network by injecting fake data packets or tampering with the content of the transmitted data. The type of attack is usually named a pollution attack. The attack model is proposed as follows:

(1) Suppose there are N_M malicious nodes in our HWSNs. The types of attack occur before the mobile deployment. The malicious nodes usually inject false information or tamper with the contents of data packets

(2) Malicious nodes in the network can randomly select neighbor nodes within their communication range as the next hop pollution nodes

(3) As shown in Figure 3, there are two types of nodes including mobile and static in our HWSNs. Generally, mobile nodes have higher energy and computing resources, so they have higher defense performance than static nodes. Therefore, we think that the static nodes are easier to be attacked. The red area in the figure is the communication range r_s of a certain malicious node. When an innocent mobile node moves into the range of $2r_s$, the mobile sensor node will establish direct communication with the malicious node. At this time, the mobile node will be in a high-probability pollution area. In order to avoid the mobile node from moving into this area, the impressionable angle area is given to reduce the probability of the mobile node being polluted

4. A Multicluster Trust Scheme

According to the given network and attack model, a malicious defense model based on trust evaluation needs to be proposed. In this section, we construct a multicluster trust computing scheme to assess the trustworthiness of each node. The trust degree of the HWSNs is represented by (T, G_i), where G_i is the i-th evaluation window time. The multicluster trust scheme is defined as follows:

Definition 1. A multicluster trust scheme MCTS includes a tuple of four probabilistic distributed algorithms:

(a) Communication $(O_{i,j}, R_{i,j}) \mapsto (T_1(i,j))$: $O_{i,j}$ is the number of the data successfully transmitted, $R_{i,j}$ is the number of forward failures, and the communication trust is $T_1(i,j)$.

(b) Energy $(d_{i,j}, \varepsilon_{\text{amp}}, f_{\text{amp}}, E_{\text{elec}}) \mapsto (T_2(i,j))$: $d_{i,j}$ is the transmission distance, ε_{amp} is in free space, f_{amp} is the unit energy consumption coefficient in multipath attenuation model, E_{elec} represents the energy consumption when receiving data, and $T_2(i,j)$ is the energy trust.

(c) Probabilistic trust against attacks $(V_M, r_s, L) \mapsto (T_3(i,j))$: the set V_M is the malicious nodes, r_s is the sensing radius, L represents the side length of rectangular detection area, and $T_3(i,j)$ is the attacked trust.

(d) Aggregation $(T_n(i,j), w_n) \mapsto (T(i,j))$: $T_n(i,j)$ is the multitype trust values containing all the above types of trust variables. w_n symbols the weight set of each trust variable. Then, the aggregation $T(i,j)$ will be output.

4.1. The Multitype Trust Evaluation Method. The previous section has given the network and attack model, combined with the definition of the multicluster trust scheme. The

specific trust evaluation method is given. We separately calculated communication trust, energy trust, and attacked trust. Finally, a weight distribution scheme combined with information entropy is proposed to aggregate the mentioned multitype trust. The multitype trust aggregation model is expressed as

$$T(i,j) = \sum_{i=1}^{n} w_n T_n(i,j), \tag{1}$$

where $0 \le w_n \le 1$, $w_1 + w_2 + \cdots w_n = 1$; w_n is the weight factor of the multitype trust; and $T_n(i,j)$ is the multitype trust.

4.1.1. Communication Trust. The communication methods between nodes in the communication trust calculation are mainly divided into two modes: (1) node i directly interacts with node j, and (2) node i communicates with node j indirectly through k intermediate nodes. This paper uses the simplified beta trust model to calculate the trust value. The trust evaluation model adopts the model of previous research work [26], which is expressed as

$$f(t) = \frac{\alpha + 1}{\alpha + \beta + 2}\left(1 - \frac{\beta}{W}\right)\left(1 - \frac{1}{\alpha + \delta}\right), \tag{2}$$

where the successful interactions is α and the unsuccessful number is β, $1 - \beta/W$ is the penalty function, and $1 - 1/(\alpha + \delta)$ is the adjustment function.

Suppose that the number of successful direct interaction is $O_{i,j}$; otherwise, it is $R_{i,j}$. Therefore, the trust expression for direct communication between nodes i and j is

$$P = \frac{O_{i,j} + 1}{O_{i,j} + R_{i,j} + 2}\left(1 - \frac{R_{i,j}}{W}\right)\left(1 - \frac{1}{O_{i,j} + \delta}\right). \tag{3}$$

The mathematical trust expectation probability from i to j is expressed as the communication trust value in a round. So, the directly connected node communication trust is

$$T_d(i,j) = \sum_{t=1}^{n} \frac{E(P_{i,j})}{n}. \tag{4}$$

Here, n represents the amount of time windows within the effective operation time of the HWSNs.

In the case of the relay communication, node i needs to use k relay nodes to communicate with node j. The relay node z recommends an indirect trust value of node j to node i in the trust calculation model of node i for node j. The indirect trust calculation model in communication trust is shown as follows:

$$T_{id}(i,j) = \frac{\sum_{k=1}^{n} T_d(i,z) \cdot T_d(z,j)}{n}. \tag{5}$$

When node i and node j are neighbor nodes, the communication trust model will be calculated by Equation (4). Otherwise, Equation (5) will give the indirect communication trust value.

4.1.2. Energy Trust. In hybrid sensor networks, energy consumption is divided into two types, namely, communication and mobile consumption. Static nodes only include communication loss, while mobile nodes include both communication and mobile loss. In communication energy loss, when node i sends n bits information to node j successfully, the communication energy consumption model can be given as follows:

$$E_c(i,j) = \begin{cases} n \cdot E_{\text{elec}} + n \cdot \varepsilon_{\text{amp}} \cdot d_{ij}^2, & d_{ij} < d_0, \\ n \cdot E_{\text{elec}} + n \cdot f_{\text{amp}} \cdot d_{ij}^4, & d_{ij} \ge d_0, \end{cases} \tag{6}$$

where d_{ij} is the realistic transmission distance, d_0 represents the threshold of distance, E_{elec} represents the consumption of energy by the circuit for transmitting or receiving data per bit, and ε_{amp} and f_{amp} denote the energy loss in the free space and the multipath attenuation. So d_0 can be expressed as $d_0 = \sqrt{\varepsilon_{\text{amp}}/f_{\text{amp}}}$.

Additionally, the energy consumption during the movement should also be considered. The energy consumption can be expressed as follows:

$$E_m = 2d \sum_{i=0}^{n} \sqrt{{F_{x_i}}^2 + {F_{y_i}}^2}, \tag{7}$$

where d represents the side length of a square grid in the network and F_{x_i} and F_{y_i} are the virtual force received by node i.

In a mobile period, the remaining energy of the node is

$$T_2(i,j) = \begin{cases} \dfrac{E_N - E_m - E_c(i,j)}{E_t}, & E_N > \dfrac{E_t}{2}, \\ 0, & E_N \leqslant \dfrac{E_t}{2}, \end{cases} \tag{8}$$

where E_N symbols the remaining energy of the current node and E_t represents the initial energy. Usually, mobile nodes have a higher initialization energy level than static nodes.

4.1.3. Probabilistic Trust against Attacks. When there are malicious nodes in the HWSNs, the innocent nodes communicating with the malicious nodes have the probability of being polluted.

In this subsection, the Euclidean distance relationship analysis between nodes is utilized to assess the probability of defense against attacks between innocent sensor nodes. The ability of an innocent node defending against attacks indicates the trust degree of the sink node to those nodes. The higher the probability of an innocent node being attacked, the lower the trust degree, and vice versa.

According to the Euclidean distance between innocent nodes and a malicious node. E_1 and E_2 are defined as two different events. E_1 denotes the Euclidean distance between innocent nodes, and a malicious node j is less than the r_c of malicious nodes, and it can be expressed as $V_n \cap N_-(j) \neq \varnothing$. E_2 denotes the Euclidean distance is longer than the

communication radius, and it can be showed as $V_n \cap N_-(j) = \varnothing$, where $j \in V_M$ and $N_-(j)$ is the neighbor of malicious node j.

Lemma 2. *Nodes i and j are randomly deployed in the L × L rectangular monitoring region, in which the communication radius $r \in (0, W/2)$. The distribution of nodes i and j is given as*

$$D(d_{i,j} \le r) = \frac{\left((3\pi/2) + \left(4\sqrt{2}/3\right) - (25/12)\right)r^4 - (8/3)Lr^3 + \pi L^2 r^2}{L^4}. \tag{9}$$

According to Lemma 2, if two nodes in the monitoring area can communicate with each other, the probability is $D(d_{i,j} \le r)$. For node a, i.e., $\forall a \in V_n$, the probabilty of the node a that can communicate with malicious node j is

$$P(a \in N_-(j)) = \frac{D(r)}{2}. \tag{10}$$

Therefore, the probability of E_1 and E_2 can be given as follows:

$$\begin{aligned} P(E_1) &= P(V_i \cap N_-(j) \neq \varnothing) = 1 - \left(1 - \frac{D(r)}{2}\right)^{N_i}, \\ P(E_2) &= P(V_i \cap N_-(j) = \varnothing) = \left(1 - \frac{D(r)}{2}\right)^{N_i}. \end{aligned} \tag{11}$$

Finally, the probability trust against attacks can be expressed as follows:

$$T_3(i,j) = \begin{cases} P(E_1), & E_1 = \text{true}, \\ P(E_2), & E_2 = \text{true}. \end{cases} \tag{12}$$

4.2. Trust Aggregation. The weight relationships between the trust measures are more objectively determined. Information entropy can express the probability of a certain specific information, so it can be used to calculate the weights of uncorrelated subtrust distributions. The probability formula of information entropy is as follows:

$$H(\mu) = -\sum Q(\mu) \log_2(\mu), \tag{13}$$

where μ is the information variable and $Q(\mu)$ is the probability distribution function. Supposing that R is the recommended trust from three types of subtrust evaluation parameters, $1 - R$ is the degree to which these trust levels are suspected. With the above assumptions, the recommendation function of trust degree is

$$H(R) = -R \log_2 R - (1 - R) \log_2 (1 - R). \tag{14}$$

When there are multiple types of recommended trust values, not all trust values have the same recommendation weight. The weight of each trust value is independent, and it occupies a different degree of importance in the overall recommendation. Therefore, the weight entropy of R_{ij}^k can be given as follows:

$$H\left(R_{ij}^k\right) = -R_{ij}^k \log_2 R_{ij}^k - \left(1 - R_{ij}^k\right) \log_2 \left(1 - R_{ij}^k\right). \tag{15}$$

In practical applications, malicious nodes in the HWSNs usually disguise or slander the recommended trust value to deviate from the correct estimated value. Moreover, the previous trust weight distribution schemes often adopt artificial weight preset methods, which further leads to the deviation of trust weight distribution from objectivity. The information entropy weight distribution scheme can effectively reduce the degree of defamation of malicious nodes. According to Equation (15), the trust weight distribution can be given as follows:

$$\omega_k = \frac{1 - \left(\left(H\left(R_{ij}^k\right)\right) / \left(\log_2 R_{ij}^k\right)\right)}{\sum_{k=1}^{n}\left[1 - \left(\left(H\left(R_{ij}^k\right)\right) / \left(\log_2 R_{ij}^k\right)\right)\right]}. \tag{16}$$

Therefore, the final trust can be expressed as follows:

$$T(i,j) = \sum_{k=1}^{n}\left(\omega_k T_{ij}^k\right). \tag{17}$$

5. The Privacy-Preserving Signature Scheme

In the HWSNs, the trust evaluation of each nodes can drive the mobile nodes to a safer location. However, the data transmission security should also be considered to prevent pollution attacks. To solve the above problem, a novel privacy-preserving signature scheme is proposed.

5.1. Related Knowledge. In this subsection, some mathematical knowledge is related to the hardness and bilinear cyclic map is provided to support our scheme.

Suppose there are two bilinear cyclic groups H and H_T with the same prime number. They both have nondegeneracy and bilinearity and are computable. Generally, the security strength of privacy-preserving signature scheme depends on the hardness of the bilinear cyclic. For $F = (h, h^x)$, where $x \in \mathbb{Z}_p^*$, no polynomial algorithm can be calculated to obtain x.

5.2. Privacy-Preserving Signature Scheme. According to the above assumption of the cyclic group of the bilinear mapping, we give a complete privacy-preserving signature scheme including six probabilistic subalgorithms. The detailed scheme construction is given as follows:

(a) Construct a bilinear cyclic group $e : H \times H \mapsto H_T$, and output $k \longleftarrow^R \kappa$, $sk = \{sk_1, \cdots, sk_{m+n+1}\}$ such that $sk_i \longleftarrow^R \mathbb{F}_p$, and $pk = (\mu, h, H, H_T, g)$, where $\mu \longleftarrow^R H \setminus \{1\}$ and $g := \{h^{sk_1}, \cdots, h^{sk_{m+n+1}}\}$, where m and n are two random positive integers

(b) The resource data transmitted in the network is divided into n data blocks $w \in \prod_i$. $f : \{0,1\}^* \times \{0,1\}^* \times \kappa \mapsto \mathbb{F}_p$ is a random function; the encryption matrix is

$$G_E = \begin{bmatrix} e_{i,1} & & \\ & \ddots & \\ & & e_{i,m} \end{bmatrix}. \tag{18}$$

The coding vector is $c = (w_1, \cdots, w_m)$, and the payload of w is $w_p = (w_{m+1}, \cdots, w_{m+n})$. Then, $\text{Encrypt}(k, Id_i, w) = (c_E, w_p) \in \prod_i^E$, where $c_E = c \cdot G_E = (w_1 e_{i,1}, \cdots, w_m e_{i,m})$

(c) According to the above parameters sk and w, the signature of the data block σ can be given as follows:

$$\sigma = \mu^{\sum_{m+n}^{z=1} w_z sk_z + \left(\sum_m^{z=1} w_z\right) G(Id_i) sk_{m+n+1}}. \tag{19}$$

Then, the combined signature of the encoded data block is (w^r, σ^r), w^r, and σ^r are computed as follows:

$$\begin{aligned} w^r &= \sum_{j=1}^{\wp} \alpha_j w_j, \\ \sigma^r &= \prod_{j=1}^{\wp} \sigma_j^{\alpha_j}. \end{aligned} \tag{20}$$

The combined signature is $\xi^r = (Id_i, w^r, \sigma^r)$

(d) According to the given public key, the signature is verified as follows:

$$\begin{aligned} \upsilon_1 &= e(\sigma, h), \\ \upsilon_2 &= e\left(\mu \prod_{\tau=1}^{m+n} g_\tau^{w_\tau} \cdot \prod_{\tau=1}^{m} g_{m+n+1}^{G(Id_i) w_\tau}\right). \end{aligned} \tag{21}$$

When $\upsilon_1 = \upsilon_2$, the verification is successful

(e) The $e_{i,1}, \cdots, e_{i,m}$ is computed by the decryption matrix as

$$G_D = \begin{bmatrix} e_{i,1}^{-1} & & \\ & \ddots & \\ & & e_{i,m}^{-1} \end{bmatrix}. \tag{22}$$

The coding data is decrypted as $c_D = c_E \cdot G_D$. Then, the original block is calculated as follows:

$$w_D = \text{Decrypt}(k, Id_i, w) = \left(c_D, w_p\right) \tag{23}$$

5.3. Defense against Eavesdropping Attacks

Definition 3. When the i-th data block $\{w_i\}_{i=1}^q$ is maliciously eavesdropped, the probability function of the native data recovery is negligible.

Theorem 4. *The privacy-preserving signature scheme can resist the eavesdropping attacks.*

Proof. When the eavesdroppers collect the i-th linear combination of data blocks $\{w_i\}_{i=1}^q$, the data block can be analyzed as another expression as follows:

$$\begin{bmatrix} w_1 \\ \vdots \\ w_q \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1m} \\ \vdots & \ddots & \vdots \\ \alpha_{q1} & \cdots & \alpha_{qm} \end{bmatrix} \cdot \begin{bmatrix} eb_{i,1} \\ \vdots \\ eb_{i,m} \end{bmatrix}, \tag{24}$$

where $eb_{i,j} = (c_{i,j}, x_{i,j}) \in \mathbb{F}_p^{m+n}$ and $x_{i,j}$ is the native message.

Then, the encryption matrix of the encoding vector w_i can be expressed as follows:

$$C_E = \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1m} \\ \vdots & \ddots & \vdots \\ \alpha_{q1} & \cdots & \alpha_{qm} \end{bmatrix} \cdot G_E. \tag{25}$$

If the attackers want to successfully native message, $\alpha_{i,j} (1 \le i \le q, 1 \le j \le m)$ must be randomly selected from $\mathbb{F}_p$, and the randomly selected matrix is denoted as C_D. According to [27], the probability that the native message is recovered can be expressed as

$$P = P(C_D) \cdot P[\text{rank } (C_D) = m], \tag{26}$$

where $P(C_D)$ is the probability of decryption matrix is computed and $P[\text{rank } (C_D) = m]$ is the probability of matrix with same rank, which can be expressed as $\prod_{i=0}^{r-1}(1 - p^{i-r}) \le 1$. Meanwhile, $P(C_D) = 1/p^m$, then, the recover probability can be rewritten as $P = P[\text{rank } (C_D) = m]/p^m$.

Finally, the probability that the native message is recovered can be expressed as

$$P = \frac{\prod_{i=0}^{q-1}\left(1 - p^{i-q}\right) \le 1}{p^m} \le \frac{1}{p^m} \le \frac{1}{2^{\lambda m}}. \tag{27}$$

This completes the proof.

6. The Privacy-Preserving Mobile-Coverage Scheme Based on Trustworthiness

In Section 5, we proposed a privacy-preserving signature scheme and proved its effectiveness against eavesdropping attacks, after ensuring the security of data transmission

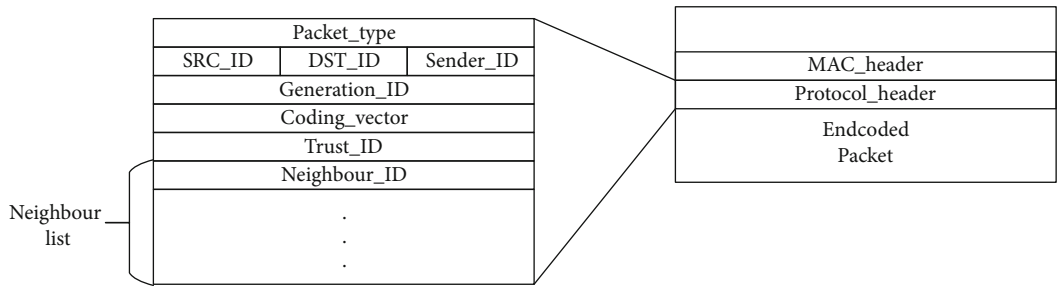

Figure 4: The flowchart of PP-MSTW.

during the movement. Then, how to select the safest mobile nodes to move in an iteration should be considered.

The privacy-preserving mobile coverage scheme based on trustworthiness (PP-MSTW) we proposed is shown in Figure 4. The mobile coverage scheme contains four phases, namely, destinations, mobility control, mobile coverage, and trust computation. Meanwhile, the privacy-preserving scheme will run during the network life when nodes exchange data. The data transmission format between nodes is shown in Figure 5. The data packet type, trust mark, neighbor node information, and the trust generation are defined in the transmitted data.

6.1. Full Construction. The definition of MCTS has be given in Section 4. Combined with the virtual force algorithm, the full construction of PP-MSTW is showed as follows:

(a) Setup (L^{λ}): a hybrid wireless sensor network is deployed in rectangular area A. The HWSN includes mobile and static sensor nodes. Moreover, the mobile nodes have higher energy than static nodes.

(b) Trust evaluation ($P \mapsto T_i, T_h$): suppose the collection of polluted nodes is $P = [M_1, \cdots, M_t]$, T_i of each node can be calculated by probabilistic trust against attacks in a generation, and the highest trust T_h will be given.

(c) Trust movement ($T_h \mapsto F_m$): the virtual force and Voronoi diagram are used to calculate the resultant force F_m experienced by the mobile nodes. Moreover, the resultant force F_m determines the distance and angle of movement.

(d) Trust update ($T_i \mapsto T_{i+1}$): after the sink node obtains the trust T_i of the previous generation, the highest trusted mobile node will perform the movement operation in a round G. When the movement is completed, the next-generation trust T_{i+1} will be calculated according to the new topology in the HWSNs.

(e) Maximum coverage: as the nodes of each generation move, the network coverage will gradually increase. When the network coverage reaches the highest value and does not change, the network deployment is completed.

As shown in Figure 4, the execution flowchart of PP-MSTW is as follows:

(1) Hybrid wireless sensor network initialization and two types of nodes are randomly deployed in the $L \times L$ rectangular region

(2) The trust of all nodes in the network will be calculated. The static node with the lowest degree of trust is confirmed as the next-generation pollution node

(3) The sink node selects the highest trusted mobile node to move within a generation G. The virtual force algorithm and the Voronoi diagram strategy determine the moving direction and distance of the node

(4) After performing the above operations, the trust of each node will be updated according to the location of the next-generation pollution node and the updated network topology

(5) As long as there is traffic between nodes in the network, the privacy-preserving scheme will be implemented

The MSTW algorithm shows the specific execution process of trust mobile, which will give updated trust U_m and network coverage p.

6.2. Simulation Result Analysis. The PP-MSTW is composed of two parts including privacy-preserving signature scheme and mobile coverage based on trustworthiness (MSTW) algorithm. In Subsection 5.3, the theoretical correctness of the signature scheme is proven. In this subsection, the communication and computation overhead of the MSTW are analyzed in detail. Simulation experiment is constructed in the softwares OMNET++ and MATLAB.

6.2.1. Communication Overhead. According to the definition of communication trust, the communication overhead can be expressed as follows:

$$O_{\text{communication}} = \frac{[(n(s) + l + 1)/(n(f + s) + l + 2)]^2}{N}, \quad (28)$$

where s is the number of successful data exchanges between nodes in each generation and f represents the failures. Meanwhile, the sink node will calculate the position of the next generation of nodes according to the virtual force algorithm, and the mobile signal l will be sent to the mobile node that needs to move. Generally, mobile nodes have the highest trust.

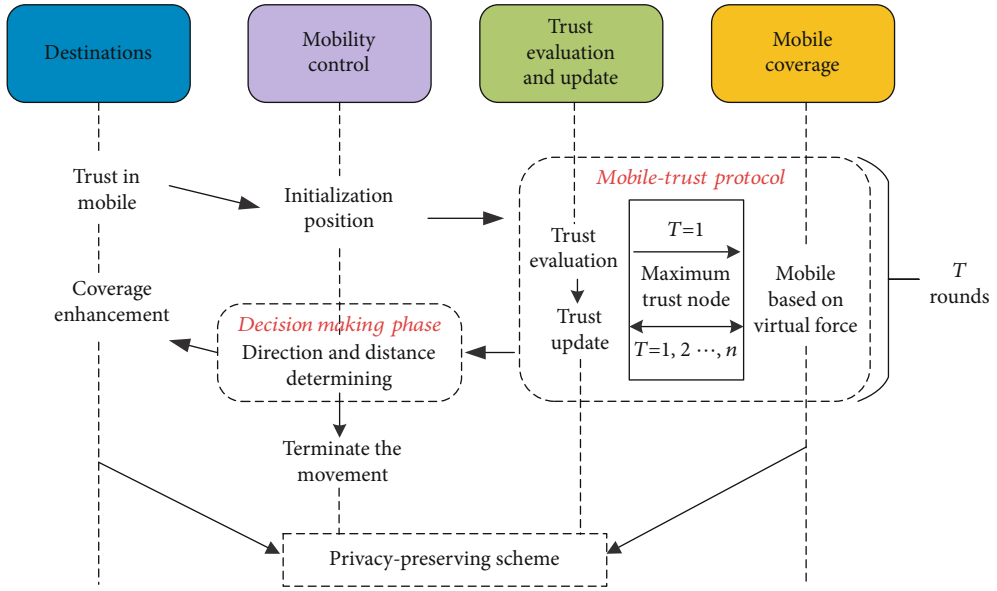

Figure 5: The definition of transmitted data format.

1: **Input:** Static nodes $\boldsymbol{S}_s = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_s\}$ and mobile nodes $\boldsymbol{S}_m = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_m\}$, in which $s + m = N$ and an $L \times L$ initial rectangular region, coverage difference accuracy κ, communication radius r_c, sensing radius r_s, $r_c = 2r_s$, iterations $\boldsymbol{G}$, all network sensor nodes initialization trust $\boldsymbol{U} = \{\boldsymbol{U}_{s_1}, \boldsymbol{U}_{s_2} \cdots, \boldsymbol{U}_{s_s}, \cdots, \boldsymbol{U}_{s_{s+1}}, \boldsymbol{U}_{s_{s+2}}, \cdots, \boldsymbol{U}_{s_{s+m}}\}$.
2: **Output:** The mobile sensor nodes trust set $\boldsymbol{U}_m$ and the network coverage $\boldsymbol{p}$.
3: **for** $i = 0, 1, \cdots, \boldsymbol{G}_n$ **do**
4: According to the virtual force and voronoi theory, the combined force $\boldsymbol{F}_s = \{\boldsymbol{F}_{s_1}, \boldsymbol{F}_{s_2}, \cdots, \boldsymbol{F}_{s_m}\}$ of mobile nodes in a rectangular region $L \times L$ is calculated.
5: Select Max $(\boldsymbol{U}_s)$; Where there are first generation pollution nodes in the network.
6: **while** $\boldsymbol{p}_{\boldsymbol{G}_i} - \boldsymbol{p}_{\boldsymbol{G}_{i-1}} \le \kappa || \boldsymbol{F}_{s_m} = \varnothing$, stop.
7: Calculate the mobile nodes trust $\boldsymbol{U}_m$.
8: Select Max $(\boldsymbol{U}_m)$ to move for avoiding the high-risk area and improving the network coverage $\boldsymbol{p}$.
9: **end for**

Algorithm 1: Mobile scheme based on the trustworthiness.

The growth rate of communication overhead about the MSTW is shown in Figure 6. As the sink node sends more and more mobile signal, the growth rate of communication overhead in the HWSNs is the lowest compared to previous studies. In HWSNs, the number of successful communications between nodes is much higher than the number of failures.

6.2.2. Computation Overhead. Network computation overhead is mainly divided into four parts including communication trust, energy trust, probabilistic trust against attacks, and virtual force. All calculations are performed on $\mathbb{F}_q$, where the number of multiplications implies the computation overhead.

In communication trust, Equation (5) shows that communication trust needs to be calculated $n^2 - n + 1$ times on a finite filed $\mathbb{F}_q$, and so the computation overhead is $O_c = n^2 - n + 1$.

In energy trust, the computation overhead in energy trust can be easily derived as $O_e = n^4$.

Our scheme is compared with other trust computing schemes; ours mainly includes probabilistic trust against attacks. According to the above formula for resisting attacks probability, the computation overhead can be expressed as follows:

$$O_a = n\left(qr^4 - pr^3 + mr^2\right). \tag{29}$$

In addition to the computation overhead in the trust evaluation process, the computation overhead of the trust mobile algorithm should also be considered. The computation overhead in trust mobile mainly includes virtual force algorithm. Research [28] shows that the computation overhead of the virtual force algorithm is $O_v = n(n-1)$.

Finally, the overall computation overhead of the MSTW within a generation G is expressed as follows:

$$\begin{aligned} O_{\text{computation}} &= O_c + O_e + O_a + O_v \\ &= n^4 + 2n^2 + n\left(qr^4 - pr^3 + mr^2 - 2\right) + 1. \end{aligned} \tag{30}$$

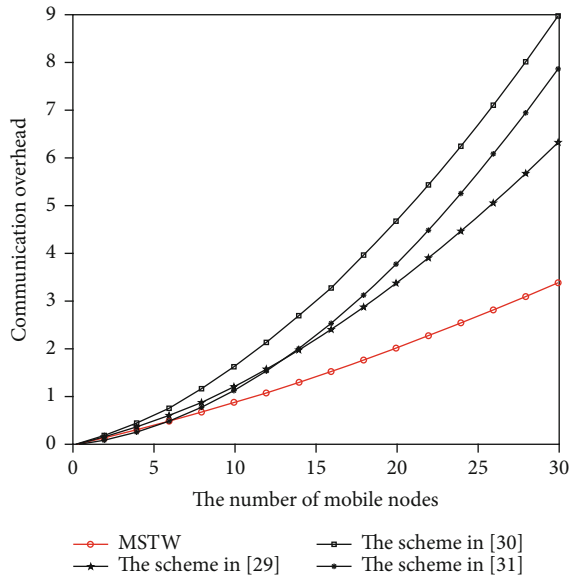

Figure 6: The communication overhead of different schemes.

Table 1: Computational comparison of different schemes.

Items	Computational complexity
The scheme in [29]	$n^4 + 2n^2$
The scheme in [30]	$n^4 + n^2 + kn + p$
MSTW	$n^4 + 2n^2 + n(qr^4 - pr^3 + mr^2 - 2) + 1$
The scheme in [31]	$n^4 + n^2 + kn$

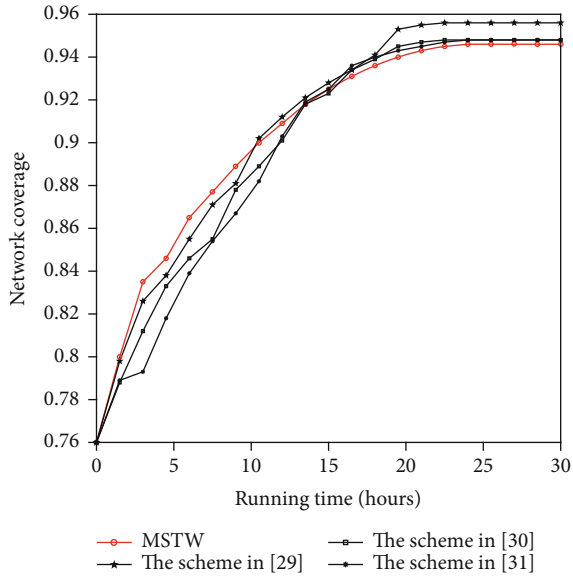

Figure 7: The coverage of different schemes in [29–31].

Table 1 demonstrates the comparison of computation overhead, in which our MSTW has almost the same computation overhead as several other previous schemes, and both are approximately equal to $O_{\text{computation}} \approx O(n^4)$. Figure 6 shows our scheme has the lowest communication overhead in each generation of data communication. In Figure 7, the maximum coverage that all nodes in HWSNs can reach is given during the network life. Compared with the previous schemes, the network coverage of our scheme is 1.3% lower than the maximum coverage achieved by the previous schemes. In summary, our MSTW can guarantee a high coverage under the condition of deployment security and does not increase the computation and communication overhead at the same time.

7. Conclusion

For security HWSN deployment, a novel privacy-preserving mobile coverage scheme based on the trustworthiness is proposed. The scheme can ensure the communication data integrity and confidentiality in the network coding communication. Firstly, a comprehensive trust evaluation method based on historical communication data, energy, and the probability of nodes being attacked is constructed. Then, a privacy-preserving signature scheme is applied to the network for resisting pollution and eavesdropping attacks. Finally, the PP-MSTW is constructed to maximize network coverage under the premise of ensuring the security of node communication.

From analyzing the mathematical theory and result of simulation, the PP-MSTW we proposed can guarantee the security in data communication under the theoretical analysis. The communication and computation overhead of the scheme are lower than those of the previous algorithms. Moreover, the scheme we proposed can obtain the optimal solution for coverage under the premise of security in the network deployment.

In the future research, we will use artificial intelligence methods to analyze network trust and then study game theory methods to adjust network defense strategies and ultimately further improve network robustness.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Key Research and Development Program of China (No. 2018YFB2100400).

References

[1] M. R. Senouci and A. Mellouk, "A robust uncertainty-aware cluster-based deployment approach for WSNs: coverage, connectivity, and lifespan," *Journal of Network and Computer Applications*, vol. 146, p. 102414, 2019.

[2] R. R. Priyadarshini and N. Sivakumar, "Enhancing coverage and connectivity using energy prediction method in

underwater acoustic WSN," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 2751–2760, 2020.

[3] P. Le Nguyen, K. Nguyen, H. Vu, and Y. Ji, "Telpac: a time and energy efficient protocol for locating and patching coverage holes in wsns," *Journal of Network and Computer Applications*, vol. 147, article 102439, 2019.

[4] S. Shao, L. Wu, Q. Zhang, N. Zhang, and K. Wang, "Cooperative coverage-based lifetime prolongation for microgrid monitoring WSN in smart grid," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, Article ID 249, 2020.

[5] P. Natarajan and L. Parthiban, "k-coverage m-connected node placement using shuffled frog leaping: Nelder–Mead algorithm in WSN," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–6, 2020.

[6] A. Verma, S. Kumar, P. R. Gautam, T. Rashid, and A. Kumar, "Broadcast and reliable coverage based efficient recursive routing in large-scale WSNs," *Telecommunication Systems*, vol. 75, no. 1, pp. 63–78, 2020.

[7] M. S. Abdalzaher and O. Muta, "A game-theoretic approach for enhancing security and data trustworthiness in IoT applications," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11250–11261, 2020.

[8] N. A. Khalid, Q. Bai, and A. Al-Anbuky, "Adaptive trust-based routing protocol for large scale WSNs," *IEEE Access*, vol. 7, pp. 143539–143549, 2019.

[9] X. Yu, F. Li, T. Li, N. Wu, H. Wang, and H. Zhou, "Trust-based secure directed diffusion routing protocol in WSN [J]," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–3, 2020.

[10] K. Cho and Y. Cho, "HyperLedger fabric-based proactive defense against inside attackers in the WSN with trust mechanism," *Electronics*, vol. 9, no. 10, p. 1659, 2020.

[11] A. Chowdhury, G. Karmakar, J. Kamruzzaman, and S. Islam, "Trustworthiness of self-driving vehicles for intelligent transportation systems in industry applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 961–970, 2020.

[12] Y. Yoon and Y. H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1473–1483, 2013.

[13] F. Bao and R. Chen, "Trust management for the internet of things and its application to service composition," in *2012 IEEE international symposium on a world of wireless, mobile and multimedia networks (WoWMoM)*, pp. 1–6, San Francisco, CA, USA, 2012.

[14] W. Yang, Z. Zheng, G. Chen, Y. Tang, and X. Wang, "Security analysis of a distributed networked system under eavesdropping attacks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 7, pp. 1254–1258, 2019.

[15] P. Zhang, Y. Jiang, C. Lin, Y. Fan, and X. Shen, "P-coding: secure network coding against eavesdropping attacks," in *2010 Proceedings IEEE INFOCOM*, pp. 1–9, San Diego, CA, USA, 2010.

[16] Y. J. Chen and L. C. Wang, "Privacy protection for internet of drones: a network coding approach [J]," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1719–1730, 2018.

[17] M. Nikravan, A. Movaghar, and M. Hosseinzadeh, "A lightweight defense approach to mitigate version number and rank attacks in low-power and lossy networks," *Wireless Personal Communications*, vol. 99, no. 2, pp. 1035–1059, 2018.

[18] L. Huang and Q. Zhu, "A dynamic games approach to proactive defense strategies against advanced persistent threats in cyber-physical systems," *Computers & Security*, vol. 89, p. 101660, 2020.

[19] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 3, pp. 1–37, 2008.

[20] J. Kaur and S. Kaur, "Novel trust evaluation using NSGA-III based adaptive neuro-fuzzy inference system," *Cluster Computing*, pp. 1–12, 2021.

[21] X. Li, F. Zhou, and J. Du, "LDTS: a lightweight and dependable trust system for clustered wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 924–935, 2013.

[22] 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.

[23] C. P. Chen, S. C. Mukhopadhyay, C. L. Chuang et al., "A hybrid memetic framework for coverage optimization in wireless sensor networks," *IEEE transactions on cybernetics*, vol. 45, no. 10, pp. 2309–2322, 2015.

[24] K. P. Naveen and A. Kumar, "Coverage in one-dimensional wireless networks with infrastructure nodes and relay extensions," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 140–153, 2019.

[25] L. Cao, Y. Yue, Y. Cai, and Y. Zhang, "A novel coverage optimization strategy for heterogeneous wireless sensor networks based on connectivity and reliability," *IEEE Access*, vol. 9, pp. 18424–18442, 2021.

[26] A. Saidi and K. Benahmed Pr, "Secure cluster head election algorithm and misbehavior detection approach based on trust management technique for clustered wireless sensor networks," *Ad Hoc Networks*, vol. 106, p. 102215, 2020.

[27] D. Laksov and A. Thorup, "Counting matrices with coordinates in finite fields and of fixed rank," *Mathematica Scandinavica*, vol. 74, pp. 19–33, 1994.

[28] C. Qi, J. Huang, X. Liu, and G. Zong, "A novel mobile-coverage scheme for hybrid sensor networks," *IEEE Access*, vol. 8, pp. 121678–121692, 2020.

[29] Z. Liao, J. Wang, S. Zhang, J. Cao, and G. Min, "Minimizing movement for target coverage and network connectivity in mobile sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 7, pp. 1971–1983, 2014.

[30] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, "Rearrangement of mobile wireless sensor nodes for coverage maximization based on immune node deployment algorithm," *Computers & Electrical Engineering*, vol. 43, pp. 76–89, 2015.

[31] Z. Fu and K. You, "Optimal mobile sensor scheduling for a guaranteed coverage ratio in hybrid wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 4, Article ID 740841, 2013.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6648351, 12 pages
https://doi.org/10.1155/2021/6648351

Research Article

Network Intrusion Detection System Based on the Combination of Multiobjective Particle Swarm Algorithm-Based Feature Selection and Fast-Learning Network

Sajad Einy,[1,2] **Cemil Oz**,[1] **and Yahya Dorostkar Navaei**[3]

[1]*Computer Engineering Department, Sakarya University, Turkey*
[2]*Application and Research Center for Advanced Studies, Istanbul Aydin University, Turkey*
[3]*Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

Correspondence should be addressed to Yahya Dorostkar Navaei; y.dorostkar@qiau.ac.ir

Received 19 November 2020; Revised 8 May 2021; Accepted 2 June 2021; Published 16 June 2021

Academic Editor: Yanhui Guo

Given the growth of wireless networks and the increase of the advantages and applications of communication networks, especially mobile ad hoc networks (MANETs), this type of network has attracted the attention of users and researchers more than before. The benefit of these types of networks in various kinds of networks and environments is that MANET does not require to hardware infrastructure to communicate and send and receive data packets within the network. It is one of the main reasons for using these MANET in various fields. On the other hand, the increased popularity of these networks has led to many challenges, one of the most important of which is network security. In this regard, a lack of regulatory and security infrastructure in MANETs has caused some problems in sending and receiving data, where intrusion in the network has been recognized as one of the most important issues. In MANETs, wireless notes act as a link between the source and destination nodes and play the role of relays and routers in the network. Therefore, malicious node penetration and the destruction of information packages become feasible. Today, intrusion detection systems (IDSs) are used as a solution to deal with the problem through remote monitoring of the performance and behaviors of nodes existing in wireless sensor networks. In addition to detecting malicious nodes in the network, IDSs can predict the behavior of malicious nodes in the future in most cases. Therefore, the present study introduced a network IDS (NIDS) entitled MOPSO-FLN by using a combination of multiobjective particle swarm optimization algorithm-(MOPSO-) based feature subset selection (FSS) and fast-learning network (FLN). In this work, we used the KDD Cup99 and dataset to select features, train the network, and test the model. According to the simulation results, this method was able to improve the performance of the IDS in terms of evaluation criteria, compared to other previous methods, by creating a balance between the objectives of the number of representative features and training errors based on the evolutionary power of MOPSO.

1. Introduction

Mobile ad hoc networks (MANETs) are a group of mobile nodes that communicate over wireless links without any backbone. MANETs do not have centralized control mechanism, and each mobile node acts as routers for transferring data packages to other specific nodes of the network in addition to being a terminal in the network [1]. Security is a paramount concern in MANETS due to having a dynamic topology and nodes that can easily enter or exit the network at any time and have access to data flow throughout the network. In addition, several mobile nodes are limited to resources in terms of computational power and energy source [2, 3]. As such, the presence of permanent security monitoring nodes in the network is almost impossible due to limited resources, and there is a need for remote control of nodes' behavior in the network and determining security necessities in MANET [4, 5].

Network intrusion detection systems (NIDS) are used to monitor node activity or network traffic activity. The main goal of NIDSs is detecting malicious nodes and predicting possible future attacks on the network [6]. An alert is

generated for further action when detecting a malicious node in the network. Various techniques have been proposed for detecting attacks by NIDS, and it is notable that an IDS's success depends on the type of technique used in this regard [7]. One of the key factors in NIDS performance is the selection of representative features from the main dataset [8]. Reducing the number of features existing in the data set (e.g., the behavior of nodes and network traffic) without affecting the classification precision can play an important role in IDS performance optimization [9].

In the proposed method, the feature selection approach based on MOPSO (multiobjective particle swarm optimization) is responsible for selecting representative features of the main dataset. The selected features are entered into a fast-learning network (FLN) as a solution. Using rapid training, FLN evaluates solutions and determines the model's error based on the selected features. The main goal of the present study was eliminating unrelated features and attributes and plugins to reduce the dimension of the data and complexity of the model while increasing its classification accuracy in determining the model of malicious nodes and network attacks at a higher pace. Therefore, the proposed method included the feature selection approach based on MOPSO to determine important features.

In this technique, features are considered as the primary particles of a multiobjective particle swarm optimization algorithm (PSO). In addition, the target function is applied to minimize the number of features used and the class error in the proposed method. Moreover, the primary particles are selected as a subset of the entire features in the database. The particles are sent to FLN in MOPSO as a solution in order to estimate the value of the fit function. The particles with the smallest function value in each repetition of the MOPSO algorithm are selected as expert particles and the optimal solutions in that round and are stored in the repository of solutions. In the next round, the location and speed of other particles are updated in line with the tendency towards expert particles. At the end of the algorithm, the expert particles that were stored in the repository of solutions are sorted by the value of the fit function, and a solution with the smallest value is selected as the optimal solution. Moreover, the features determined by the optimal solution are determined as the behavioral pattern of malicious nodes and used to predict future malicious nodes in the network.

2. Related Works

The need for high-speed services in providing network services has always been a necessity of networks and no further emphasis is required in this regard. NIDS is an important tool for protecting the network [1]. These systems analyze the entry paths of nodes into the system regarding protected systems and decide whether the entry routes contain nodes from an attack or not [10, 11]. NIDS raises an alert in case of detecting an attack. Therefore, this part of the article is dedicated to the assessment of some NIDSs considered in publications.

A monitored NIDS is a system that can learn from training samples during previous attacks to detect new attacks. Application of intrusion detection based on artificial neural networks (ANNs) is effective for reducing the number of samples that are classified falsely (false positive negative or false negative) owing to ANNs' ability to learn from actual samples. In 2019, Ali et al. proposed a developed training model for neural networks entitled a fast-learning network (FLN). The method is presented based on feature selection according to optimized PSO with the title of PSO-FLN [12]. Selvakumar and Muneeswaran (2019) developed a feature subset selection (FSS) approach with the use of a firefly algorithm, which affects the speed of classification model analysis [13]. In 2018, Chiba et al. proposed a NIDS based on the backpropagation neural network (BPNN) using an advanced learning algorithm. These scholars applied a new architecture for the network, the function of which affected anomaly detection, including feature selection, normalization of the data, architecture of the neural network, and activation function [14]. Al-Azam et al. (2020) proposed a wrapper-based feature selection algorithm for IDS, which used a feature selection pigeon optimizer algorithm. A new method has been offered for achieving a feature selection optimization technique to be combined with classification methods and has been compared with traditional feature selection methods based on collective intelligence algorithms [15]. In 2020, Zhou et al. presented a new framework based on feature selection and group learning techniques to detect intrusion. In this method, a metaheuristic algorithm entitled CFS-BA is suggested for reducing sizes. Afterwards, a group classification approach is combined with C4.5, random forest (RF), and feature penalty-based forest (Forest PA), and the classification voting method was applied to detect the attack [16]. In 2010, Senthilnayaki et al. presented a new feature selection algorithm using the max-dependency max-significance algorithm. The algorithm is used to select a minimum number of features from the data set. In addition, a new algorithm is proposed based on k-nearest neighbors to classify datasets [17].

3. Proposed Method

In this article, a method is proposed for NIDS based on the combination of MOPSO and FLN-based FSS. The proposed method used the KDD Cup 99 dataset to determine intrusion detection patterns in the network and evaluate the model. Features are primarily selected to decrease classification difficulty and increase classification accuracy by selecting related features. In single-objective feature selection tasks, feature selection has a goal for optimization. Feature selection is mainly carried out to find the best combination of features for the most optimal classification performance. Multiobjective feature selection (MOFS) is responsible for feature selection by turning it into a multiobjective optimization problem, where the goal is to create a balance for optimizing multiple goals. The main objectives of this optimization method for FSS include reducing the number of features based on the class label and decreasing attack detection errors in the network, which will increase the accuracy of predicting test samples. As a result, a solution for the multiobjective feature selection optimization problem is a set of dominant

solutions, in which each solution is a vector of two components, number of features, and classification error rate. The goal was to minimize the number of unrelated features by using the feature selection problem as a minimization issue, thereby minimizing the classification error rate. The proposed method is formulated below. The flowchart of the proposed method is shown in Figure 1.

3.1. Formulation of Proposed Method. Kennedy and Eberhart first reported PSO in 1995, inspired by flock behavior. The main objective of PSO is to find the optimal solution in the search space of a target function similar to a flock's search pattern in the quest for the best food source. A set of generated particles randomly search for the best solutions in PSO. In this regard, a search is carried out by particles' adjustment of their own flight speed and direction based on Equations (1) and (2), respectively [18].

$$x_{id}(\mathrm{t}+1) = x_{id}(t) + v_{id}(t+1), \tag{1}$$

$$\begin{aligned} v_{id}(t+1) = {} & w \times v_{id}(t) + r_1 \times c_1 \times [p_{id}(t) - x_{id}(t)] \\ & + r_2 \times c_2 \times [g_{id}(t) - x_{id}(t)], \end{aligned} \tag{2}$$

where id is the number of dimensions, w is the inertia weight, which controls particle exploration, r_1 and r_2 are random numbers in the range of zero-one ($r_1, r_2 \in [0, 1]$), and c_1 and c_2 are acceleration constants used to control the effect of personal and overall best particles. In addition, P_{id} is the best personal position for a particle (P_{best}), and g_{id} shows the best overall position found by neighbors ($\mathrm{g}_{\mathrm{best}}$).

Obviously, PSO has high-speed convergence ability in single-objective problems, which is a favorable choice for MOPs. The Pareto-optimal solution is used in the design of MOPSO to generate a set of leaders who control the direction of particle flight and direct the search process toward optimal condition. In addition, the dominant solutions found are stored in the overall external memory (called repository) and are later used by particles as global leaders. The global guidance is selected using the roulette wheel method and based on hypercubes. Moreover, MOPSO adopts a geography-based strategy to maintain solution diversity.

In fact, the external repository "archive" includes two sections: a controller and a network. The goal of the controller is deciding whether a new solution can be added to the archive or not; updating or pruning the archive depends on the dominant relationship. Nevertheless, an adaptive network method is called whenever the archive is full. In contrast, the network is used to promote diversity among solutions. In fact, the target space is divided into areas known as a hypercube. In general, the hypercube is geographical regions encompassing a number of solutions created based on target functions. A fit function is assigned to each hypercube based on the number of existing particles. Therefore, hypercubes with a very large number of particles have less fit value. A hypercube is selected using the roulette wheel method, followed by selecting a random particle from the hypercube. Therefore, the network facilitates the process of selecting solutions in low-population areas in the target space, compared to samples locating in swarm regions [18, 19]. The speed updating function is as follows:

$$\begin{aligned} v_{id}(t+1) = {} & w \times v_{id}(t) + r_1 \times c_1 \times [p_{id}(t) - x_{id}(t)] \\ & + r_2 \times c_2 \times [REP(h) - x_{id}(t)]. \end{aligned} \tag{3}$$

In Equation (3), REP(h) is a dominant solution for selection from the repository, where the index h is selected based on the value of the fit function of hypercubes. For example, we can consider h as a set of features whose REP(h) is the accuracy of classification problem with these features.

In this research, the feature selection problem was considered as a multiobjective optimization issue resolved by using a MOPSO. Accordingly, the number of features selected shows the dimensions of the problem independently with a binary search space in the range of zero-one. Given the fact that MOPSO is an initial population-based metaheuristic method, the primary population complies with the potential subset of features that are directly related to class labels. In the two-dimensional search space in the problem, the first dimension of x_1 is a real positive number that shows the error rate, whereas the second dimension of x_2 is a real positive number that exhibits the number of features. In addition, F function leads to a balanced set of decision-making vectors, which reduce both error rate and the number of features presented by $[F(x_1), F(x_2)]$.

In the proposed method, the initial population is adjusted to the features existing in the KDD-CUP dataset. Therefore, each particle is a binary vector, the element of which is equal to the number of features, and each element refers to one feature in the dataset. Therefore, each particle existing in the swarm shows feature selection with a value of one. As such, the length of a particle is equal to the number of features existing in the dataset. Figure 2 exhibits an example of a representation of a particle, where the number of features existing in the dataset was estimated at 41.

As shown, each particle in PSO is considered a set of features existing in the dataset. In this regard, a number of elements of the vector can be randomly zero and one. The elements with zero value indicate unselected features, whereas elements with one value show feature selection related to the element. As a result, it is clear that the selected subset of features includes the feature set of $[F_2, F_6, F_8, F_{41}]$.

In the proposed method, the transfer function of Sigmoid (S) was used to define the probability of feature selection or lack of selection to select features for primary particle vectors. The function of this function is such that if the random probability is less than the threshold value of the transfer function, which is generally considered to be 0.5, the value of zero will be allocated to this feature in the related element. Otherwise, the value one will be recorded for the features, and the desired feature will be assessed based on the objective function. In this method, sigmoid is used as a random function to select or not to select a feature in a solution. So, we used this threshold value to remove features that are less than half probability to be selected and features that are more than half probability to be selected. The primary location and velocity of each particle are determined by assessment functions following selecting the initial population based on the nature

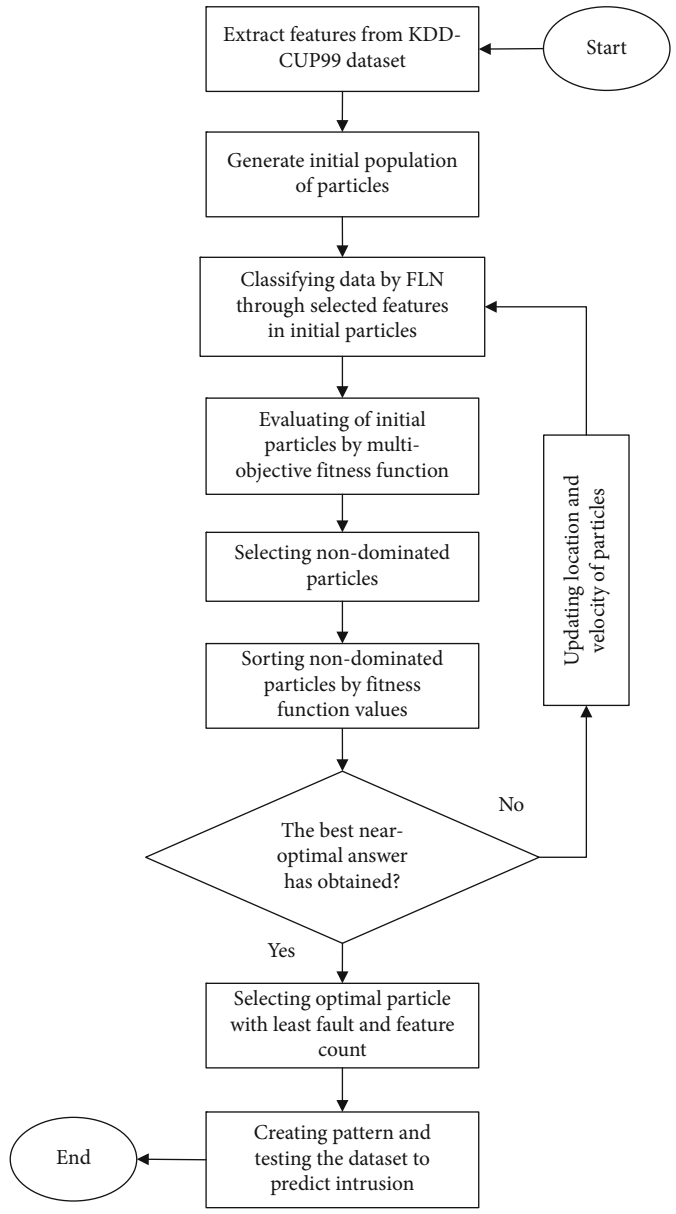

Figure 1: Flowchart of proposed method.

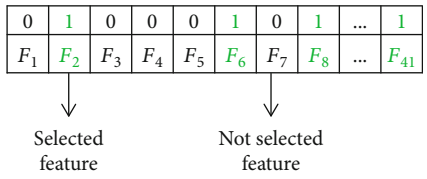

Figure 2: A representation of the initial population vector of the dataset sample.

of PSO. In this technique, the location of each particle is considered as features selected from the features existing in the dataset, and the velocity of each particle is regarded as the convergence rate to high classification rate and reduction of classification error. The features with the highest assessment function value and higher surrounding particle swarm are selected as the output of the primary feature selection stage. The best particle location and velocity results are stored at this stage and the particle position is updated. The process continues until reaching a final response that creates a balance between goals.

3.2. Proposed Objective Function. As mentioned before, the proposed method applied the MOPSO to select a subset of features selected for the class label. In this technique, multiple objectives are combined, and two general categories of features are obtained in the end in the form of minimization. Moreover, the selected subsets of features are assessed based on two main objectives of reducing the number of features and classification error rates. The fit function is presented in the form of Equation (4) in line with the evaluation of the initial population, selection of the population of experts,

and finding particles with the largest weight.

$$\text{Minimize } F(x) = \begin{cases} f_1(x) = \dfrac{L}{A}, & L \in A,\ A \in \mathbb{R}^+, \\ f_2(x) = 1 - \dfrac{FP + FN}{P + N}, & (P + N) \in \mathbb{R}^+, \end{cases} \tag{4}$$

where L is the number of features selected from datasets and A is the total number of features. The criteria related to the confusion matrix was used to evaluate the error rate of each particle based on the features selected in each step, where true positive (TP) is used to show the category of normal nodes that are accurately detected to be normal by the classification model and based on the selected features. In addition, false positive (FP) is applied to demonstrate the category of normal nodes that are falsely detected as an intrusion by the classification model and based on the selected features. Moreover, true negative (TN) is used to show the category of intrusion nodes that are accurately detected by the classification model and based on the selected features. Finally, false negative (FN) is applied to indicate the category of intrusion nodes that are falsely detected to be normal by the classification model and based on the selected features. In Equations (3)–(11), P is equal to TP + FN, and N is equal to FP + TN. The first target function $f_1(x)$ is related to the ratio of selected features to the total features existing in the dataset, whereas the second function $f_2(x)$ is used to evaluate the classification error rate.

3.3. FLN-Based Classification. FLN is a parallel connection from a leading single-layer network and a three-layer neural network containing three inputs, hidden and output layers. In general, FLN is an artificial neural network, which is a double parallel forward neural network (DPFNN), where the classification error rate is estimated using an analysis approach called the least-square technique [12]. This describes a multilayer parallel connection and a single-layer neural network. As discussed earlier, optimal particles in MOPSO show the subset of features selected from the dataset, which are entered into FLN as input. In the proposed method, FLN is located at the core of MOPSO and is responsible for estimating the classification errors of solutions. In FLN, the coded solutions are transferred to the middle layer through the input layer, where they are weighted and trained. The main difference between FLN and neural networks is the lack of waiting for training weights by all hidden players in FLN and transferring bias amounts and weights to the output layer in each layer of hidden layers where the amount of error is less than the specified threshold. This allows an increase in the speed of learning and classification of solutions in addition to preventing overfitting in the network.

In the proposed method, initial solutions that are randomly selected in MOPSO are classified by FLN in addition to assessing the number of features and the importance of selected features in the fitness function, and their error value is specified as the second goal in the multiobjective fitness function. The particles with the lowest number of important features and lowest classification error rates are selected as expert particles and nondominated solutions in each stage and are stored in the expert solutions repository. In the next stage of the proposed algorithm, the error threshold amount is considered based on the error rate of optimal particles in the previous phase. The solutions entered into FLN in each hidden layer that applies to the threshold condition are directed toward the exit.

As mentioned, the proposed method uses fast-learning networks to identify malicious nodes. Fast-learning networks are expanded from artificial neural networks, so, it has inherited the training properties of the model. In neural networks, for the purpose of classifying or detecting malicious nodes, each input feature is examined, and a weight is assigned to each input feature. These weights in each neuron indicate the importance of the value of that feature in determining malicious node. Thus, the amount of weights in neurons is considered as a pattern for identifying malicious nodes. Given that the nodes are in two classes, normal and malicious nodes, a pattern is created for each class by the proposed fast-learning networks.

In the proposed method, the fitness function is used to select a feature subset in the training set. Once, the optimal feature subset in the training data set has been selected, and the minimum amount of intrusion detection error using this data set has been ensured; the same feature subset selection pattern can be used in the test dataset. In other words, the feature subset that has been selected in the training set as the optimal feature subset is used as a template, and among the test data set, only this feature subset is selected to predict intrusions in test set by FLN.

3.4. Complexity Analyzing of Proposed Method. The time complexity of the proposed method can be analyzed as follows. If N is the total number of features in the dataset, each particle as a solution selects n ($1 \le n \le N$) features of all features in the dataset. If the number of iterative generations to find the optimal answer be M for the proposed problem, so a maximum of $M * N$ iterations is required for the problem. On the other hand, since a fast-learning neural network is used at the core of the feature selection method, the complexity of neural networks is order of $O(n^2)$ [20], where n is equal to the number of input features. The fast-learning network method transmits the results to the output layer when the desired accuracy is achieved and does not wait for the completion of the training steps. Hence, the time order of this method is less than $O(n^2)$. Therefore, it can be concluded that the time complexity in the proposed method is the maximum the order of $O(\mathrm{MN}^2)$. In case of more features in the dataset, the time complexity of the proposed method will also increase.

4. Implementation

In the proposed method, primary particles are first selected randomly from the dataset of KDD-CUP99 [21]. In this data set, there are more than 54,000 instances of node connections in the network in which each has 42 features and has classified into two classes of normal nodes and malicious nodes.

Table 1: A part of the initial particle population matrix.

Particle #	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
1	1	0	1	0	0	1	1	1	1	1	1	1	1	0
2	1	1	0	1	0	1	1	1	0	0	0	1	0	1
3	0	0	1	0	1	1	1	0	1	0	1	0	1	1
4	1	1	1	0	0	0	1	0	0	0	0	1	0	1
5	1	0	0	1	1	0	1	0	0	0	1	1	0	0
6	0	1	1	0	1	0	1	0	1	1	1	1	1	1
7	0	0	1	1	1	1	1	1	0	0	0	0	0	0
8	1	1	1	1	0	1	1	0	1	1	0	0	1	1
9	1	1	1	1	0	0	0	1	1	0	0	1	0	1
10	1	1	1	0	0	1	1	0	1	1	0	1	1	1

The size of the initial population is defined by the number of features existing in the dataset and as an n-dimensional vector. A 100×42-dimensional matrix of random numbers in the range of (0,1) is created to determine the initial population. The $A_{i,j}$ element of the matrix shows the possibility of the presence of the j_{th} feature in the i_{th} solution (particle). According to the Sigmoid function applied in the proposed method, the values of the initial population matrix elements are converted to binary based on a threshold of 0.5. In other words, if the $A_{i,j}$ element has a value below the threshold, the j_{th} feature will not exist in the i_{th} solution (particle). On the other hand, the j_{th} feature of one of the subsets of selected features will have the ith solution (particle) if the value of the mentioned element is above the threshold. Table 1 shows initial particle population.

As observed in Table 1, the initial particle population is distributed in the proposed method in a binary form, and each of the initial populations shows a solution for selecting a subset of features existing in the KDD dataset. According to the adjusted parameters, the initial population matrix is used as an input of MOPSO. In addition, the proposed algorithm evaluates the initial population based on the fitness function in the first step while considering the adjusted parameters and the initial population. Afterwards, it obtains the level of competency of each solution. Therefore, the number of solutions found will increase with more repetitions of the stages, and iterations will continue until reaching the cessation condition. Ultimately, the presented solutions are assessed, and the best solution is selected among the existing solutions. MOPSO receives the initial population and evaluates its competency. In the first step, the algorithm finds the nondominated solutions or dominant solution, saving them in the solutions repository. In the next phase, other solutions and particles are directed toward the solution. Accordingly, a number of dominant solutions may be found in each stage, the value of the fitness function of which might be higher than the threshold and are stored in the respiratory. It is notable that in the proposed method, the amount of competency is equal to the aggregation of two objectives, and the higher the competency of a solution, the lower the number of selected features and the highest the accuracy of classification and intrusion detection by using the features based on FLN class. Therefore, the dominant solutions improve both objectives used in the proposed method. Figure 3 shows the distribution of solutions in the problem space and the dominant solutions in the first step.

As observed in Figure 3, the solutions are randomly distributed in the problem space in the first step of MOPSO in the proposed method. The problem space includes two objectives of F1 and F2, with the former existing to reduce the number of features selected from all features in the dataset and the latter corresponding to the vertical axis of Figure 3 to reduce classification error rate based on the selected features. Given the random selection of the initial particle population and since the nature of particles is binary, the existence or lack of existence of a feature in each particle can affect the results obtained for each solution. Therefore, with regard to the problem space, the Pareto front tends to the origin of the coordinates where both objectives are minimal.

Eventually, it could be expressed that the 6 important features based on the multiobjective fitness function in the MOPSO has selected. The selected features have indexes {2,7,13,19,26,27} entitled {"Srv_count", "Count", "Wrong_Fragment", "land", "ds_host_srv_serror_rate", and "dst_same_srv_rate"}. These features have the greatest impact on the node class label, and based on these features, network intrusion can be detected with the highest accuracy and least complexity.

Due to the fact that different feature selection methods use different policies to select the feature subset, they can therefore select different subsets of feature as the representative feature subset. In the proposed method, different solutions have been created in the initial population, but by examining the rate of classification error related to the subset of selected features, the optimal solution can be selected. Solutions created during iterations can solve the intrusion detection problem, but their intrusion detection accuracy will be less than the near-optimal solution. The proposed method has selected the best solution by evaluating the solutions in terms of intrusion detection accuracy.

With regard to the implementation of the proposed method on the initial population, the mentioned solutions were selected as an expert generation, were present in all

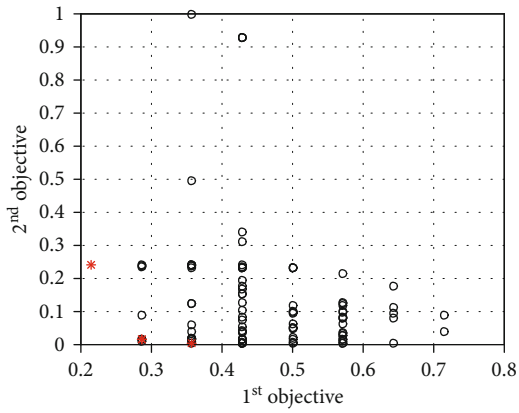

Figure 3: Distribution of expert solutions.

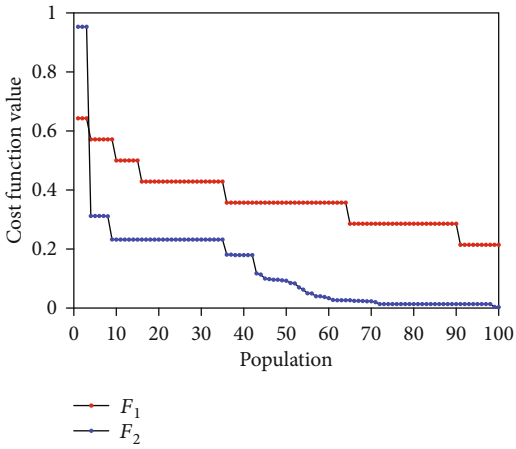

Figure 4: The convergence of target functions' values toward the optimal amount.

iterations, and were improved accordingly. In the last generation of iteration, most particles tended to the expert particles or dominant particles that were repeated in the previous steps and were reserved in the repository. Therefore, the values of the fitness function were close to optimal for all particles in the last generation of iteration. As mentioned before, the value of the fitness function was obtained from a combination of two F_1 and F_2 functions. Accordingly, the final solutions decreased the errors of the proposed method in addition to reducing the number of features in the subset of selected features. Figure 4 shows the convergence of F_1 and F_2 functions toward the optimal point.

As shown in Figure 4, the optimal state of the two objectives had lower values in functions F_1 and F_2 considering that the nature of both target functions was minimized. Therefore, the data presented in Figure 3 revealed that the diagram related to function F_1, which showed the number of selected features, was gradually decreased and reached 0.2 in the end. The diagram showed that an intrusion system could be established with 20% of the total KDD data, and the rest of the data had no use in determining the group related to nodes in the dataset. In addition, with regard to the diagram related to the F_2 function in Figure 4, the errors of

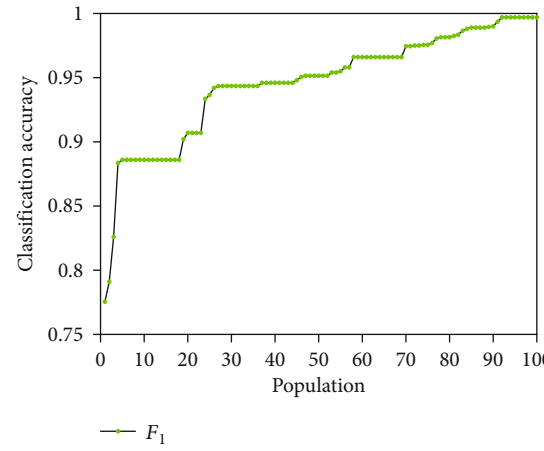

Figure 5: FLN accuracy for intrusion detection patterns.

the proposed method decreased gradually and ultimately reached zero.

A general interpretation of Figure 4 demonstrated that the proposed technique was able to reduce the errors of the intrusion detection system to the lowest level by decreasing the number of features existing in the dataset and selecting the most appropriate features. Given the use of FLN in the method, it could be expressed that the values obtained in F_2 fitness function can be used as an FLN classification error in the proposed technique. As such, the accuracy of the FLN method has shown in Figure 5 for training the intrusion detection techniques in the network.

As observed in Figure 5, the accuracy of the proposed method tended to be optimal with regard to the tendency of the solutions toward the dominant solutions of MOPSO, reaching 99.7%. Therefore, we extracted the most adequate solution obtained from the proposed PSO algorithm, according to which the test dataset was predicted.

4.1. Proposed Model Evaluation. As observed in the previous section, FLN was developed in the present article for classification and prediction of destructive nodes in the network, and the simulation results were determined based on the selection of a feature subset according to MOPSO. There are several criteria for assessing the performance of the proposed method, the most important of which are performance criterion, error rate criterion, and sample the correct prediction rate criterion. The performance criterion monitors the performance of the developed model, meaning that the ideal performance could be drawn by having the label of the data's classes. In addition, evaluation criteria based on the confusion matrix could be used for a two-class problem in order to assess the quality of the proposed method in detecting intrusion and reducing node classification errors in the network. These criteria included accuracy, recall, precision, classification rate (CR), detection rate (DR), false-positive rate (FPR), and F-measure, defined as follows [22]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (5)$$

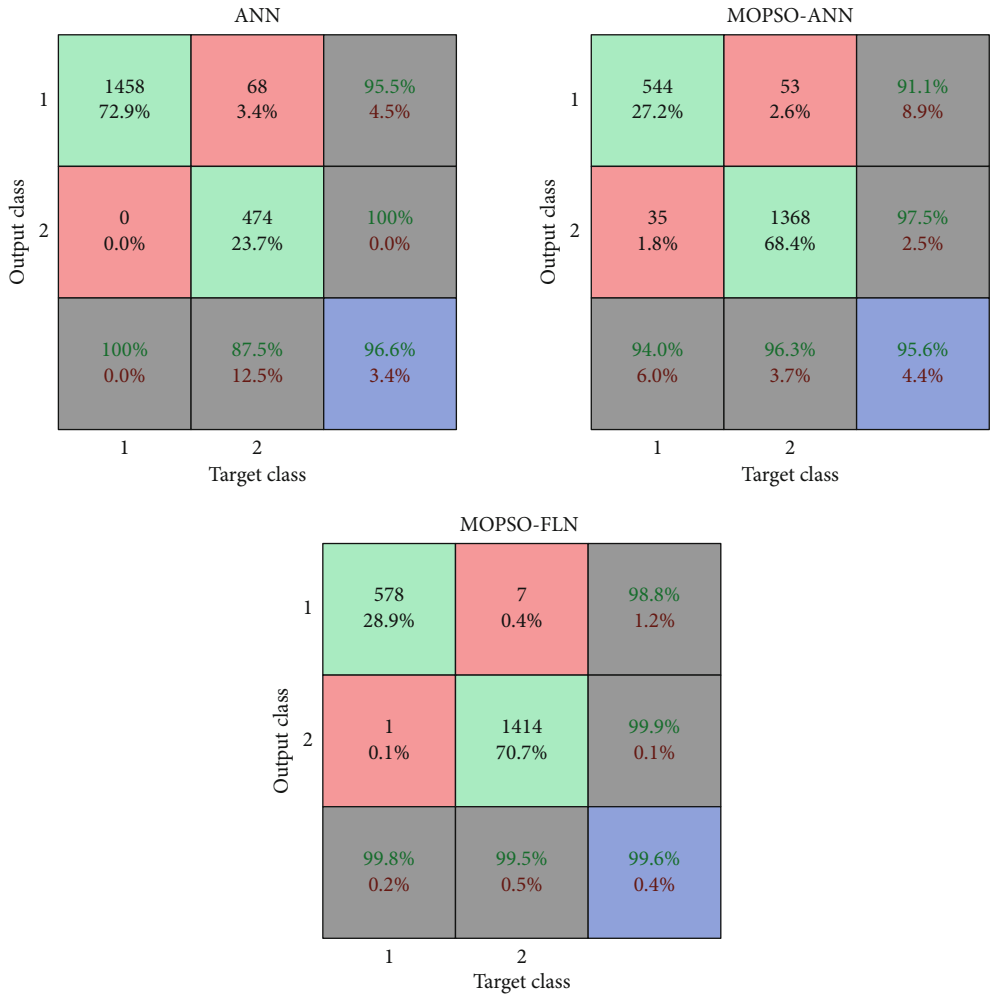

Figure 6: Comparison of the confusion matrix of the proposed method and neural network.

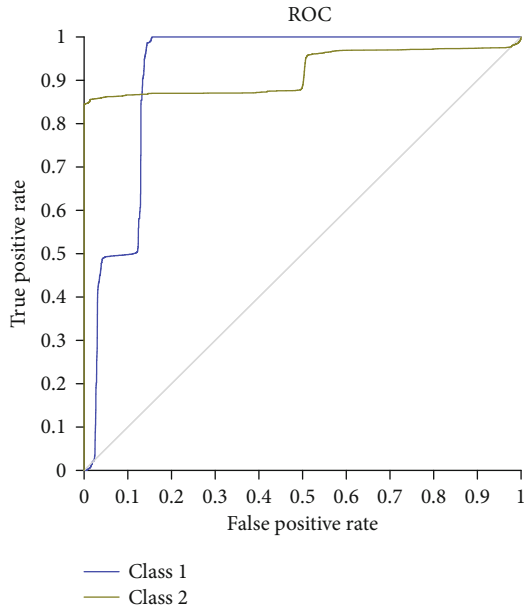

Figure 7: ROC curve of the proposed method.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{7}$$

$$\text{Classification Rat(CR)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{8}$$

$$\text{Detection rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{9}$$

$$\text{False Positive rate (FPR)} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \tag{10}$$

$$F - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{11}$$

The mentioned evaluation criteria are used as a tool to assess the efficiency of the proposed method and compare it with other existing techniques. Therefore, we compare the proposed method with neural networks without using feature subset selection and neural network approach according to MOPSO-based feature selection. Figure 6 shows the

Table 2: Comparison of values related to the evaluation criteria.

Method	CR (accuracy)	False-positive rate (FPR)	Precision	DR (recall)	*F*-measure
MOPSO-FLN	**99.6**	**0.0137**	**99.44**	99.79	**99.61**
MOPSO-ANN	95.6	0.0888	96.27	97.51	96.88
ANN	96.6	0.0446	84.7	**99.99**	91.71

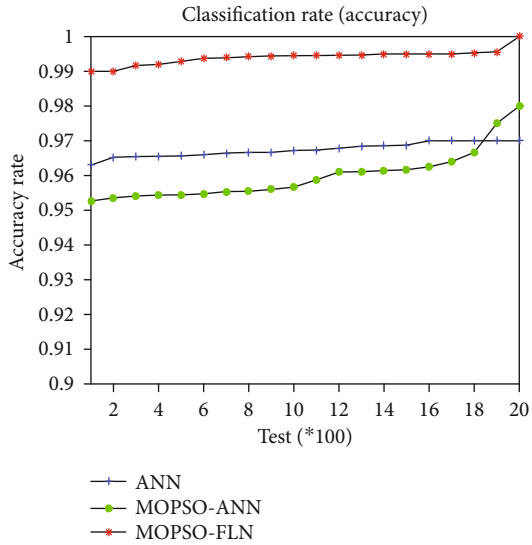

Figure 8: Comparison of the classification rate criterion (accuracy).

comparison of the confusion matrix related to the proposed method and the neural network.

As shown in Figure 6, 99.6% of the total data was classified accurately in the proposed method. Meanwhile, 96.6% and 95.6% of the data were classified correctly in the ANN and MOPSO-ANN methods, respectively.

Another criterion used to evaluate the proposed method is the ROC curve. The ROC curve shows the relation of the true-positive rate to the true-negative rate. The ROC curve of proposed method has shown in Figure 7.

As shown in Figure 7. The proposed method has identified normal and malicious nodes with high accuracy. Table 2 shows a comparison of the values related to the proposed method, ANN, and MOPSO-ANN.

As observed in Table 2, the proposed method had a more adequate performance in terms of the evaluation criteria, compared to the ANN and MOPSO-ANN methods. Figure 8 illustrates the comparison of the classification rate criterion (accuracy) between MOPSO-FLN, ANN, and MOPSO-ANN in 10 steps of 10-fold cross-validation.

According to Figure 8, there were improvements in the proposed method regarding the classification rate (accuracy), compared to ANN and MOPSO-ANN. In the neural network method, the model may experience overfitting given that the training process is carried out completely. In this phenomenon, the model focuses on the training samples and learns all features and relations among the training samples. In addition, its accuracy is maximized in the classification of the training samples. Now, when new test samples that were not previously observed by the model are entered into the system, the model may lack the sufficient accuracy to detect the relations between the features of the new samples that are different from the training samples, which leads to the less efficient performance of the system. Accordingly, the proposed method continues the training steps until reaching the desired accuracy in order to prevent overfitting and increase an IDS's performance. As such, it seems that the FLN method had better accuracy, compared to the method of artificial neural networks. In fact, the proposed method was able to detect a higher percentage of attack and healthy nodes accurately. Figure 8 depicts a diagram that compares the detection rate criteria (recall) between MOPSO-FLN, ANN, and MOPSO-ANN.

The accuracy rate in the proposed method may be in the form of the ratio of the detected healthy nodes among all healthy nodes existing in the dataset that might be detected accurately or be among the false-negative samples. The classification rate is in fact a sum of true-positive samples on all true-positive and false-negative samples. True positive refers to healthy nodes that are detected accurately, whereas false negative refers to healthy nodes that are falsely detected as attack nodes. This relationship shows the proposed model's ability to detect healthy nodes accurately. The higher the value of this relationship, the lower the number of samples related to false negative, where the predicted class has negative nodes, and the actual class has healthy nodes. The lower values of false-negative samples increase the performance of the proposed method in detecting healthy nodes.

According to Figure 9, the proposed method was improved in terms of the detection rate (recall), compared to ANN and MOPSO-ANN. FLN has a higher detection rate, compared to neural networks, considering its lower training than the mentioned technique.

With regard to the structure of accurate learning networks, the training process is discontinued, and the results are transferred to the output when the desired accuracy is reached. Therefore, the features of healthy nodes may not be fully learned but overfitting is avoided in the process, which is an advantage of the proposed method. In fact, the proposed technique has high accuracy for new and unknown samples. In addition to the detection rate in the proposed method, the criterion of the positive error rate of discovered samples is of paramount importance. Figure 10 shows a diagram related to the comparison of the positive error rate criterion between MOPSO-FLN, ANN, and MOPSO-ANN.

As observed in Figure 10, the proposed method had a lower value in terms of the positive error rate, compared to ANN and MOPSO-ANN. In this regard, the positive error rate in the proposed method referred to attacks that could not be detected by IDS. In fact, FLN had a lower positive error rate, compared to the neural networks method,

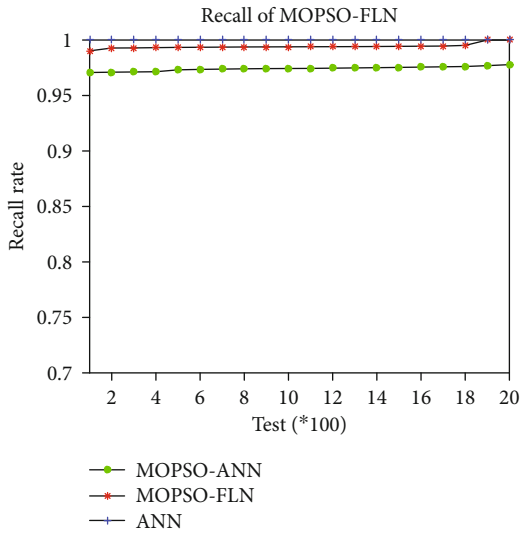

Figure 9: Comparison of detection rate criterion (recall).

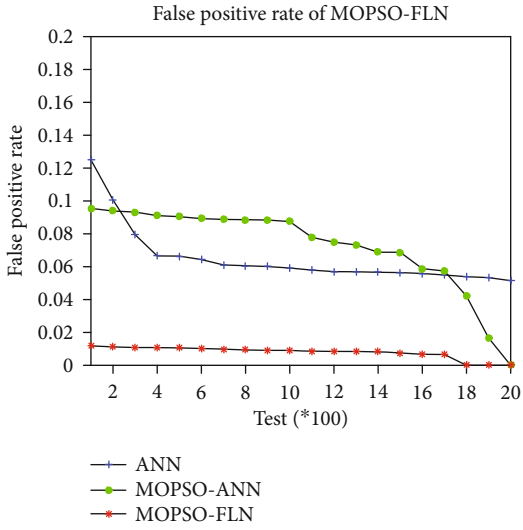

Figure 10: Comparison of the positive error rate criterion.

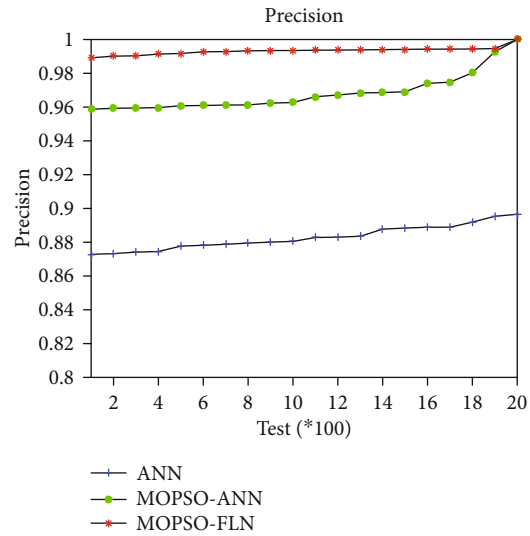

Figure 11: Comparison of the accuracy criterion of the proposed method and neural network.

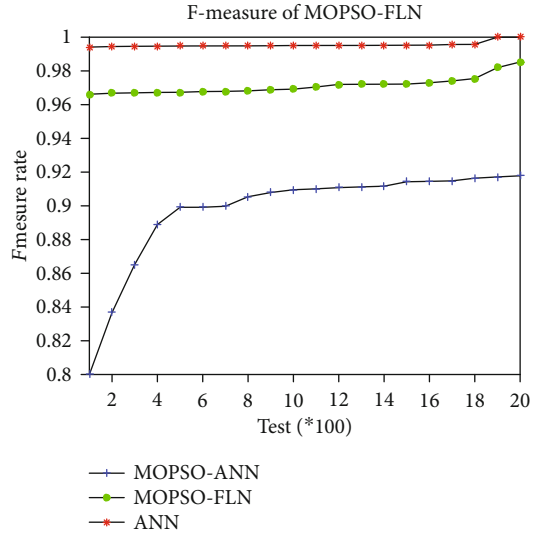

Figure 12: F-measure comparison.

considering its focus on attacks. In spite of full training on attacks existing in the training dataset, neural networks fail to detect some new attacks, about which they had no previous training. Meanwhile, the proposed method was able to detect new attacks by creating a balance between learning training samples and the network's speed. After a positive error rate, we evaluated the attack detection accuracy of the proposed method. In IDSs, accuracy is the form of the ratio of true-positive samples to true-positive samples and true-negative samples, which estimates a reflection of attack detection ability in the classification methods. The higher this value, the higher the classification method's ability to detect and identify new attacks. Figure 11 illustrates a diagram related to the comparison of the positive accuracy criterion between MOPSO-FLN, ANN, and MOPSO-ANN.

According to Figure 11, the proposed method was improved in terms of the accuracy criterion, compared to ANN and MOPSO-ANN. The accuracy of the proposed method improved considering the focus of FLN on attacks and its ability to considerably detect new attacks, compared to neural networks. Figure 11 is a complete representation of the presence of overfitting in neural networks and the lack of presence of this phenomenon in the proposed method. In general, overfitting decreases a model's accuracy per new samples. In fact, FLN can detect most attacks that are among new samples and have not been previously observed in the model. The final criterion assessed in the present study was F-measure, which was a combination of two accuracy and detection rate criteria. The criterion was recognized as a general criterion of the performance of classification methods and IDSs. The higher the value of the criterion, the higher the IDS's ability to classify healthy samples and predict attacks in the training dataset and new attacks that enter

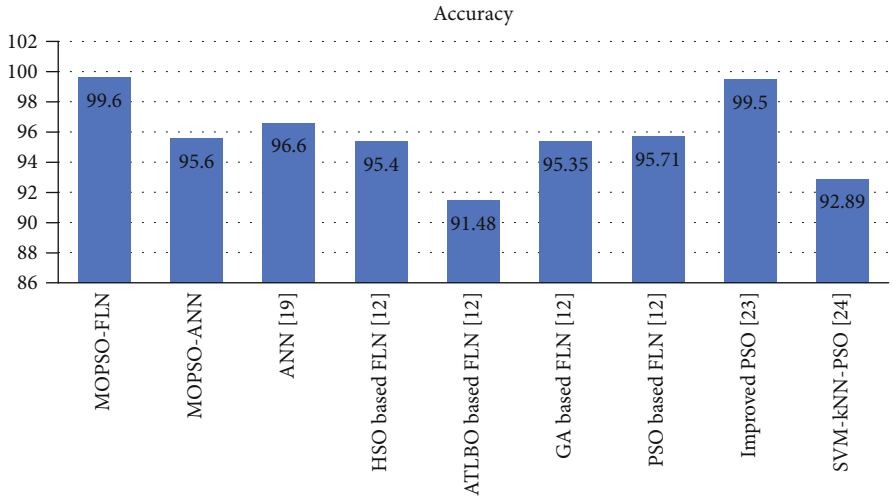

Figure 13: Comparison of the proposed method with previous techniques. All methods compared to the proposed method used a different feature subset selection approach. It can be seen that the accuracy of the proposed method is higher than other methods due to the selection of the optimal feature subset. According to Figure 13, MOPSO-FLN had higher intrusion detection and prediction accuracy, compared to previous methods.

the system. Figure 12 shows a diagram related to the comparison of MOPSO-FLN, ANN, and MOPSO regarding F-measure.

According to Figure 12, the proposed method improved in terms of F-measure criterion, compared to ANN and MOPSO-ANN. In other words, FLN had a better performance in detecting healthy nodes and new attacks in the network when combined with MOPSO-based feature subset selection, compared to neural networks.

4.2. Comparison of Proposed Method with Previous Techniques. In this subset, we compared MOPSO-FLN with other methods existing in publications [12–24] to assess the validity of the proposed method in terms of the prediction accuracy criterion. Figure 13 shows a comparison of the proposed method with previous techniques.

5. Discussion and Conclusion

With regard to the use of the MOPSO-based feature selection approach, it was aimed at reducing performance errors in the classification model and prediction of test samples in addition to finding the best features representing all the features in the dataset. Optimal features extracted in the proposed method by MOPSO were evaluated in each stage of optimization algorithm iteration in order to increase the speed of particles' movement toward particles with high values at Pareto front and reduce data classification errors based on these features in each step. Therefore, selecting these features least to simple distinguishing of the samples related to classes by the classification model and high classification accuracy.

Furthermore, the proposed method was compared to other popular approaches presented in articles. According to the results, the neural network had a relatively lower accuracy, compared to the proposed method, if used independently and without selecting important features subset from the dataset related to intrusion in wireless networks. The difference of about 4% in the accuracy of the proposed method and the neural network-based method was another evidence of the importance of selecting a subset of features using FLN.

The proposed method was also compared to several other approaches that use metaheuristic optimization algorithm-based feature subset selection. According to the results, the proposed method had higher test sample prediction accuracy, compared to the intrusion detection approach, which was a combination of MOPSO-based feature selection and fast neural network. Accordingly, it could be concluded that the proposed method was able to extract important features by using MOPSO-based feature subset selection. In addition, the model yielded acceptable results in terms of the detection and prediction of intrusion in wireless networks by using an evaluation function, which was a combination of a number of features and classification error.

Data Availability

This research and proposed methodology was simulated, and all data are included already in the paper. So, there is no need for extra data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. K. Saxena, S. Sinha, and P. Shukla, "A review on intrusion detection system in mobile ad-hoc network," in *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, pp. 549–554, Bhopal, India, 2018.

[2] S. Muruganandam, J. A. Renjit, and R. S. Kumar, "A survey: comparative study of security methods and trust manage solutions in MANET," in *2019 Fifth International Conference on*

Science Technology Engineering and Mathematics (ICONSTEM), pp. 125–131, Chennai, India, 2019.

[3] V. Singh, D. A. Singh, and M. M. Hassan, "Survey: black hole attack detection in MANET," in *Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE)*, Sultanpur, UP, India, 2019.

[4] A. Gupta and A. Dubey, "A survey on various applications and blackhole attack in mobile ad hoc network," *Recent Trends in Parallel Computing*, vol. 5, pp. 1–6, 2018.

[5] R. Fotohi and S. Jamali, "A comprehensive study on defence against wormhole attack methods in mobile ad hoc networks," *International journal of Computer Science & Network Solutions*, vol. 2, pp. 37–56, 2014.

[6] G. Kumar Ahuja and G. Kumar, "Evaluation metrics for intrusion detection systems-a study," *Evaluation*, vol. 2, pp. 11–17, 2014.

[7] P. Yang, Z. Li, P. Yang, and Y. Dong, "Information-centric mobile ad hoc networks and content routing: a survey," *Ad Hoc Networks*, vol. 58, pp. 255–268, 2017.

[8] A. Sultana and M. A. Jabbar, "Intelligent network intrusion detection system using data mining techniques," in *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pp. 329–333, Bangalore, India, 2017.

[9] S. Sindhuja and R. Vadivel, "A study on intrusion detection system of mobile ad-hoc networks," in *Soft Computing for Problem Solving*, pp. 307–316, Springer, 2020.

[10] E. Rosas, N. Hidalgo, V. Gil-Costa et al., "Survey on simulation for mobile ad-hoc communication for disaster scenarios," *Journal of Computer Science and Technology*, vol. 31, no. 2, pp. 326–349, 2016.

[11] D. Rajalakshmi and K. Meena, "A survey of intrusion detection with higher malicious misbehavior detection in MANET," *International journal of civil engineering and technology*, vol. 8, no. 10, pp. 99–110, 2017.

[12] M. H. Ali, B. A. D. Al Mohammed, A. Ismail, and M. F. Zolkipli, "A new intrusion detection system based on fast learning network and particle swarm optimization," *IEEE Access*, vol. 6, pp. 20255–20261, 2018.

[13] B. Selvakumar and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," *Computers & Security*, vol. 81, pp. 148–155, 2019.

[14] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, "A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection," *Computers & Security*, vol. 75, pp. 36–58, 2018.

[15] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," *Expert systems with applications*, vol. 148, p. 113249, 2020.

[16] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 174, 2020.

[17] B. Senthilnayaki, K. Venkatalakshmi, and A. Kannan, "Intrusion detection system using fuzzy rough set feature selection and modified KNN classifier," *The International Arab Journal of Information Technology*, vol. 16, no. 4, pp. 746–753, 2019.

[18] M. Habib, I. Aljarah, H. Faris, and S. Mirjalili, "Multi-objective particle swarm optimization: theory, literature review, and application in feature selection for medical diagnosis," in *Evolutionary Machine Learning Techniques*, pp. 175–201, Springer, Singapore, 2020.

[19] C. A. Coello Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, vol. 2, pp. 1051–1056, Honolulu, HI, USA, 2002.

[20] C. H. Dagli, "Complexity analysis of multilayer perceptron neural network embedded into a wireless sensor network," *Procedia Computer Science*, vol. 36, pp. 192–197, 2014.

[21] KDD Cup, *Computer Network Intrusion Detection*, 1999, https://www.kdd.org/kdd-cup/view/kdd-cup-1999.

[22] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," in *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 277–282, Subotica, Serbia, 2017.

[23] A. Dickson and C. Thomas, "Improved PSO for optimizing the performance of intrusion detection systems," *Journal of Intelligent Fuzzy Systems*, vol. 38, pp. 6537–6547, 2020.

[24] A. Aburomman and R. Mamun, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Applied Soft Computing*, vol. 38, pp. 360–372, 2016.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6627639, 10 pages
https://doi.org/10.1155/2021/6627639

Research Article

A New Approach Customizable Distributed Network Service Discovery System

Xiangzhan Yu, Zhichao Hu, and Yi Xin

School of Cyberspace Science, Harbin Institute of Technology, 150001, China

Correspondence should be addressed to Xiangzhan Yu; yxz@hit.edu.cn

Received 31 December 2020; Revised 15 March 2021; Accepted 23 April 2021; Published 12 May 2021

Academic Editor: Lihua Yin

Computer systems and applications on the internet provide services to outsiders and, at the same time, the vulnerabilities may be exploited by attackers and leak some sensitive private information. To collect and monitor the service information provided by the network environment such as IoT (Internet of Things), vehicular networks, cloud computing, and cloud storage, it is particularly important that a system can provide faster service discovery for discovering and identifying specific network services. The current service discovery systems mainly use port scanning technology, including Nmap, Zmap, and Masscan. However, these technologies hard code the service features and only support common services so that cannot cope with real-time updates and changing network services. To solve the above problems, this paper proposed a customizable distributed network service discovery system based on stateless scanning technology of Masscan and proposed a customizable interactive pattern set syntax. The system used random destination address technologies to scan for Ipv4 address allocation and used a distributed deployment scheme. Experimental results show that the system has high scanning speed and has high adaptability to new services and special services.

1. Introduction

With the growth of internet devices and applications, various large scale cyberattacks continue to emerge, and internet vulnerabilities also show a surging trend [1, 2]. Despite the recent growth in computer networking best practices, the continual improvement in Internet-based services has presented new challenges in maintaining security and preserving privacy [3, 4]. Even though some enterprises have discovered vulnerabilities and released repair patches, many users still do not update, leading to potential security threats and providing attackers with access to attacks. At the same time, many web apps and services are installed on the devices hosting a web client and providing the interface for user control with open ports, where security and privacy are the critical issues [5, 6]. Censorship needs to know these security risks, that is, to count and supervise the service information in a large-scale network.

The IoT, vehicular networks, cloud computing, cloud storage, and other environments can provide users with flexible and convenient service access [7, 8]. While greatly improving the convenience of life, privacy issues caused by security problems are also becoming more and more serious [9, 10]. For example, in vehicular networks, security plays a dominant role as applications based on vehicular networks usually correspond to passengers' safety (e.g., self-driving) and privacy information (e.g., driving history) [11]. So the security of the network should be one of the most important issues in the upcoming days. Searching and gathering the specific information of the devices on the internet provide data to analyze the vulnerabilities which can enhance system's security and preserve privacy [4, 12]. A common tool to deal with this problem is port scanning, but current scanning tools have two disadvantages. In one hand, supported services are mostly hard coded in the system, and for less common, newer services, you need to wait for the developer's update support. It has poor scalability, as evidenced by the famous Masscan, which only supports HTTP, SSL, and other common protocols but ignores industrial network protocols and instant messaging protocols. On the other hand, the traditional scanning methods are noninteractive detection, so they are failed for service identification with multiple interactions.

In order to solve the above problems, we designed a customizable distributed network service discovery system (CDNSDS) in this paper. The main contributions of this work are as follows:

(i) We designed a system architecture, which includes three subsystems: central control subsystem (CCS), schedule control subsystem (SCS), and scanning proxy subsystem (SPS). The CCS is the brain; it receives the user's instructions and manages and assigns tasks to the SCS. The SCS is the bridge connecting CCS and multiple SPS. The last subsystem is the SPS, the key factor for performance. We optimized Masscan, the most efficient scanning tool currently, and used a distributed program to improve concurrency

(ii) To be customizable, we had compiled a pattern set of syntax conventions. The syntax conventions can convert the user's customized services description, including interactive service, to standard syntax which is accepted by a scanning tool in the SPS

The rest of the paper is organized as follows: a related work is described in Sections 2 and 3 elaborates the proposed system CDNSDS; experimental results are followed in Section 4. Finally, we summarize the research in Section 5 with a discussion as well as a future work.

2. Related Work

In the study of empirical security, fast Internet-scale network service discovery has opened a new avenue, while scanning technology plays an important role. One of the earlier scanning tools is Nmap [13], which maintains a full-connected state to track hosts that have been scanned and to handle timeouts and retransmissions. In this state, the unresponsive requests cost too much time; it takes several weeks or many machines for Nmap to scan the public address space. To overcome the issue of efficiency, Zakir et al. [14] designed a scanning tool Zmap based on no per-connected state. For Zmap, there is no need to track connection timeouts, and it accepts response packets with the correct state fields during the scanning. The manner of Zmap is similar to SYN cookies. Compared to Nmap, with the same accuracy, Zmap is capable of scanning the IPv4 public address space for under 45 minutes on a single machine [15], which is over 1300 times faster than the most aggressive Nmap default settings. Further, drawing on the data collected by Zmap from ongoing Internet-wide scanning, Zakir et al. [16] designed a public search engine named Censys, which supports full-text searches on protocol banners and querying a wide range of derived fields. With Censys, it becomes simple to help researchers answer security-related questions.

Although Zmap has greatly improved in performance, scanning technology is still in progress. The fastest internet port scanner Masscan [17], an open source project, only takes six minutes to scan the IPv4 public address space, transmitting 10 million packets per second. For the sake of high performance, Masscan takes endeavour from three aspects. For one thing, similar to Zmap is the use of no per-connected state. Because Masscan can simultaneously maintain the number of connections which is set by the program itself, the number can be set very large, so the scanning speed is much faster than other scanners. For another, Masscan uses a custom TCP/IP stack, and a designated network device and PF_RING DNA driver are necessary conditions. It is a lightweight protocol stack that means the underlying packet processing, connection control, etc. will bypass the operating system protocol stack, so the protocol stack process is simpler and there will be a substantial increase in performance. In addition, the configuration of Masscan is more flexible, not limited to single-port probing, and a user can specify the port segment. Through a target address randomization algorithm, it can be more effective to random host range for target that can evade from detecting of Intrusion Detection System (IDS).

Except for the above famous scanners, a number of research efforts focus on empirical security. In order to scan anonymously, Rodney et al. [18] performed scanning through Tor, which can hide the source's IP address from the target. Andrei et al. [19] proposed a public, large-scale analysis of firmware images, which supported a global understanding of embedded systems' security. At the same time, the Heartbleed vulnerability is the measurement and analysis in [20]. In the weak keys detecting, researchers [21–23] reported they had computed the RSA private keys for HTTPS hosts on the internet and traced the underlying issue to widespread random number generation failures on network devices. Arzhakov et al. [24] proposed a multithread network scanner with a very flexible architecture that allows us to parallelize the process of sending requests and receiving responses from remote hosts. Focused on automated web scanners, Fang et al. [25] gave a new direction for the detection of the fingerprint using a finite state machine to abstract differences of scanners.

3. Service Discovery System

3.1. System Architecture. The traditional service discovery systems may cause issues such as triggered IDS alarm and single-node detection poor performance. In this paper, we design and implement the CDNSDS and the architecture is shown in Figure 1.

CDNSDS includes three subsystems.

(i) Central control subsystem (CCS): it is the brain, which receives a user's instructions and manages and assigns tasks to the SCS. Users can get the task process and results and manage the attribute and state of the scanning node. In this subsystem, we design a pattern set of syntax conventions to support customizability.

(ii) Schedule control subsystem (SCS): it is a bridge connecting CCS and multiple SPS. It provides task division, scheduling management, and results of the temporary service.

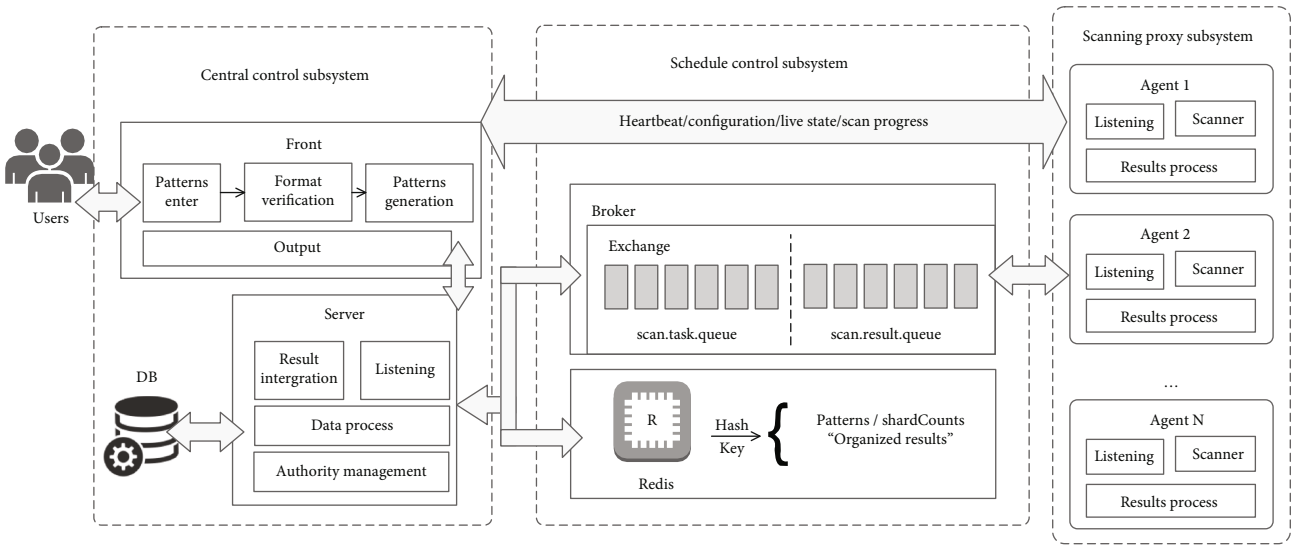

Figure 1: System architecture.

Table 1: The property of pattern set.

Name	State	Description	Format	Essential
msg	s	After the state is transferred to this node, the text in the msg attribute needs to be filled into the application layer load; the data packet is constructed and probed	Hex/string	Y
waiting	s	After sending the probe packet in this state, state will be transferred to receive state when receives a response packet	"r" + index	Y
isbanner	r	Output application layer load to output file	True/false	N
patterns	r	Probe pattern set, which is used to guide state transitions	Hex/string	N
len	r	Limit of payload length	Range	N
goto	r	The state is after filtering of patterns and len	{index, state_with_id}	Y

(iii) Scanning proxy subsystem (SPS): it is the key factor for performance. The SPS consists of several distributed agent modules. Each agent is a scan node with optimized Masscan that performs real-time scan task from the SCS.

3.2. Central Control Subsystem (CCS). The CCS provides users with customizable service probe interfaces. A pattern set of syntax conventions is defined for customization as follows.

We use s for send state and r for receive state. Denote $D = \{s_i, r_j\}$ as instructions, s_i is the ith state of send, and r_j is the jth state of receive. The attributes of different state are split with character '.'. Property set of send state s is

$$P_s = \{\text{msg}, \text{waiting}\}, \tag{1}$$

and property set of receive state r is

$$P_r = \{\text{isbanner}, \text{patterns}, \text{len}, \text{goto}\}. \tag{2}$$

Table 1 shows the list of the property descriptions of P_s and P_r. An example state transition diagram is shown in Figure 2.

In this example, there are three kinds of state node: (1) the green solid nodes $s0$, $s1$, and $s2$ are send state; (2) the hollow blue nodes $r1$, $r2$, and $r3$ are receive state without banner output that means the property isbanner equals to false; and (3) the solid nodes $r0$, $r4$, and $r5$ are receive state with banner output that means the property isbanner equals to true.

3.3. Schedule Control Subsystem (SCS). The SCS is aimed building an efficient and reliable communication environment between the CCS and SPS, while providing intermediate data storage and high-speed read service.

The SCS contains three modules: state management module, message queue module, and cache module.

(i) Status management module: to better understand the survival status and scanning progress of each scanning agent node, the SCS is logically responsible for building the communication environment between the central control system and each scanning node. At the same time, to deal with the problems such as downtime of the CCS and change of server address, the SCS also provides an interface to dynamically manage the connection configuration of the scanning nodes to ensure the normal

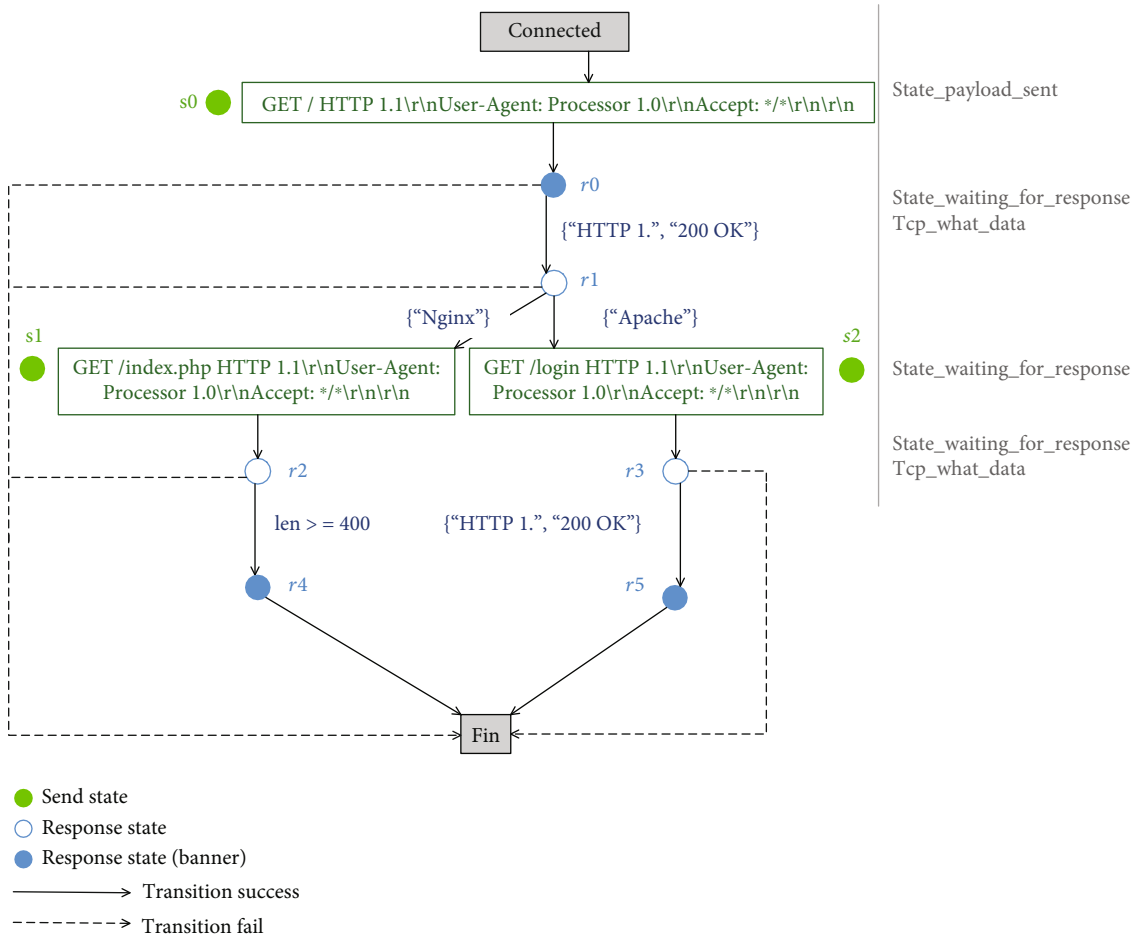

Figure 2: State transition diagram.

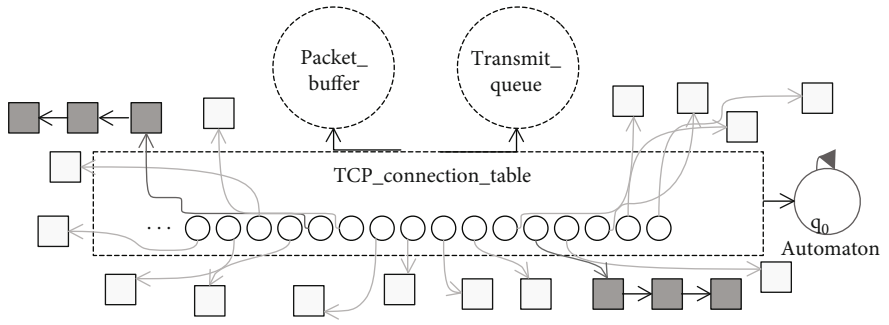

Figure 3: TCP connection table in user-mode protocol stack.

delivery of heartbeat packets and scanning progress packets.

(ii) Message queue module: it contains task queue management and result queue management. The detection tasks issued by the CCS will split into specified slices for smaller granularity and detection. Each scanning agent node consumes only one slice at a time, and these task slices will be handed over to the task queue management. From the scanning agent's point of view, each slice represents a scanning task, and the result of the task may be success, fail, or timeout. The task queue manages the various results that may exist after each task slice is received. The slice that fails to scan is reenlisted for other scanning nodes to probe again. When the task is successfully scanned, the result data will be passed back from the scanning node to the result queue, which will store that result slice temporarily for the CCS.

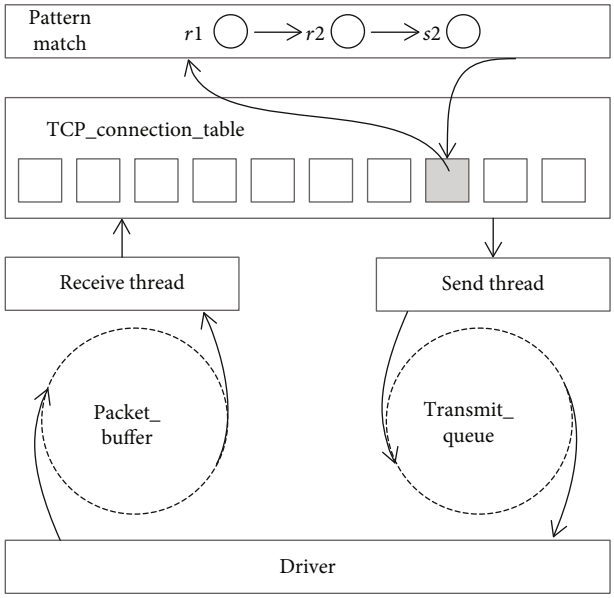

Figure 4: Interactive service detection hierarchy.

1. c ⟵ $fe[r, a, b]_K(m)$
2. If c ∈ M, Then
3. Return
4. Else
5. Return $Fe[r, a, b]_K(c)$
6. EndIf

Algorithm 1: $Fe[r, a, b]_K(m.)$

Similarly, the result queue will also manage the status of the results processed by the central control system as described above.

(iii) Cache module: after the scanning node successfully detects the target address set, it will pass the result slice back to the result queue and then open the next detection task. Due to the large number of potential detection nodes, if each slice's results are stacked in the CCS, it will increase the pressure on its storage and processing. Therefore, the module provides a dumping service to the cache and notifies the CCS to consolidate, deduplicate, and persist the results after receiving. Thus, the cache module provides memory-level high-speed data processing functions.

For each packet received, the SCS will determine whether it is a task slice, heartbeat data, or task result.

(i) Task slice packet: it is handed over to the "task queue" to manage and monitor the execution (dispatch) result of this slice.

(ii) Heartbeat packet: it is forwarded to the central control system to update the survival and progress status.

1. $L \longleftarrow m \bmod a$; $R \longleftarrow \lfloor m/a \rfloor$
2. For $j \longleftarrow i$ to r, do
3. If j is *odd*, Then
4. $tmp \longleftarrow (L + F_j(R)) \bmod a$
5. Else
6. $tmp \longleftarrow (L + F_j(R)) \bmod b$
7. EndIf
8. L ⟵ R ; $R \longleftarrow tmp$;
9. EndFor
10. If r is *odd*, then
11. Return $aR + L$
12. Else
13. Return $bR + L$
14. EndIf

Algorithm 2: $fe[r, a, b]_K(m)$

1. c ⟵ $fe[r, a, b]_K^{-1}(m)$
2. If c ∈ M, Then
3. return
4. Else
5. return $Fe[r, a, b]_K^{-1}(c)$
6. EndIf

Algorithm 3: $Fe[r, a, b]_K^{-1}(m)$

1. If (r is odd) Then
2. $R \longleftarrow mod\ a$; $L \longleftarrow \lfloor m/a \rfloor$
3. Else
4. $L \longleftarrow m \bmod a$; $R \longleftarrow \lfloor m/a \rfloor$
5. End
6. For $j \longleftarrow r$ to 1, do
7. If j is *odd*, Then
8. $tmp \longleftarrow (R - F_j(L)) \bmod a$
9. Else
10. $tmp \longleftarrow (R + F_j(L)) \bmod b$
11. EndIf
12. $R \longleftarrow L$; $L \longleftarrow tmp$;
13. EndFor
14. Return $aR + L$

Algorithm 4: $fe[r, a, b]_K^{-1}(m)$

(iii) Task slice result data: it is temporarily stored in the cache module. The task slice result data is temporarily stored in the cache module to remind the central control system for integration.

3.4. Scanning Proxy Subsystem (SPS)

3.4.1. Interactive Service Detection. When the SPS performs a scan task, there are four situations after sending the first TCP handshake request.

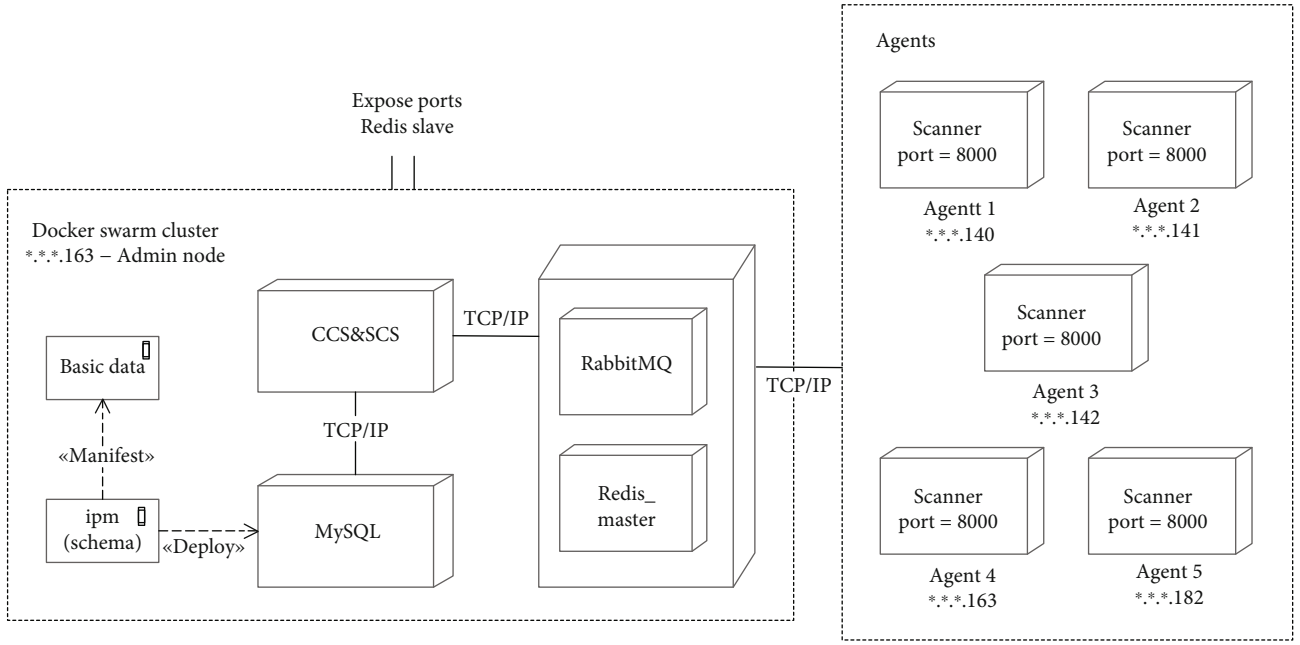

Figure 5: System deployment diagram.

(i) SYN-ACK: the host port of corresponding target is open, and we can continue to probe service.

(ii) RST: the target is open, but the destination port is close.

(iii) ICMP unreachable: the target is close.

(iv) No response: connection timed out.

It is obviously that in the first case we can keep detecting while other cases can be directly abandoned.

Normally, scanners based on semiconnected state will send RST to close connection after receiving SYN-ACK. Such a scheme is not suitable for interactive detection. This issue can be resolved by two possible solutions.

(i) Using the operating system protocol stack, reestablish the connection to the open target port for deep probing

(ii) Send ACK to finish three-way handshake instead of RST

The first solution theoretically provides reliable connection, but the number of connections is limited. The second solution requires a user-mode protocol stack and is more efficient than the former. Fortunately, Masscan already provides this functionality. Therefore, the second solution is adopted in this paper.

In order to record all active connections, a TCP connection table is needed to maintain the management of Transmission Control Block (TCB) which contains all the important information about the connection, as shown Figure 3.

The interactive service detection hierarchy is displayed in Figure 4. Through asynchronous threads, the sending and receiving are separated.

During service discovery, packets need to bypass the original system stack; otherwise, the original system stack will send RST packet because of the absence of connections. This paper proposed two solutions.

(i) ARP cheat: send an ARP packet with an unreal IP in same subnet to router.

(ii) Modify Linux iptables: drop traffic with a specified port.

3.4.2. Randomize Target Address. The CCS delivered tasks in the form of fragments. Under the premise of address randomization, in order to avoid duplication of detection intervals of all nodes, the system sets the range as $S_{\text{range}} = N_{\text{hosts}} \times N_{\text{ports}}$ to serialize the scan range.

Denote IP segment $A = \{A_1, A_2 \cdots A_n\}$, port setment $B = \{B_1, B_2 \cdots B_m\}$, IP-port consist data:

$$(A_1B_1), (A_1B_2), \cdots, (A_1B_m), (A_2B_1), \cdots, (A_nB_1), \cdots, (A_nB_m). \tag{3}$$

Scan range mapping set is follows:

$$R_1 = (A_1B_1), R_2 = (A_1B_2), \cdots, R_{n\times m} = (A_nB_m). \tag{4}$$

Using the above mapping set, the conversion from the index of the range to host addresses and ports can be achieved according to equations (5) and (6). We use this conversion to find the IP and port of the ith scan task.

$$\text{ip}_i = \text{pick}\left(\text{addresses}, \frac{i}{\text{port}_{\text{count}}}\right), \tag{5}$$

Table 2: The results of interactive service detection.

Speed (pkt/s, k: 10^3, w: 10^4)	Service	Port	Service time (s)	Port time (s)
0.1k	3078	3167	852	713
0.5k	3190	3154	178	150
1k	3239	3240	95	79
2k	3151	3157	52	46
3k	3033	3080	38	34
5k	2189	2251	26	25
1w	1218	1054	17	17

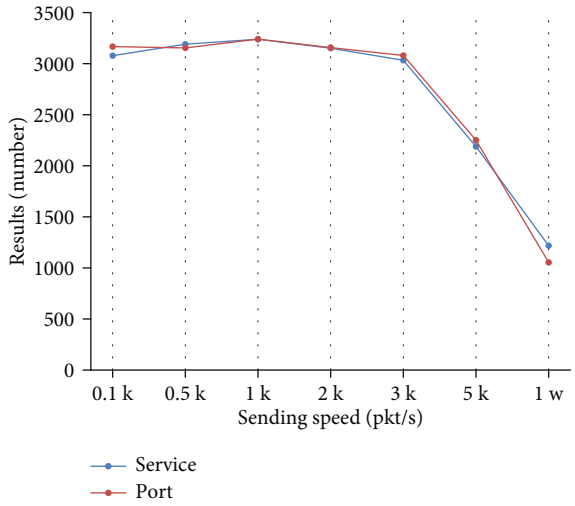

Figure 6: Interactive service detection results chart (k: 10^3, w: 10^4).

$$\text{port}_i = (\text{ports}, i\%\text{port}_{\text{count}}). \quad (6)$$

After serialization, let us suppose fragment range set is r, then randomization of r is that

$$\exists r' \in R, \quad \text{satisfying } r \neq r', \text{card}(A) = \text{card}(B). \quad (7)$$

Due to the condition r' is not unique, the degree of randomization is judged by comparing the same number of elements. The less the number, the higher the degree. In the scan module, randomize target address using generalized Feistel [26] encryption to achieve $k \times M \to M$, where k is any number and M is the target host range. In the k intervention, a mapping process to achieve the same range of random is as follows:

This method is a modification of Feistel encryption, function is Fe$[r, a, b]$, r is rounds, $a, b \in N$, and $ab \geq k$.

The process of randomize address recovery is as follows.

4. Results and Discussion

4.1. Experimental Environment and Deployment. In order to satisfy cluster operations and distributed schedule, the system adopted Docker Swarmkit and deployed in 5 nodes. Among them, in Docker Swarm Mode, the CCS and SCS

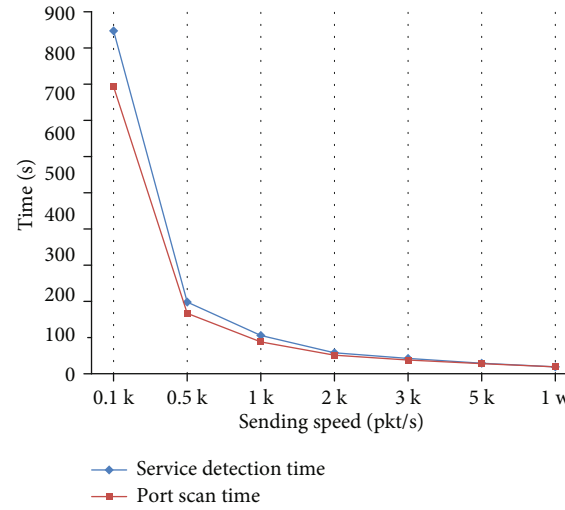

Figure 7: Interactive service detection time chart (k: 10^3, w: 10^4).

Table 3: The results of testing in multimode.

Speed (k: 10^3, w: 10^4)	Multinode: results/time (s)/target	Single-node: results/time (s)/target
0.5k	25641/1743/3	25691/8630/3
0.7k	25764/1248/3	25669/6174/3
1k	25666/877/3	25576/4334/3
2k	25709/448/3	25496/2174/3
4k	25510/230/3	25389/1094/3
6k	25246/157/3	25373/734/3
8k	25377/122/3	25520/557/3
1w	25459/101/3	25527/447/2
2w	25046/57/3	25026/230/2
3w	24968/43/3	25212/160/2
4w	24844/36/3	25137/124/3
5w	23063/31/3	25118/103/2
6w	21221/28/3	25132/88/3
7w	19018/26/2	24998/74/2
8w	16642/25/3	25277/69/3
9w	15190/24/2	25169/64/3
10w	13740/25/3	25203/59/3
20w	10590/25/1	25317/47/2

are deployed in the admin node, and another five SPS are deployed in other nodes. The system deployment diagram is shown as Figure 5.

4.2. System Testing in Single Node. In this paper, we choose a single node to probe HTTP service, the target host segment is 169.54.23.0/16, and the probe port is 80. Each set of experiments is the average of three testing. The testing results are shown as Table 2.

According to the data, we can get the following charts. As Figures 6 and 7 show, when sending rate less than 3000 pkt/s,

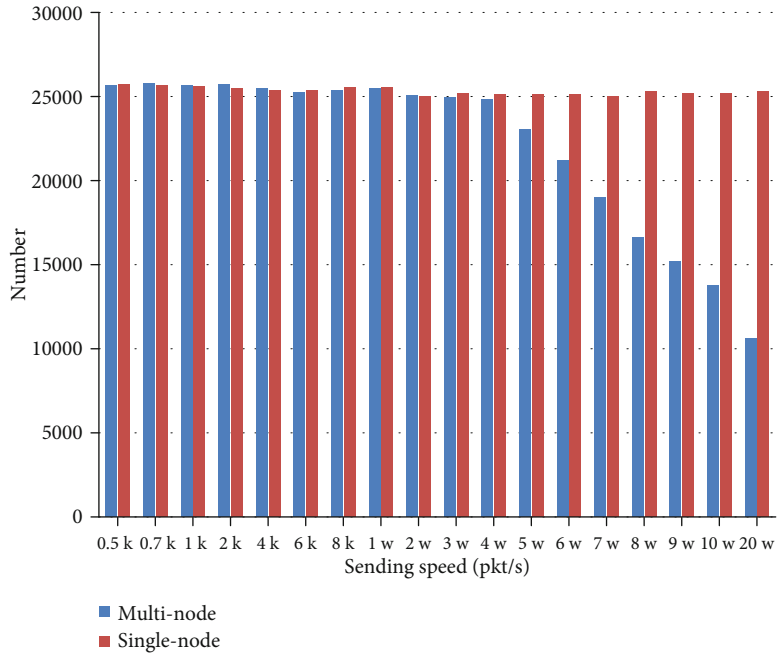

Figure 8: Fixed scanning range (k: 10^3, w: 10^4).

Table 4: The impact of scope on the result.

Scope	Multinode: results/time (s)	Single-node: results/time (s)
/24	18/13	15/13
/22	89/14	89/14
/20	199/14	200/17
/18	842/16	854/34
/16	2862/29	2854/96
/14	8271/74	8247/323
/12	13226/237	13405/1129
/10	25961/878	26281/4341
/8	40920/3427	40632/17102
/6	343482/13881	338102/69277

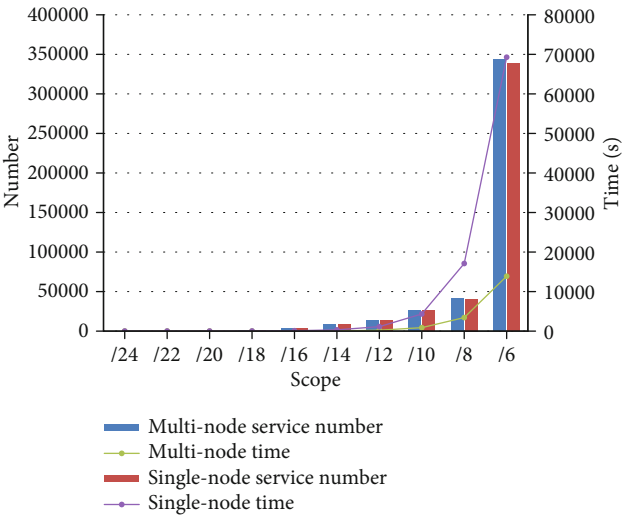

Figure 9: Fixed sending rate.

the results are generally flat, but after that, they decreased significantly. The reason is that when the rate of sending packets increases, the time reduces, so it takes time to wait for service probe packets or SYN-ACKs. Therefore, there is such a situation that the response packets arrive after the scanner shut down. 3000 pkt/s is a stable sending rate in this testing.

4.3. System Testing in Multinode. There are a total of five nodes for testing; the pattern set comes from the analysis of the protocol icoco with port 80. In order to compare the distribute platform and single node, we test in the following two aspects.

(i) Scanning range is fixed, and sending rate changes

(ii) Sending rate is fixed, and scanning range changes

For the first aspect, the target host segment is 169.54.23.0/10 and port is 80. The results are shown as Table 3, and the trend is shown in Figure 8. In Table 3, results mean the number of icoco service.

For the second aspect, the fixed sending rate is set 1000 pkt/s, the results for different scope are shown in Table 4, and the trend is shown in Figure 9.

As can be seen from Figure 9, the accuracy between multinode and single-node is similar. However, as the scanning range increases, the multinode shows better performance.

Consider ratio changes at the same sending rate of the single-node and multinode, as shown in Figure 10. In an ideal

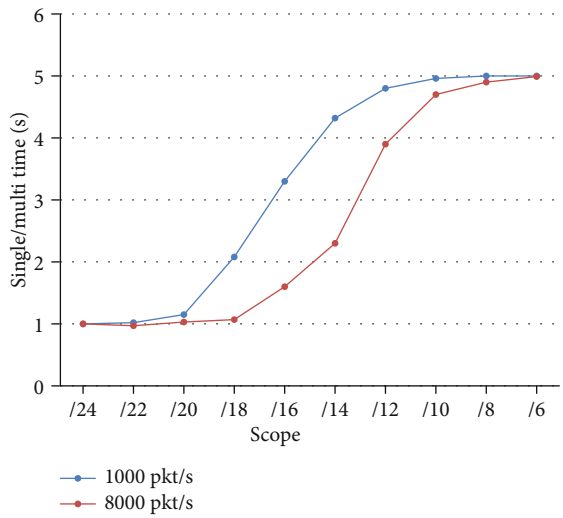

Figure 10: Single-node and multinode ratio diagram.

environment, the detection time of N nodes is $1/N$ of the single node.

As can be seen from Figure 10, reaching the ideal ratio value "5" is determined by the packet rate and the scope of the probe host. It can be derived as follows in conclusion:

(i) If the detection range is fixed, the lower the sending rate, the easier it is to approach the ideal ratio

(ii) If the sending rate is fixed, the larger the detection range, the easier it is to approach the ideal ratio

Based on this conclusion, for better detection results, the system parameters can be adjusted by three factors: the number of nodes, the range of detected host, and network bandwidth.

5. Conclusions

The current service discovery system cannot deal with real-time updates and changing network services. Existing scanners only support the probing of common public protocols. This paper designed a Customizable Distributed Network Service Discovery System (CDNSDS) to solve the issue. CDNSDS consists of three subsystems: CCS, SCS, and SPS. In the CCS, a pattern set of syntax conventions is defined to assist users in customizing scan features. At the same time, the SPS provides an efficient Masscan-based scanning module. In the SPS, we describe interactive detection technology, including TCP connection management, and randomize target address in detail. Finally, the Docker Swarm Mode is used to distribute container choreography, and the experiment shows that the CDNSDS has high efficiency and accuracy, especially in industrial control protocols.

As a future work, the system should extend the syntax of the pattern set to make it better adapted to the changing protocol, such as dynamically constructing the sending packet for the reply packet. At the same time, it is necessary to calculate the relationship expressions of the transmission rate, transmission range, and the optimal ratio mathematically to arrange the distributed nodes to conduct detection with higher timeliness.

Data Availability

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

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Key R&D Program of China (2016QY05X1000) and the National Natural Science Foundation of China (201561402137).

References

[1] W. Wu, R. Li, G. Xie et al., "A survey of intrusion detection for in-vehicle networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 919–933, 2020.

[2] L. Yin, Y. Sun, Z. Wang, Y. Guo, F. Li, and B. Fang, "Security measurement for unknown threats based on attack preferences," *Security and Communication Networks*, vol. 2018, 13 pages, 2018.

[3] A. Almohaimeed, S. Gampa, and G. Singh, "Privacy-preserving IoT devices," in *2019 IEEE Long Island systems, applications and technology conference (LISAT)*, pp. 1–5, Farmingdale, NY, USA, 2019.

[4] L. Yin, R. Li, J. Ding et al., "δ-Calculus: a new approach to quantifying location privacy☆," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1323–1342, 2020.

[5] E. Dandil, "C-NSA: a hybrid approach based on artificial immune algorithms for anomaly detection in web traffic," *IET Information Security*, vol. 14, no. 6, pp. 683–693, 2020.

[6] L. Meftah, R. Rouvoy, and I. Chrisment, *Capturing Privacy-Preserving User Contexts with IndoorHash*, A. Remke and V. Schiavoni, Eds., Distributed Applications and Interoperable Systems DAIS 2020, Springer, Cham, 2020.

[7] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2020.

[8] L. Yin, X. Luo, C. Zhu, L. Wang, Z. Xu, and L. Hui, "ConnSpoiler: disrupting C&C communication of IoT-based botnet through fast detection of anomalous domain queries," *Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1373–1384, 2020.

[9] M. Seliem, K. Elgazzar, and K. Khalil, "Towards privacy preserving IoT environments: a survey," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 1032761, 15 pages, 2018.

[10] X. Cao, F. Zhangjie, and X. Sun, "A privacy-preserving outsourcing data storage scheme with fragile digital watermarking-based data auditing," *Journal of Electrical and Computer Engineering*, vol. 2016, Article ID 3219042, 7 pages, 2016.

[11] S. Su, Z. Tian, S. Liang, S. Li, S. Du, and N. Guizani, "A reputation management scheme for efficient malicious vehicle identification over 5G networks," *IEEE Wireless Communications.*, vol. 27, no. 3, pp. 46–52, 2020.

[12] T. Rose, K. Kifayat, S. Abbas, and M. Asim, "A hybrid anomaly-based intrusion detection system to improve time complexity in the Internet of Energy environment," *Journal of Parallel and Distributed Computing*, vol. 145, pp. 124–139, 2020.

[13] A. V. Arzhakov and I. F. Babalova, "Analysis of current internet wide scan effectiveness," in *In young researchers in electrical and electronic engineering (EIConRus), 2017 IEEE conference of Russian*, pp. 96–99, IEEE, 2017.

[14] D. Zakir, W. Eric, and J. Alex, "ZMap: fast internet-wide scanning and its security applications," in *In Proceedings of the 22nd USENIX Security Symposium*, pp. 605–619, Washington, D.C., USA, 2013.

[15] B. A. Navamani, C. Yue, and X. Zhou, "An analysis of open ports and port pairs in EC2 instances," in *In CLOUD computing (CLOUD), 2017 IEEE 10th international conference on*, pp. 790–793, IEEE, 2017.

[16] D. Zakir, A. David, M. Ariana, and B. Michael, "R a search engine backed by internet-wide scanning," in *In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 542–553, New York, USA, 2015.

[17] R. D. Grahan, *MASSCAN: Mass IP Port Scanner*, 2013, https://github.com/robertdavidgraham/masscan.

[18] R. Rodney, J. E. Vincent, and W. P. Mark, "Large scale port scanning through tor using parallel Nmap scans to scan large portions of the IPv4 range," in *In Proceedings of the 2017 IEEE International Conferrence on Intelligence and Security Informatics*, pp. 185–187, Beijing, CHN, 2017.

[19] C. Andrei, Z. Jonas, F. Aurelien, and B. Davide, "A large-scale analysis of the security of embedded firmwares," in *In proceedings of the 23rd USENIX security*, pp. 95–110, San Diego, USA, 2014.

[20] D. Zakir, L. J. Frank, and W. Nicholas, "The matter of heartbleed," in *In Proceedings of the 14th ACM Internet Measurement Conference*, pp. 475–488, Vancouver, BC, CA, 2014.

[21] H. Nadia, D. E. Zakir, and H. Alex, "Detection of widespread weak keys in network devices," in *In Proceedings of the 21st USENIX Security Symposium*, pp. 1–21, Bellevue, WA, 2012.

[22] H. Marcella, F. Joshua, and H. Nadia, "Weak keys remain widespread in network devices," in *In Proceedings of the 14th ACM Internet Measurement Conference*, pp. 275–290, Santa Monica, USA, 2016.

[23] K. Michael and B. Joseph, "Upgrading HTTPS in mid-air: an empirical study of strict transport security and key pinning," in *In Proceedings of 2015 Network and Distributed System Security Symposium*, pp. 1–15, San Diego, USA, 2015.

[24] A. V. Arzhakov and D. S. Silnov, "Architecture of multithreaded network scanner," in *In micro/nanotechnologies and Electron devices (EDM), 2017 18th international conference of young specialists on*, pp. 43–45, IEEE, 2017.

[25] Y. Fang, X. Long, L. Liu, and C. Huang, "DarkHunter: a fingerprint recognition model for web automated scanners based on CNN," in *In proceedings of the 2nd international conference on cryptography, security and privacy*, pp. 10–15, ACM, 2018.

[26] J. Black and P. Rogaway, "Ciphers with arbitrary finite domains," in *In proceedings of the Cryptographer's track at the RSA conference 2002*, pp. 114–130, San Jose, USA, 2002.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5537651, 13 pages
https://doi.org/10.1155/2021/5537651

Research Article

Link Prediction and Node Classification Based on Multitask Graph Autoencoder

Shicong Chen,[1] **Deyu Yuan**,[1,2] **Shuhua Huang**,[1,2] **and Yang Chen**[3]

[1]*School of Information and Cyber Security, People's Public Security University of China, Beijing 100038, China*
[2]*Key Laboratory of Safety Precautions and Risk Assessment, Ministry of Public Security, Beijing 100038, China*
[3]*School of Public Administration, Nanjing University of Finance & Economics, Nanjing 210023, China*

Correspondence should be addressed to Deyu Yuan; yuandeyu@ppsuc.edu.cn

Received 3 February 2021; Revised 23 March 2021; Accepted 5 April 2021; Published 19 April 2021

Academic Editor: Lihua Yin

The goal of network representation learning is to extract deep-level abstraction from data features that can also be viewed as a process of transforming the high-dimensional data to low-dimensional features. Learning the mapping functions between two vector spaces is an essential problem. In this paper, we propose a new similarity index based on traditional machine learning, which integrates the concepts of common neighbor, local path, and preferential attachment. Furthermore, for applying the link prediction methods to the field of node classification, we have innovatively established an architecture named multitask graph autoencoder. Specifically, in the context of structural deep network embedding, the architecture designs a framework of high-order loss function by calculating the node similarity from multiple angles so that the model can make up for the deficiency of the second-order loss function. Through the parameter fine-tuning, the high-order loss function is introduced into the optimized autoencoder. Proved by the effective experiments, the framework is generally applicable to the majority of classical similarity indexes.

1. Introduction

Nowadays, with the explosive growth of network data, the mainstream network representation learning algorithms are gradually difficult to adapt to the intricate data types. A variety of approaches were proposed to address privacy [1] and security [2] issues. The network is the carrier of the sophisticated relationships between data. Taking social networks as an example, large websites such as Twitter and Facebook have been consistently developing for a long time so that they can possess millions of online users. The user information scale is enormous, and the network structure is rather intricated. Thus, a mass of relationships between online users are worth exploring. By capturing the structural characteristics of real-world networks, experts and scholars can deal with multiple data analysis tasks efficiently, such as community detection [3], link prediction [4, 5], and node classification [6]. The emergence of network representation learning [7, 8] technology is of vital significance to social network analysis.

In the field of link prediction based on the classical similarity index, the CN [9] index calculates the number of common neighbors to predict the potential links between node pairs. The AA [10] index imposes a penalty on lower-connected neighbors. The Jaccard [11] index measures the similarity by comparing the proximities and differences between sample sets of common neighbors. The LP [12] index introduces the influencing factor of a third-order local path to the algorithm. The Katz [13] index improves the prediction accuracy by optimizing the LP index, by which it comprehensively extends the local path to the global path.

Motivated by Natural Language Processing [14], lots of network representation learning algorithms based on the Continuous Bag-of-Word model and Random Walk have gradually appeared. Essentially, it is a network mapping technique that each node is uniquely represented in form of low-dimensional vectors. By measuring the similarities between embedding vectors, these latent representations are probably to find the potential correlations between different entities denoted by nodes. Specifically, the low-dimensional space

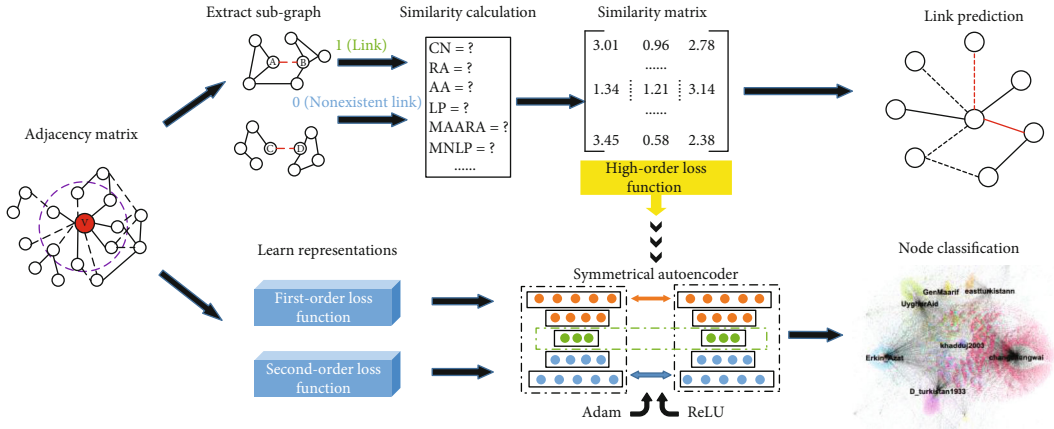

Figure 1: Architecture of our proposed multitask graph autoencoder.

can visualize the potential links in the complex network that are hard to be observed. Network representation learning not only is broadly employed to handle sophisticated social network tasks but also can be parallelized to reduce computational time.

Perozzi et al. [15] utilized the Random Walk mechanism to traverse all the network nodes deeply and preferentially. Given the initial node and walk step size, the algorithm samples a neighbor node as the next access node at random and then constitutes node access sequences of specified length in order so as to express the cooccurrence relation between nodes. After obtaining associated sampling data, the algorithm inputs sampling data into the skip-gram model for training, and the neighborhood structure of discrete nodes is then represented by vectors. Struc2vec [16] redefines node similarity from the perspective of a spatial structure. The algorithm constructs the weighted hierarchy graph by computing the node pair distances in different layers. Eventually, it leverages the generated node sequences that are structurally similar to learn network representations. Tang et al. [17] use the gradient descent method to separately optimize the first-order proximity and the second-order proximity. During the process of training, Tang et al. apply the negative sampling [18] method to decrease the time complexity.

Here, the contributions of our paper are demonstrated as follows: (1) We propose a new link prediction algorithm of mixed local neighbor and path, namely, MLNP. (2) For the deficiency of loss functions in structural deep network embedding (SDNE), our work establishes an architecture of multitask graph autoencoder (MTGAE), which designs a framework of high-order loss function from the perspective of capturing the similarity information. (3) We confirm the universal effectiveness of the loss function framework on different datasets. The specific model flow chart is shown in Figure 1.

2. Related Works

2.1. Autoencoder. As a special form of feedforward neural network, an autoencoder [19–22] is often used for dimensionality reduction feature learning in a graph embedding field. Let R^N be an N-dimensional adjacency matrix representing a graph network as input and $x_i \in R^N$ be an adjacency vector comprised of the local neighborhood structure information. The autoencoder consists of two components: the encoder $g(x_i)$: $R^N \to R^D$ and the decoder $f(y_i)$: $R^D \to R^N$. Specifically, it maps the adjacency vector to the low-dimensional embedding space composed of several nonlinear functions and acquires the approximate representation vector by effective way of compressing the graph-structured data. Then, we decode the embedding vector and represent it as the reconstruction vector $\hat{x}_i$. During the backward pass, the reconstruction loss error between the input and the output is minimized by adjusting the weight matrix cyclically. The representation vectors of latent space for different layers are computed as follows:

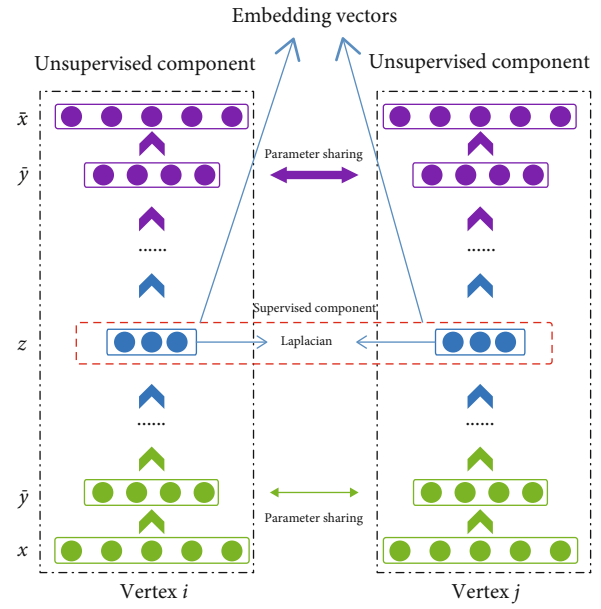

Figure 2: Traditional structural deep network embedding model.

$$y_i^{(k)} = \sigma\left(W^{(k)} y_i^{(k-1)} + b^{(k)}\right), \quad k = 2, 3, 4, \cdots, K, \tag{1}$$

Mixed local neighbor and path.
Input: edge list
Output: similarity matrix, AUC score
1: Input adjacency matrix A
2: Divide all edges into the training set and probe set
3: Construct the third-order path matrix with attenuation parameters αA^3
4: Construct optimized common neighbor matrix S_{MAARA}
5: Construct matrix based on the method of preferential attachment $S_{\mathrm{PA}} = k_x \times k_y$
6: Calculate the similarity matrix that incorporates multiple methods $S_{\mathrm{MNLP}} = S_{\mathrm{MAARA}} * S_{\mathrm{PA}} + \alpha A^3$
7: Calculate the AUC score of the MLNP index

Algorithm 1:

Multitask graph autoencoder.
Input: the network $G = (V, E)$ with adjacency matrix M, node labels, the parameters $\alpha, \beta, \gamma, \nu$
Output: network representation Y and updated parameter θ
1: Apply Adam optimizer and ReLU activation function
2: Construct the similarity matrix M
3: $X = A$
4: **Repeat**
5: Based on X, apply Equation (1) to obtain $\widehat{X}$ and $Y = Y^K$
6: $\mathrm{Loss}_{\text{high-order}} = \left\| (\widehat{X} - X) \odot M * \gamma \right\|_F^2$
7: $\mathrm{Loss}_{\mathrm{2nd}} = \left\| (\widehat{X} - X) \odot B \right\|_F^2$
8: $\mathrm{Loss}_{\mathrm{1st}} = 2\mathrm{tr}(Y^T L Y)$
9: $\mathrm{Loss}_{\mathrm{mix}} = \alpha \mathrm{Loss}_{\mathrm{1st}} + \mathrm{Loss}_{\mathrm{2nd}} + \mathrm{Loss}_{\text{high-order}} + \nu \mathrm{Loss}_{\mathrm{reg}}$
10: Use $\partial L / \partial \theta$ to backpropagate through the whole network to obtain the parameter θ
11: **Until** converge
12: Obtain the network representations $Y = Y^K$

Algorithm 2:

where $W^{(k)}$ is the weight matrix of the kth layer, $y_i^{(k-1)}$ is the $(k-1)$th layer latent vector, $b^{(k)}$ is the biases of the kth layer, and $\sigma(\cdot)$ denotes the sigmoid nonlinear activation function.

2.2. Structural Deep Network Embedding. In 2016, Wang et al. [23] put forward a structural deep network embedding model in two aspects. The first-order proximity captures local structure features of the network by judging whether nodes are linked by a direct edge [24], which can be thought of as the supervised component. Meanwhile, the second-order proximity preserves global structure features by observing the differences between the neighborhood structure of nodes, which can be regarded as the unsupervised component. Two concepts of proximity describe the characteristics of the network structure from complementary viewpoints. The SDNE model gives weights to the first-order and second-order proximity loss functions for iterative optimization, respectively. The SDNE architecture is shown in Figure 2.

The first-order loss function makes the corresponding embedding vectors of adjacent nodes $y_i^{(k)}$ and $y_j^{(k)}$ approximate in embedding spaces. The objective function is calculated as follows:

$$\mathrm{Loss}_{\mathrm{1st}} = \sum_{i,j=1}^{n} s_{i,j} \left\| y_i^{(k)} - y_j^{(k)} \right\|_2^2 = 2\mathrm{tr}\left(Y^T L Y\right), \tag{2}$$

where $\mathrm{tr}(\cdot)$ denotes the matrix trace, $s_{i,j}$ is the element of the adjacency matrix, L is the Laplace vector matrix, and Y is the encoded vector matrix of the hidden layer.

Intuitively, the second-order proximity compares the neighborhood structure of node pairs, and the proximity is computed as follows:

$$\mathrm{Loss}_{\mathrm{2nd}} = \sum_{i=1}^{n} \left\| (\widehat{x}_i - x_i) \odot b_i \right\|_2^2 = \left\| \left(\widehat{X} - X\right) \odot B \right\|_F^2, \tag{3}$$

where $\widehat{x}_i - x_i$ is the reconstruction error, $\odot$ denotes the Hadamard product, and b_i is a penalty coefficient, where $b_i = \{b_{i,j}\}_{j=1}^{n}$. If $s_{i,j} = 0$, $b_{i,j} = 1$; otherwise, $b_{i,j} = \beta > 1$. Affected by the sparsity of the network, the quantity of zero elements in the adjacency matrix is far more than that of nonzero elements. We assume that the adjacency matrix is directly

addressed as the input of SDNE; it is simpler to reconstruct the zero elements. However, this is not in accordance with our previous expectations, and a reasonable solution is to impose a higher penalty coefficient β on the reconstruction error of nonzero elements. The ultimate goal of the SDNE model is to jointly optimize the proximity loss functions, and the integral loss function is shown in

$$\text{Loss}_{\text{mix}} = \alpha\text{Loss}_{1\text{st}} + \text{Loss}_{2\text{nd}} + \nu\text{Loss}_{\text{reg}}, \tag{4}$$

where Loss_{reg} denotes the regularization term to avoid overfitting. Because of the robustness of the sparse network, performances of overall optimization are hardly affected by variations of parameters α and β.

3. Proposed Link Prediction Algorithm

In this paper, we innovatively propose an MLNP link prediction algorithm that integrates methods of common neighbors, high-order path, and preferential attachment. We adjust the structural factors of the LP index by weighing prediction accuracy against computational efficiency. The calculation method is shown in

$$s_{xy} = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \left(\frac{1}{\log|\Gamma(z)|}\right)^2 \times k_x \times k_y + \alpha A^3, \tag{5}$$

where A is the adjacency matrix, α is the attenuation parameter, and k_x and k_y denote the degrees of pairwise nodes. More importantly, $\Gamma(\cdot)$ means the neighbor nodes. By utilizing the MAARA matrix based on the AA index and RA [25] index, we highlight the importance of nodes with tremendous influence. In specific, the algorithm enhances the contribution of nodes with higher degree centralities to similarity and weakens the contribution of nodes with lower degree centralities to similarity. We distinguish common neighbors with different degree centralities to reflect the correlations between pairwise nodes more accurately. The node similarity is calculated as follows:

$$S_{xy} = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \left(\frac{1}{\log|\Gamma(z)|}\right)^2. \tag{6}$$

According to the theory of preferential attachment [12], the probability of potential links between the central node and other neighbor nodes is directly proportional to the degree centrality of the central node. Furthermore, the likelihood one link connecting pairwise nodes v_x and v_y is also directly proportional to $k_x \times k_y$. To summarize, the Hadamard product of the reconstructed MAARA matrix and PA matrix compresses the local neighborhood information so that we can thoroughly take the properties of nodes themselves, the number, and influence of common neighbors into consideration.

The above method conducts structural optimizations for the common neighbor index and explains its superiority from the theoretical level. Inspired by the idea of the global

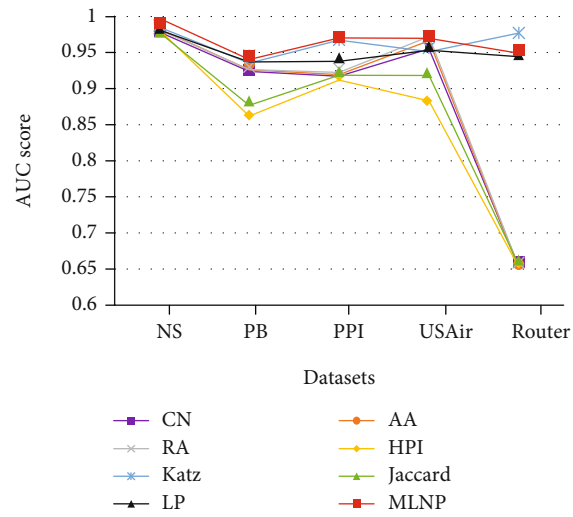

Figure 3: AUC score of different link prediction algorithms on five datasets.

Table 1: Statistics of node classification datasets.

Network	Node	Edge	Label	Average degree	Average path length
Europe-flight	399	5995	4	30.1	2.28
Brazil-flight	131	1074	4	16.4	2.71

path, as the number of intermediate nodes in local paths increases, the weight parameter of the high-order path will decay. Intuitively, the number of second-order paths is equal to the number of common neighbors that have been discussed, indicating that the weight of the third-order path is the highest. Hence, our work innovatively introduces the factor of third-order path combined with the above matrices into the ultimate similarity matrix so as to produce a substantial boost on prediction accuracies. The basic algorithm procedure is shown in Algorithm 1.

4. Multitask Graph Autoencoder

4.1. High-Order Loss Function. The deficiency in second-order proximity of the SDNE model is explained as follows: When imposing a penalty coefficient β on nonzero elements, the only criterion for measuring similarities is whether an edge exists between pairwise nodes. Factually, the properties of common neighbors, the length of paths, and even the attenuation parameters will bring about deviations in the process of computing similarities. The adjacency matrix only describes the actual condition, while the similarity matrix reveals the hidden structural similarity of the network. For instance, a couple of individuals who have more common friends are more likely to establish friendships, even though they do not get acquainted with each other before. In network topology, we can directly observe the explicit links but may

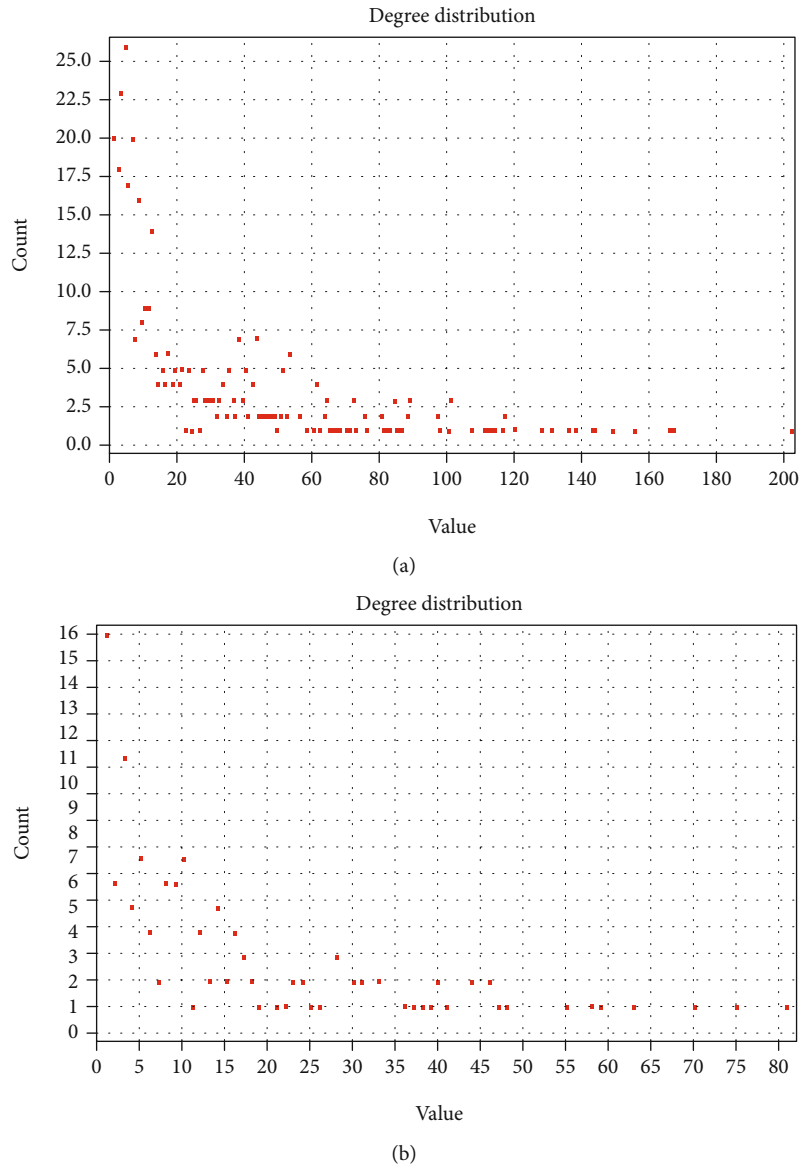

Figure 4: Degree distribution of (a) Europe-flight network and (b) Brazil-flight network.

ignore the potential links simultaneously. Thus, the idea of applying the adjacency matrix only is single that seeking the potential links inferred by the algorithm is the key to lifting the capability of our model.

The high-order proximity and second-order proximity are complementary in that they, respectively, punish matrix elements according to the explicit similarity and the hidden similarity of the network structure. By using the backpropagation algorithm, we cyclically minimize the introduced high-order loss function error. In detail, the reconstructed high-order loss function is defined as follows:

$$\text{Loss}_{\text{high-order}} = \sum_{i=1}^{n} \left\| (\widehat{x_i} - x_i) \odot M * \gamma \right\|_2^2, \tag{7}$$

where M is the similarity matrix and γ is the adjustment parameter. Parameter γ directly controls the fluctuation range of similarity and constrains the reconstruction weight. We believe that γ should be consistent with β (1-20), or the different loss functions will exhibit extreme imbalance. Our model has its advantages in addressing the tasks of link prediction and semisupervised node classification at the same time. In specific, we borrow the idea of link prediction, which takes the output similarity matrix as an intermediate product, and then, we input the processed vector matrix into a stacked autoencoder.

4.2. Optimization of Autoencoder. In our experiment, we use the Keras [26] module to implement two layers of encoder and decoder at the CPU-enabled Tensorflow [27] backend. The hidden layer dimensionality of our model architecture is fixed at N-256-128-256-N. Due to the abandonment of the deep belief network [28] structure for parameter pretraining, the SGD optimizer and Sigmoid activation function applied by the original SDNE algorithm may lead to the

cessation of training. Alternatively, our architecture attempts to apply the Adam [29] algorithm with a fixed learning rate and ReLU [30] activation function for optimization.

The Adam optimizer has the characteristics of inertia retention and environmental perception. The method of calculating a new round of gradient descent is the linear weighting of the current real gradient with the gradient used in the previous round for gradient descent. The superiority of its adaptive learning efficiency lies in overcoming the network sparsity problem effectively. Compared with the SGD optimizer which is easy to converge to the local optimum and trapped in the saddle point, the Adam optimizer is recognized for accelerating the convergence speed and maintaining the convergence stability. However, the adaptive learning rate algorithm of the Adam optimizer performs worse in the fields of object recognition and syntax component analysis. In the deep neural network (DNN), the gradient of the Sigmoid activation function is very small at a position away from point 0. During the backpropagation phase, the information loss problem caused by gradient disappearance may occur, and computation of the partial derivative involved with division may increase the time complexity of the algorithm. ReLU activation function, however, can effectively alleviate this type of vanishing gradient issue and perform well in enhancing computational efficiency. To summarize, the complete algorithm is shown in Algorithm 2.

5. Link Prediction Experiments

5.1. Datasets and Evaluation Metrics. For link prediction, we evaluate our MLNP algorithm on five classical graph-structured datasets. The fundamental information of datasets is introduced as follows.

NS [31] is a collaboration network of scientists who have published distinguished papers on the topic of complex networks. An observed link is present if there is a cooperative relationship between scientists in papers. PB [32] is an American political blog network that documents the links between blogs extracted from network websites. PPI [33] is a protein-protein interaction network. The nodes denote macromolecules of proteins, and the links indicate the interactions between a couple of proteins. USAir [34] is an aviation network of the United States that each node corresponds to a termination. If there is a direct air route between terminations, it means that there is a connection between nodes. Router is a router [35] network on the Internet, where nodes denote routers and edges directly connect the two routers for packet exchange through optical fiber or other means.

Our work adopts the most widely used AUC score to evaluate our proposed MLNP algorithm on link prediction tasks. It can be explained as the likelihood that the randomly selected test links score higher than stochastically selected nonexistent links. In contrast to the *precision@k* evaluation indicator, the AUC score overall measures the prediction accuracy. It is defined as follows:

$$\mathrm{AUC} = \frac{n' + 0.5n''}{n}. \tag{8}$$

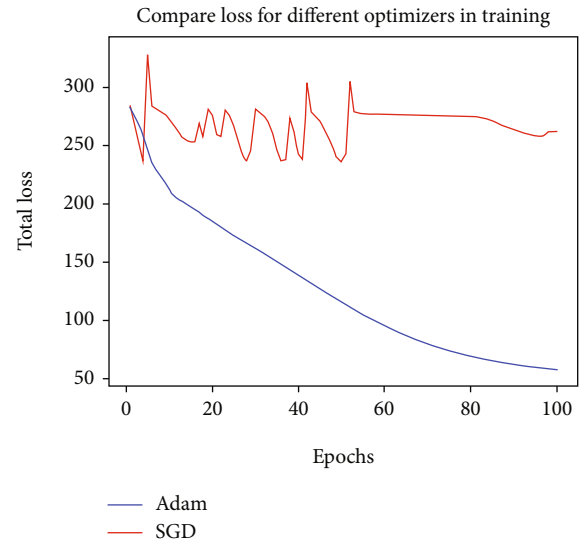

Figure 5: Comparison of loss convergence between Adam optimizer and SGD optimizer.

Table 2: Parameter settings.

Epoch	α	β	l1	l2
100	$1e-4$	5	$1e-6$	$1e-5$

Among n times of experimental comparisons, n' denotes the occurrences of missing links that score higher than nonexistent links, while n'' denotes the occurrences of having the same score.

5.2. Result Analysis. To guarantee a more fine-grained comparison, we empirically choose 90% links at random as the training set, and the remaining 10% links constitute the probe set for prediction. We summarize the consequences of link prediction for five datasets in Figure 3.

In comparison to other strong baselines [36], the experiment results explicitly show that the formulated MLNP algorithm consistently achieves the best AUC performance on three datasets {NS, PB, and PPI}, respectively, 1.24%, 0.33%, and 0.3% higher than the best baseline. Although the prediction accuracy of our method is slightly 0.24% lower than the RA index on the USAir dataset and 2.78% lower than the Katz index on the Router dataset, it remains competitive compared with the rest of the similarity indexes.

On small-scale datasets, we explicitly observe that our method outperforms all other baselines, even exceeding the Katz index based on the global path. To our surprise, the prediction accuracy reaches 99.7% on the NS dataset. However, the formulated algorithm gets worse AUC performances than the RA index and Katz index on large-scale datasets. The possible reasons are twofold. Firstly, with the increase of diameter and average path length of the network, it is far from enough that the MLNP algorithm only captures local information. Secondly, the Katz index preserves the global structures adequately by traversing the network. The experiments reveal that

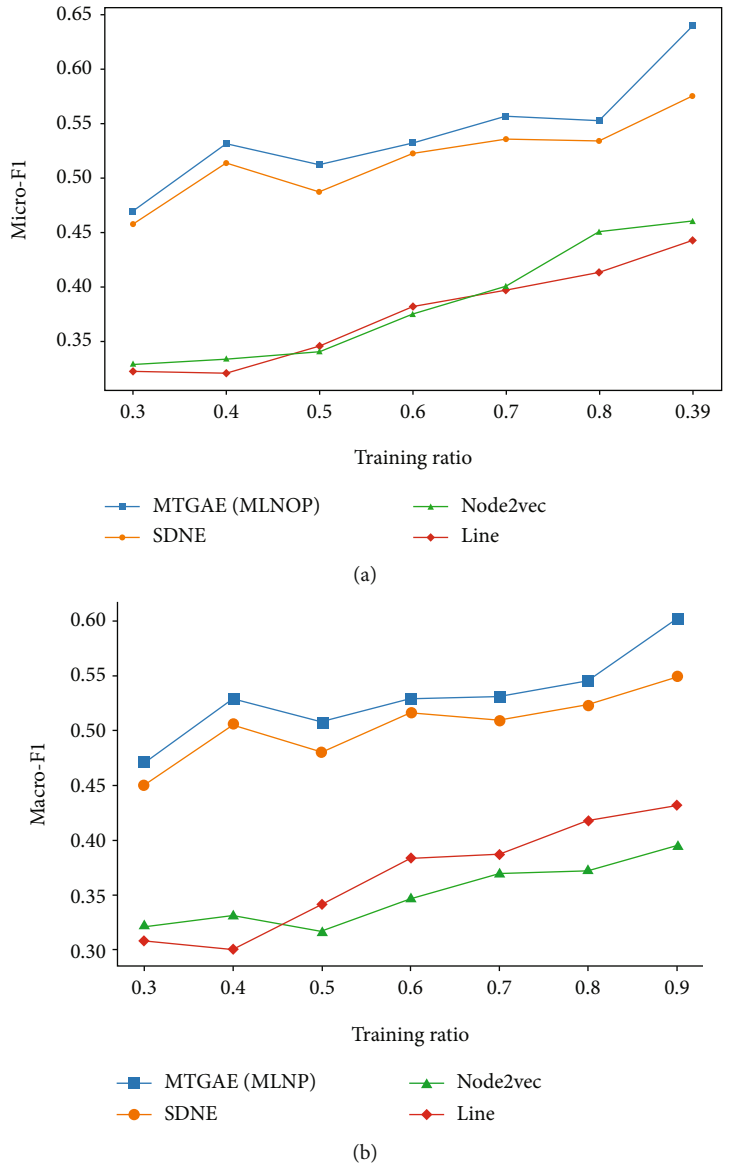

Figure 6: (a) Micro-F1 and (b) Macro-F1 of MTGAE on Europe-flight dataset.

the MLNP algorithm is quite effective for optimization of the original similarity index. We attribute the efficacy of our innovation to multiple integrated methods.

6. Node Classification Experiments

6.1. Datasets and Evaluation Metrics. We select two air transportation networks of Europe-flight and Brazil-flight to assess the effects of representations. Specifically, the dataset contents comprise nodes, links, and node labels that 399 nodes and 5995 links exist in the Europe-flight network, and 131 nodes and 1074 links exist in the Brazil-flight network. Both datasets divide node labels into four categories, and the detailed statistics of network attributes are computed in Table 1.

To ensure that the adopted similarity theory can traverse the network locally and globally, we calculate the degree distribution of nodes as well. According to the simulation consequences, although the quantity of network nodes decreases, the structure information is adversely more intact due to the relatively high link density and average node degree. The exact degree distributions of datasets are shown in Figure 4.

Empirically, we employ the current popular F1-measure indicator [367] to evaluate the quality of graph embedding representations, and the calculation method is defined as follows:

$$\begin{aligned}\text{precision} &= \frac{\sum_{A\in C}\text{TP}(A)}{\sum_{A\in C}(\text{TP}(A)+\text{FP}(A))},\\ \text{recall} &= \frac{\sum_{A\in C}\text{TP}(A)}{\sum_{A\in C}(\text{TP}(A)+\text{FN}(A))},\\ \text{F1-measure} &= \frac{2*\text{precision}*\text{recall}}{\text{precision}+\text{recall}}.\end{aligned} \tag{9}$$

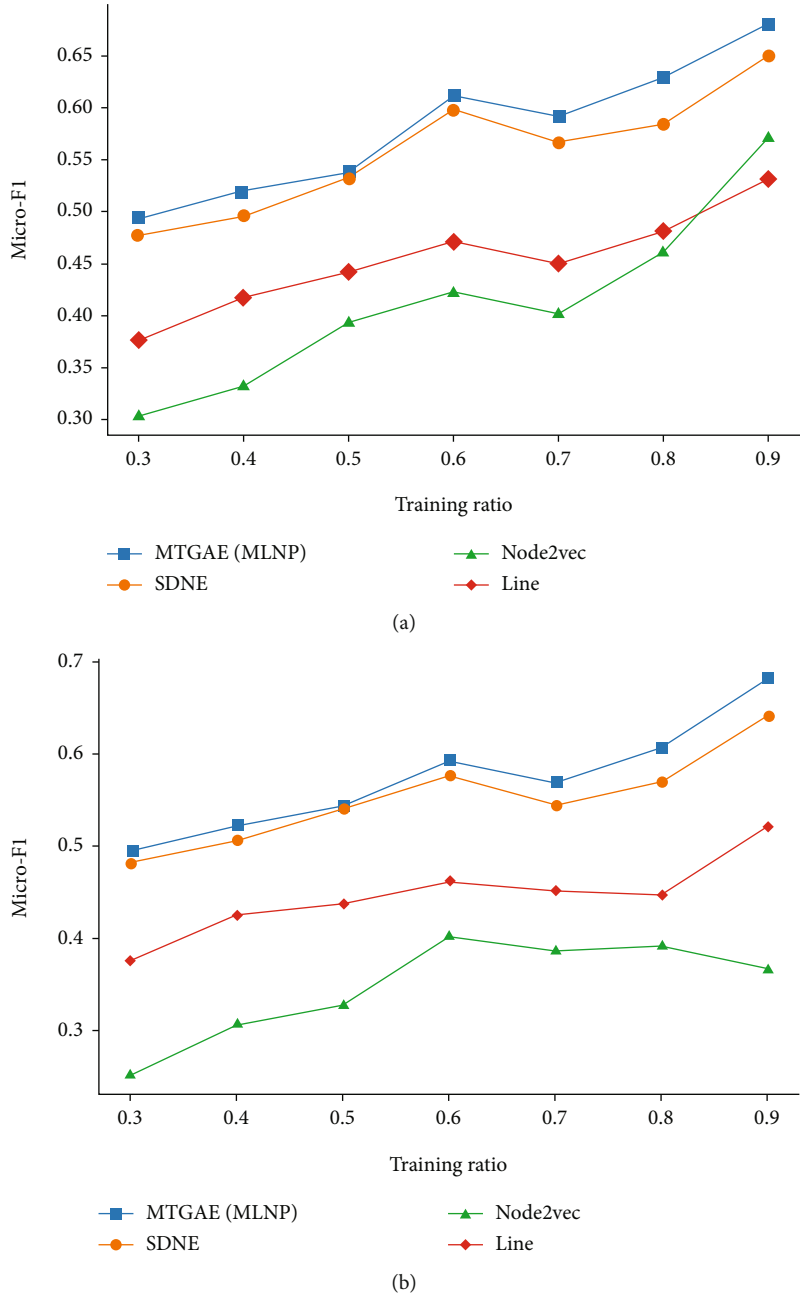

Figure 7: (a) Micro-F1 and (b) Macro-F1 of MTGAE on Brazil-flight dataset.

6.2. Loss Convergence Comparison of Optimizers. To check the loss convergence of the Adam optimizer and SGD optimizer, we apply the control variable method to perform 100 iterative training epochs on the premise of consistent model parameters. The simulation experiments of loss convergence are shown in Figure 5.

In this experiment, the results obviously reveal that the architecture combined with the Adam optimizer converges more quickly and more stably. Under the same circumstances, there is no doubt that the capability of the Adam optimizer is better compared with the SGD optimizer.

6.3. Result Analysis. We set the training batch size of our model to the total number of nodes in one network. To ensure the consistency of other model parameters, our work configures the training parameters of the MTGAE model shown in Table 2. Specifically, the weight parameters of first-order and second-order proximity should remain strictly constant. Affected by the negative effect of overfitting, the autoencoder applies the regularization to limit the weight threshold value in the fully connected neural network.

The feature learning of network structure is insufficient when we train on fewer nodes. Considering the contingent consequences that may appear, we determine to give up sampling 10% and 20% of the observed links in networks for training. Instead, when the training percentage increases from 30% to 90%, every time, we calculate the mean value of 10 experiments to compare the performances between

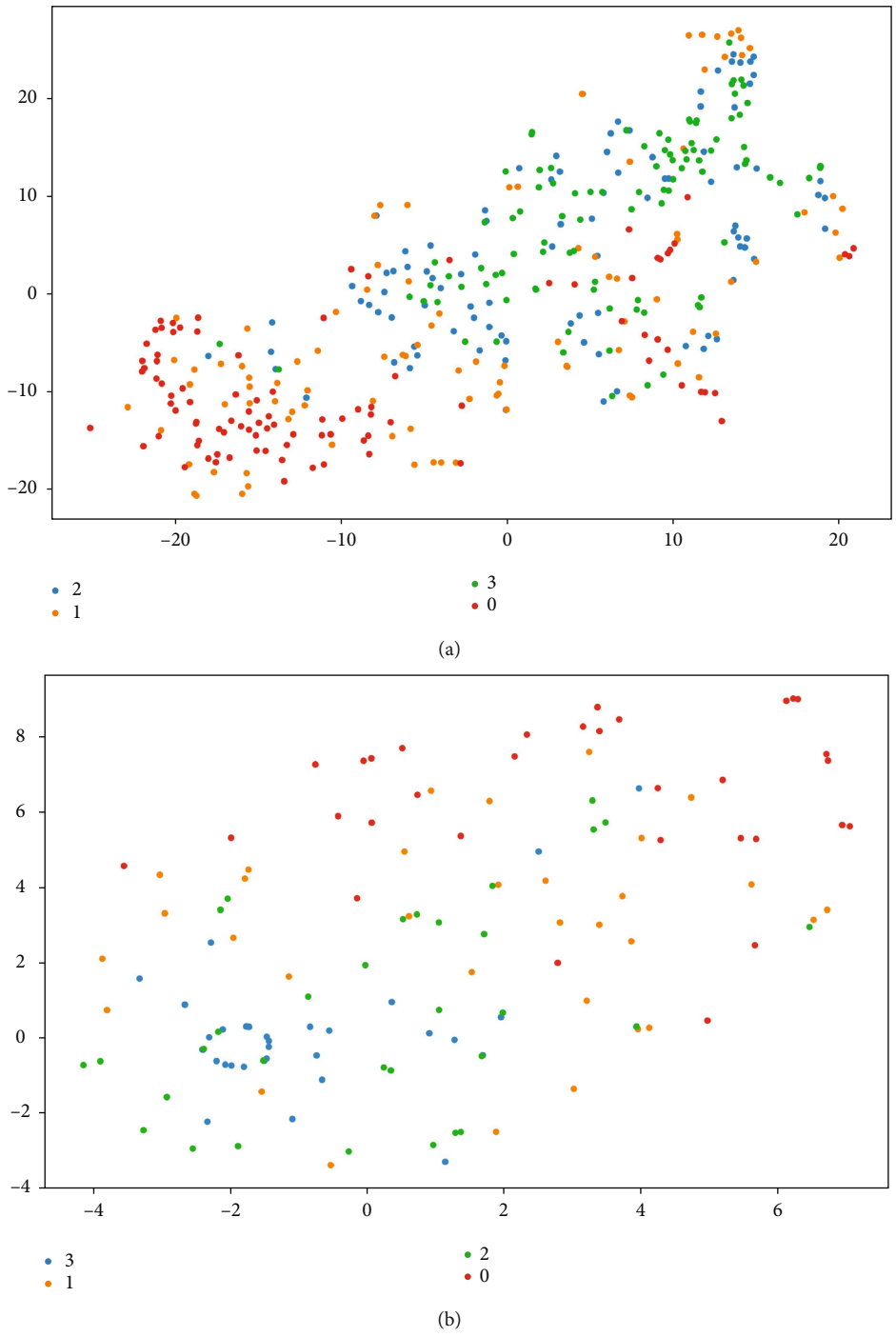

Figure 8: Visualization of (a) Europe-flight network and (b) Brazil-flight network.

the MTGAE (MLNP) algorithm and the classical SDNE algorithm. Moreover, the mainstream algorithm of Line and Node2vec [37] is chosen as benchmarks as well, and the actual consequences of node classification are shown in Figures 6 and 7.

By carefully calculating the experiment results, we discover that under different proportions of training sets, the proposed MTGAE (MLNP) model applied in Europe-flight and Brazil-flight networks boosts the average Micro-F1 by 2.42% and 2.25%, respectively, and enhances the average Macro-F1 by 2.54% and 2.21%, respectively. When the training percentage is up to 90%, it means that the algorithm completely learns the network representations, and the promotion of node classification accuracy reaches the climax, even 5%-6%. It can be seen that whatever proportion of the training set is divided by the experiment, both the Micro-F1 and Macro-F1 of our algorithm are generally higher than those of the related algorithms. We find that our algorithm

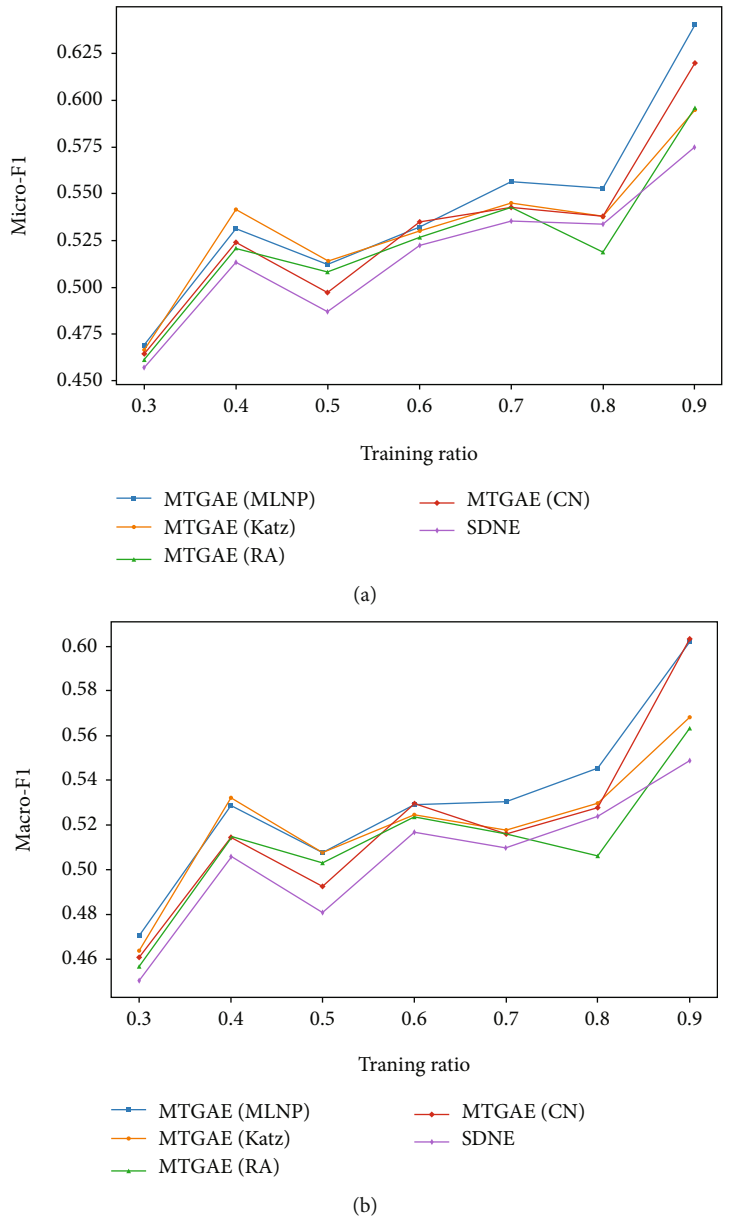

Figure 9: (a) Micro-F1 and (b) Macro-F1 of loss function framework on Europe-flight dataset.

can promote both the evaluation metrics, indicating that the introduced high-order proximity can capture the structure features better in latent spaces and achieve ideal classification effects. The visualizations of the two datasets are shown in Figure 8.

6.4. Horizontal Contrast of Loss Function Framework. In order to verify the universal validity of the high-order loss function framework, we separately adopt the same processing method as the MLNP index for the CN index, RA index, and Katz index. In two different networks, the horizontal contrasts of our experiments are shown in Figures 9 and 10.

The results reveal that no matter what kind of similarity index we introduce into the framework of the high-order loss function, the MTGAE model is superior to the SDNE model except for a couple of special cases on two datasets. Only when we randomly sample 80% of the links in the Europe-flight network and stochastically sample 50% of the links in the Brazil-flight network for training, the SDN model can behave better slightly than one or two other models. The underlying cause is the particularity of datasets. Moreover, it can be found that when we convert the MLNP index and Katz index to high-order loss functions, the improvement margin of node classification is more apparent. The accurate results are shown in Table 3. We choose the MTGAE model with the best prediction accuracy to display the specific improvement margin compared with the SDNE model. Hence, the experiment consequences demonstrate that the introduced framework of high-order loss function is generally effective in boosting the accuracy of node classification.

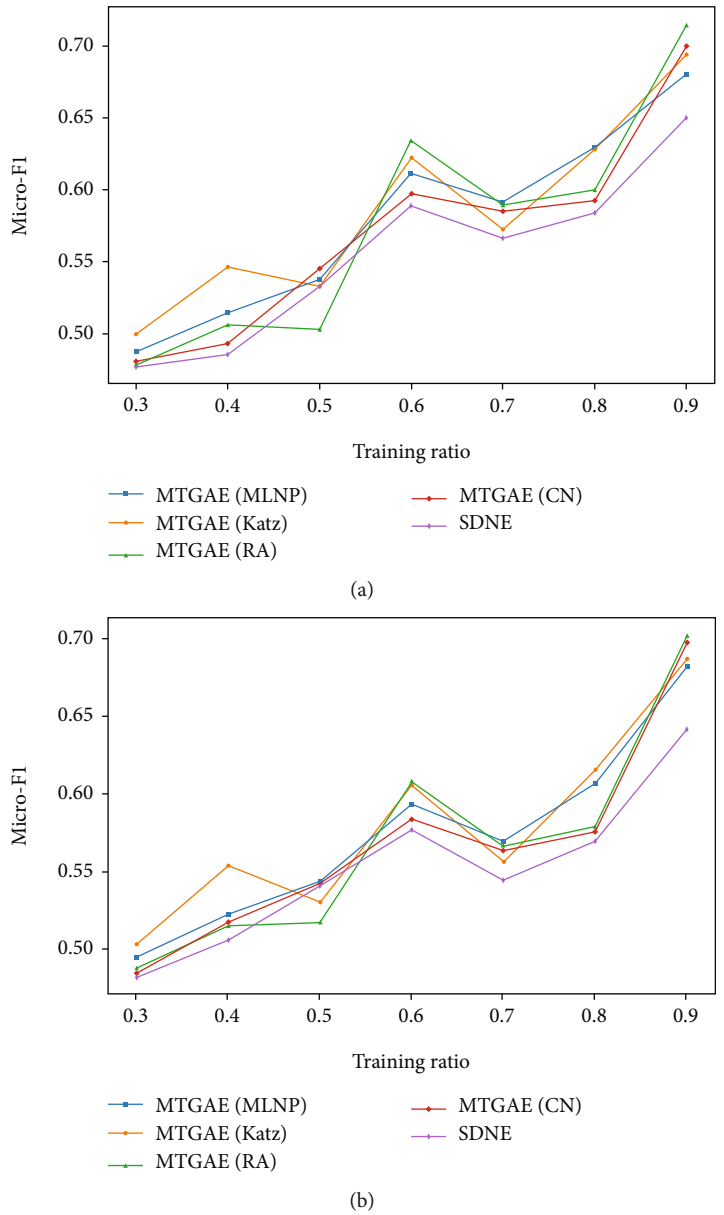

Figure 10: (a) Micro-F1 and (b) Macro-F1 of loss function framework on Europe-flight dataset.

Table 3: Performance of MTGAE model compared with SDNE.

Training ratio	Europe-flight				Brazil-flight			
	SDNE		MTGAE		SDNE		MTGAE	
	Micro	Macro	Micro	Macro	Micro	Macro	Micro	Macro
0.3	0.457	0.451	0.469	0.471	0.477	0.482	0.499	0.503
0.4	0.513	0.506	0.541	0.532	0.496	0.506	0.547	0.554
0.5	0.487	0.481	0.514	0.508	0.533	0.541	0.545	0.544
0.6	0.522	0.516	0.535	0.523	0.589	0.577	0.633	0.608
0.7	0.535	0.510	0.556	0.531	0.567	0.544	0.592	0.569
0.8	0.534	0.524	0.553	0.546	0.584	0.568	0.629	0.615
0.9	0.575	0.549	0.64	0.603	0.650	0.641	0.714	0.702

7. Conclusions

In this paper, we put forward an MLNP similarity algorithm that integrates multiple similarity theories. In addition, we establish an architecture of the MTGAE model which introduces the high-order loss function into an optimized autoencoder by preprocessing the similarity index. The extraordinary innovation of the MTGAE model is that it successfully applies the link prediction methods to the field of node classification. Specifically, the MLNP index of link prediction is used as an intermediate product to construct the high-order loss function. The above algorithms perform favorably well in both applications of link prediction and node classification. Furthermore, our work applies different similarity matrices as the high-order loss functions to verify the universal validity of the framework. The results demonstrate that our framework of high-order loss function adapts to the majority of popular similarity indexes.

With the continuous development and innovation of deep learning, numerous deep models with side information of nodes and edges emerge in an endless stream. However, some static models can no longer satisfy the needs of a broad range of practical applications. Experts and scholars have gradually turned their attention to dynamic graph embedding models. Although some professors have put forward algorithms to address the dynamic network, quite efficient methods to handle the multidimensional features still lack. The dynamic network is increasingly becoming a significant research object. Embedding the features of nodes and edges into autoencoder architecture and building dynamic evolution models are becoming significant research directions to extend graph embedding technologies. In the future, the majority of models to address the network representation learning problems have broad application prospects in such as recommender systems [38] and mobile computing [39].

Data Availability

The data used to support the findings of this study are publicly available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61771072), Special Project of People's Public Security University of China (Grant No. 2020JWCX01), and Open Project of the Key Laboratory of the Police Internet of Things Application Technology (Ministry of Public Security of China).

References

[1] Z. Sun, L. Yin, C. Li, W. Zhang, A. Li, and Z. Tian, "The QoS and privacy trade-off of adversarial deep learning: an evolutionary game approach," *Computers & Security*, vol. 96, p. 101876, 2020.

[2] L. Yin, B. Fang, Y. Guo, Z. Sun, and Z. Tian, "Hierarchically defining Internet of Things security: from CIA to CACA," *International Journal of Distributed Sensor Networks*, vol. 16, no. 1, Article ID 1550147719899374, 2020.

[3] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics*, vol. 2008, no. 10, pp. 155–168, 2008.

[4] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proceedings of The 33rd International Conference on Machine Learning*, pp. 2071–2080, New York, New York, USA, 2016.

[5] L. Y. Lv, C. H. JIN, and T. Zhou, "Similarity index based on local paths for link prediction of complex network," *Physical Review E*, vol. 80, no. 4, pp. 211–223, 2009.

[6] W. Li, D. Yin, D. Yuan, B. Wang, and Y. Gu, "Particle propagation model for dynamic node classification," *IEEE Access*, vol. 8, pp. 140205–140215, 2020.

[7] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Network representation learning: a survey," *IEEE transactions on Big Data*, vol. 6, pp. 3–28, 2018.

[8] L. Cai, Y. Xu, T. He, T. Meng, and H. Liu, "A new algorithm of DeepWalk based on probability," *Journal of Physics: Conference Series*, vol. 1069, no. 1, pp. 130–135, 2019.

[9] F. Lorrain and H. C. White, "Structural equivalence of individuals in social networks," *Journal of Mathematical Sociology*, vol. 1, no. 1, pp. 49–80, 1971.

[10] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[11] P. Jaccard, "Etude comparative de la distribution florale dans une portion des Alpes et des Jura," *Bulletin of the Torrey Botanical Club*, vol. 37, p. 547, 1901.

[12] T. Zhou, L. Lv, and Y. C. Zhang, "Predicting missing links via local information," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 71, no. 4, pp. 623–630, 2009.

[13] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[15] B. Perozzi, R. Alrfou, and S. Skiena, "Deepwalk: online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 701–710, Newyork, USA, 2014.

[16] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, "struc2vec: learning node representations from structural identity," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 385–394, Halifax, NS, Canada, 2017.

[17] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: large- scale information network embedding," in *Proceedings of the 24th international conference on world wide web*, pp. 1067–1077, Florence, Italy, 2015.

[18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 3, pp. 2672–2680, 2014.

[20] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layerwise training of deep networks," in *International conference on neural information processing systems*, pp. 153–160, Stony Brook University, 2006.

[21] M. Ranzato, Y. L. Boureau, and Y. Lecun, "Sparse feature learning for dep belief networks," *Advances in Neural Information Processing Systems*, vol. 20, pp. 1185–1192, 2007.

[22] N. K. Tomas and W. Max, *Variational Graph Auto-Encoders*, vol. 28, no. 3, 2016Springer, 2016.

[23] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd international conference on knowledge discovery and data mining*, pp. 1225–1234, San Francisco, CA, USA, 2016.

[24] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the Association for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[25] Q. Ou, Y. D. Jin, T. Zhou, B. H. Wang, and B. Q. Yin, "Power-law strength-degree correlation from resource-allocation dynamics on weighted networks," *Physical Review E*, vol. 75, no. 2, article 021102, 2007.

[26] N. Ketkar, "Introduction to keras," in *Deep learning with Python*, pp. 97–111, Apress, Berkeley, CA, 2017.

[27] M. Abadi, A. Agarwal, P. Barham et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015, https://github.com/tensorflow/tensorflow.

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

[29] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, San Diego, USA, 2015.

[30] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning*, Toronto, Canada, 2010.

[31] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical review E*, vol. 74, no. 3, 2006.

[32] R. Ackland, *Mapping the US Political Blogosphere: Are Conservative Bloggers More Prominent*, Presentation to Blog Talk Downunder, Sydney, 2005, http://incsub.org/blogtalk/images/robertackland.pdf.

[33] C. Von Mering, R. Krause, B. Snel et al., "Comparative assessment of large-scale data sets of protein-protein interactions," *Nature*, vol. 417, no. 6887, pp. 399–403, 2002.

[34] V. Batageli and A. Mrvar, *Pajek Datasets*http://vlado.fmf.unilj.si/pub/networks/data/default.htm.

[35] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with rocketfuel," *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, pp. 2–16, 2004.

[36] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabási, "Hierarchical organization of modularity in metabolic networks," *Science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[37] A. Grover and J. Leskovec, "node2vec: scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, San Francisco, CA, USA, 2016.

[38] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[39] L. Yin, "Threat-based declassification and endorsement for mobile computing," *Chinese Journal of Electronics*, vol. 28, no. 5, pp. 1041–1052, 2019.